

**МІНІСТЕРСТВО ОХОРОНИ ЗДОРОВ'Я УКРАЇНИ
ЗАПОРІЗЬКИЙ ДЕРЖАВНИЙ МЕДИКО-ФАРМАЦЕВТИЧНИЙ
УНІВЕРСИТЕТ**

О.А. Рижов, Н.І.Строїтелєва

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У ФАРМАЦІЇ
Навчальний посібник для самостійної роботи з підготовки до практичних
занять для студентів 2 курсу
спеціальності 226 «Фармація, промислова фармація»

м.Запоріжжя, 2023

УДК 615.1:004.9
Р 93

*Затверджено на засіданні Центральної методичної Ради ЗДМФУ
та рекомендовано для використання в освітньому процесі
(протокол № 5 від «23» травня 2023 р.)*

Розробники:

О.А Рижов – д-р фарм.н., проф., завідувач кафедри медичної та фармацевтичної інформатики і НТ ЗДМУ
Н.І. Строїтелева – к.ф.-м.н., доцент кафедри медичної та фармацевтичної інформатики і НТ ЗДМУ

Рецензенти:

Б.С.Бурлака - д.фарм.н., доцент кафедри технології ліків ЗДМФУ
Д.Г. Іванченко – д.фарм.н., доцент кафедри біологічної хімії ЗДМФУ
доцент

Р93

Рижов О.А.

Інформаційні технології у фармації: навчальний посібник для самостійної роботи з підготовки до практичних занять для студентів 2 курсу спеціальності 226 «Фармація, промислова фармація» / Рижов О.А., Строїтелева Н.І. – Запоріжжя: ЗДМФУ, 2023. – 169 с.

Навчальний посібник «Інформаційні технології у фармації» призначений для самостійної роботи студентів з метою отримання практичних навичок з вирішення фармацевтичних задач за допомогою сучасних комп'ютерних технологій. Посібник розрахований на студентів 2 курсу очної та заочної форм навчання для підготовки для практичних занять з навчальної дисципліни «Інформаційні технології у фармації»

© Рижов О. А., Строїтелева Н. І. 2023.
©Запорізький державний медико-фармацевтичний університет,
2023

ЗМІСТ

ВСТУП	4
ПРАКТИЧНЕ ЗАНЯТТЯ№1. ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА СТУДЕНТА НА БАЗІ ХМАРНИХ ТЕХНОЛОГІЇ MS OFFICE 365	5
ПРАКТИЧНЕ ЗАНЯТТЯ№2. АРХІТЕКТУРА СУЧАСНИХ КОМП'ЮТЕРНИХ ЗАСОБІВ. ОСНОВНІ ФУНКЦІОНАЛЬНІ ВУЗЛИ (КОМПОНЕНТИ) ТА ПРИНЦИПИ ЇХ ВЗАЄМОДІЇ. РІЗНОВИДИ ОПЕРАЦІЙНИХ СИСТЕМ.	32
ПРАКТИЧНЕ ЗАНЯТТЯ№3. ОСНОВИ ПРОЕКТУВАННЯ НАВЧАЛЬНИХ ЕЛЕМЕНТІВ ДЛЯ ОНЛАЙН КУРСІВ EDX. ОБРОБКА МЕДИЧНИХ ЗОБРАЖЕНЬ У ПРОГРАМІ ACD CHEMSKETCH. 3D – МОДЕЛЮВАННЯ ХІМІЧНИХ СПОЛУК....	57
ПРАКТИЧНЕ ЗАНЯТТЯ№4. СТАТИСТИЧНІ МЕТОДИ ОБРОБКИ РЕЗУЛЬТАТІВ ФАРМАЦЕВТИЧНИХ ДОСЛІДЖЕНЬ ЗА ДОПОМОГОЮ ЕЛЕКТРОННИХ ТАБЛИЦЬ MS EXCEL	79
ПРАКТИЧНЕ ЗАНЯТТЯ№ 5. КОМП'ЮТЕРНІ ДАНІ: ТИПИ ДАНИХ, ОБРОБКА ТА УПРАВЛІННЯ. SASHE OBJECTSCRIPT. РОБОТА В ТЕРМІНАЛЬНОМУ РЕЖИМІ. ОПЕРАТОРИ ВВЕДЕННЯ-ВИВЕДЕННЯ, АРИФМЕТИЧНІ ОПЕРАТОРИ, ОПЕРАТОРИ ВИБОРУ, МАСИВИ І РЯДКОВІ ЗМІННІ.	101
ПРАКТИЧНЕ ЗАНЯТТЯ№6. АЛГОРИТМІЗАЦІЯ ЦИКЛІЧНИХ ПРОЦЕСІВ. ФОРМАЛЬНА ЛОГІКА. МАСИВИ ТА ВБУДОВАНІ ФУНКЦІЇ НАВІГАЦІЇ В МАСИВАХ. ІНСТРУМЕНТАЛЬНЕ СЕРЕДОВИЩЕ STUDIO INTERSYSTEM.	118
ПРАКТИЧНЕ ЗАНЯТТЯ№ 7. ОСНОВИ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПІДХОДУ К АНАЛІЗУ ЗАДАЧ ФАРМАЦЕВТИЧНОГО ПРОФІЛЮ. UML – МОДЕЛЮВАННЯ ОБ'ЄКТІВ ПРЕДМЕТНОЇ ГАЛУЗІ	133
ПРАКТИЧНЕ ЗАНЯТТЯ№8. ПРИНЦИПИ ОРГАНІЗАЦІЇ WEB - САЙТУ. МОВА РОЗМІТКИ HTML.	155
ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ:	16969

ВСТУП

Предметом вивчення навчальної дисципліни «Інформаційні технології у фармації» є сучасні інформаційні технології, які включають системні та прикладні комп'ютерні програми та технології й засоби об'єктно-орієнтованого моделювання у фармацевтичній галузі

Навчальний посібник містить 8 практичних занять згідно з робочою програмою дисципліни, що передбачає 54 години на самостійну роботу студента. Кожне заняття містить розвинену теоретичну інформацію щодо програмного та апаратного забезпечення персонального комп'ютера, а також формалізації та моделювання інформаційних систем у фармації. У кожному занятті також наведені практичні рекомендації, що призначені для опанування комп'ютерних систем навчання та процесів розробки прикладних програмних засобів у фармації. Велика кількість наведених прикладів з вирішення практичних завдань сприяє полегшенню самостійного засвоєння студентом навчального матеріалу. Перелік рекомендованої сучасної літератури, що міститься у підручнику, може бути використаний студентом самостійно для подальшого поглибленого вивчення прикладних програмних засобів у фармації та вдосконалення своєї майстерності.

ПРАКТИЧНЕ ЗАНЯТТЯ №1

Тема: Організація інформаційного середовища студента на базі хмарних технологій MS Office 365

Мета роботи: Опанувати навичками організації інформаційного середовища для навчання за допомогою MS Office 365

Теоретичні відомості

1. Хмарні технології в роботі лікаря

Електронна медицина (від англ. e-health; також е-медицина) — це діяльність з використанням електронних інформаційних ресурсів у сфері охорони здоров'я та забезпечення оперативного доступу медичних працівників та пацієнтів до них. Е-медицина є відносно новим терміном охорони здоров'я за підтримки електронних процесів і комунікацій, що почав набувати розповсюдження у світі з 1999 року.

Форми електронної медицини

Термін е-медицина включає в себе широкий спектр послуг та систем, які знаходяться на перетині медицини/охорони здоров'я та інформаційних технологій, у тому числі:

–електронні медичні картки: дозволяють передачу та обмін даними про пацієнта між різними медичними фахівцями;

–електронний рецепт: електронний доступ до надання або зміни призначень, друк рецептів для пацієнтів та іноді електронна передача рецептів від лікарів до фармацевтів;

–телемедицина: фізичні та психологічні методи лікування на відстані, в тому числі телемоніторинг стану пацієнтів;

–інформування населення: використання електронних ресурсів на медичні теми для інформування пацієнтів, забезпечення обізнаності населення з питань охорони здоров'я;

–віртуальні медичні бригади: складаються з медичних фахівців, які співпрацюють і обмінюються інформацією про пацієнтів за допомогою цифрового обладнання (наприклад, веб-конференції);

–мобільна медицина: включає в себе використання мобільних пристроїв для збору даних про фізичний чи психологічний стан пацієнта, які передаються лікарям чи дослідникам, а також моніторинг життєво важливих органів пацієнта в режимі реального часу, пряме надання допомоги (за допомогою мобільної телемедицини);

–медичні інформаційні системи: спеціалізовані програми для планування графіку прийому лікарями пацієнтів, управління даними пацієнта, управління розкладом роботи та інших адміністративних завдань, пов'язаних із організацією охорони здоров'я;

–розробка власного раціону: предметна галузь е медицини надає сучасні методи діагностики, моніторингу (контролю і самоконтролю) та оцінки стану здоров'я людини, що допомагають визначити зміни стану організму у залежності від тих чи інших зовнішніх і внутрішніх чинників. Важливим компонентом є індивідуалізація процесу підбору оптимального раціону залежно від характеристики стану різних функцій організму пацієнта.

Окремим випадком електронної медицини є спільний проект Міністерства оборони України та волонтерів зі створення інноваційної медичної інформаційної системи, яка має стати основою для реформування військової медицини «Е-Здоров'я». Головна місія проекту — порятунок життя поранених в зоні АТО через залучення сучасних ІТ-технологій. Проект передбачає впровадження централізованої системи ведення електронної медичної картки та підтримки процесів надання допомоги військовослужбовцям.

Мобільна медицина - цей новий напрям в медицині, пов'язаний з використанням облаштувань типу стільникових телефонів або планшетних комп'ютерів, які лікар постійно носить з собою. Використання хмарних технологій в медицині дозволяє постійно, в режимі реального часу, взаємодіяти з пацієнтом/лікарем. За допомогою спеціальних пристроїв ми можемо зареєструвати електрокардіограму, виміряти вагу, тиск, подивитися, як тиск міняється протягом доби. Отримані дані передаються на облаштування лікаря або "в хмару". Таким чином, ці дані можна проаналізувати в динаміці.

Хвиля mHealth

У всьому світі мобільна медицина відома під назвою mHealth. Це поняття включає використання мобільного зв'язку та пристроїв для надання послуг як пацієнтам, так і лікарям. За прогнозами аналітичної компанії Research2guidance, до кінця 2017 р. mHealth об'єднає понад 3,4 млрд користувачів смартфонів і планшетів. Перспектива цілком реальна, про що свідчать цифри статистики: якщо в 2004 р. у США лише кожен четвертий лікар-практик використовував смартфон у професійних цілях, то через 6 років цей показник перевищив 50%. Медичний інформаційний онлайн-сервіс Health24 зберігає дані у G-Cloud від De Novo.

De Novo, лідируючий оператор хмарних сервісів України, який забезпечує безпечне зберігання персональних даних користувачів медичного інформаційного онлайн-сервісу Health24. Використання хмарних ресурсів стало можливим завдяки інструментам забезпечення безпеки, які працюють в G-Cloud і є підтвердженими сертифікатом Державної служби спеціального зв'язку та захисту інформації України.

Будь ласка, перейдіть за посиланням <https://health24.life/index> . Зайдіть до довідкового центру та ознайомтесь з можливостями системи.

2. Сутність хмарних технологій

Сутність хмарних технологій полягає в обміні даними в універсальному

вигляді без кодувань і перекодувань, зберіганні власних даних на віддалених носіях — міцних і надійних серверах або на тисячах чужих комп'ютерах (саме за таким принципом функціонують програми Cloud, Торрент, Dropbox та ін.), а також в обробці складних задач за допомогою мобільних програм. Поступово в «хмари» уходять історія хвороби, різноманітні зображення організму (рентгенограма, ЕКГ, тощо), лікарські консилиуми. Отже, основними напрямками використання хмарних технологій у роботі лікаря є:

- **перегляд медичних новин та журналів** (системи Google Reader, NewsRob). Бути в курсі останніх досягнень і новин медицини — один із пріоритетів для лікаря. На сьогодні в Інтернеті існує чимало ресурсів (наприклад, Annals of Internal Medicine, JAMA, The New England Journal of Medicine, Archives of Internal Medicine, Medscape), що пропонують лікарю медичні подкасти — аудіо- чи відеоматеріали із записом виступів на певну тему. Для того, щоб прослухати та/або переглянути подкасти, можна використовувати додаток Downcast;

- **оцінка активності захворювання, проведення підрахунків ризиків.** Вимірювання цих показників за допомогою спеціальних онлайн-калькуляторів уже стало повсякденною необхідністю. Для потреб ревматології існує додаток DAS (Disease Activity Score) Calc, який передбачає підрахунок активності захворювання з урахуванням швидкості осідання еритроцитів, рівня С-реактивного білка в сироватці крові, а також загального стану пацієнта. Використовувати такий додаток набагато зручніше, ніж проводити підрахунки самому, визначаючи, чи досягнута ремісія у конкретного пацієнта. Подібні калькулятори існують також і в кардіології, зокрема, для підрахунку кардіоваскулярного ризику за шкалою SCORE (Systematic Coronary Risk Evaluation);

- **розшифровка лабораторних даних.** Додаток Lab Gear містить дані про

граничні норми лабораторних показників, а також додаткову інформацію про вимірюваний показник та основні клінічні симптоми, що супроводжують захворювання, асоційовані зі зміною цього показника. Програмний додаток Medscape для смартфонів теж знадобиться лікарю-практику, оскільки містить зручний калькулятор для підрахунку близько 130 різних медичних формул;

- **ведення облікових записів.** Спеціальний додаток My Pain стане в нагоді спеціалістам, які спостерігають пацієнтів із хронічним больовим синдромом. Програма дозволяє записувати щоденні (а за необхідності — погодинні) дані щодо зміни рівня болю для складання графіка, а також враховувати супутні фактори, такі як тригери болю, причини його посилення або послаблення, характер і місце розташування болю;

- **отримання мобільного доступу до власних файлів.** У лікаря часто виникає потреба зберегти файл великого обсягу або отримати доступ до матеріалів, що зберігаються на домашньому чи робочому комп'ютері, з мобільного пристрою. Для цієї мети підходить Dropbox — сервіс, який дозволяє створити на ПК спеціальну папку, синхронізовану з web-сервером і будь-яким мобільним пристроєм. Відповідно, до файлів, збережених в Dropbox, можна отримати дистанційний доступ зі смартфона або іншого пристрою, що позбавляє лікаря необхідності багаторазово пересилати файли по e-mail або навіть записувати матеріали великого обсягу (наприклад, презентації, відео, статті у форматі pdf) на флеш-накопичувачі. Dropbox розпізнає практично всі типи файлів, у тому числі pages, key, numbers, pdf, doc, xls тощо. Суттєво скоротити обсяг паперових записів допомагає оптимізований для текстів сервіс Evernote, у якому зручно зберігати, наприклад, рецепти, різні підтвердження та повідомлення, отримані по e-mail. Щоб зберегти багатосторінковий pdf-файл зі статтею з медичного журналу, зручно використовувати Genius Scan з Evernote, оскільки при отриманні зображень з багатосторінкового документа Evernote зберігає кожен сторінку у вигляді окремого файлу, а Genius Scan об'єднує їх у

цілісний документ для подальшої відправки в Evernote або Dropbox;

- **оптимізація тайм-менеджменту.** Розклад прийому пацієнтів і зустрічей, нагадування про наради й обстеження — це великий обсяг інформації, який лікарю доводиться систематизувати щодня, аби дотримуватися робочого ритму. Існує безліч додатків, у тому числі безкоштовна програма Wunderlist, які дозволяють створювати нотатки, встановлювати ієрархію папок, визначаючи пріоритети, нагадування (у тому числі голосові) на необхідну дату.

Експерти відмічають, що використання описаних програмних додатків істотно полегшить роботу лікаря, позбавивши його великого обсягу паперової роботи.

3. Хмарні технології: історія появи поняття

Останні роки все більшої популярності набувають так звані хмарні технології або хмарні обчислення (Cloud computing). Цей термін став вживатися в світі інформаційних технологій з 2008 року. Першою людиною, який виголосив словосполучення «cloud computing» був Ерік Шмідт - генеральний директор компанії Google.

Термін **"хмарні"** (від англ. "cloud technology") - не зовсім влучний. Дослівний переклад слова "cloud" - "хмара" (звідси «хмарні технології»), однак це ж слово перекладається як "розсіяний, розподілений". Тож хмарні технології є "розподіленими технологіями", тобто опрацювання даних відбувається з використанням не одного стаціонарного комп'ютера, а розподіляється по комп'ютерах, підключених до Internet.

«Хмарні» технології— це парадигма, що передбачає віддалену обробку та зберігання даних; надають користувачам Інтернету доступ до комп'ютерних ресурсів серверу і використання програмного забезпечення як онлайн-сервісу; дозволяють споживачам використовувати програми без установки і забезпечують доступ до особистих файлів з будь-якого комп'ютера, що має доступ в Інтернет.

Суть концепції «хмарних обчислень» полягає в наданні кінцевим користувачам віддаленого динамічного доступу до послуг, обчислювальних ресурсів і додатків (у тому числі до операційних систем та інфраструктури) через Інтернет. «Хмарні обчислення» являють собою масштабований спосіб доступу до зовнішніх обчислювальних ресурсів у вигляді сервісу, що надається за допомогою Інтернету, при цьому користувачеві не потрібно ніяких особливих знань про інфраструктуру «хмари» або навичок управління цією «хмарною» технологією.

Отже, *хмарні технології* — це технологія, яка надає користувачам Інтернету доступ до комп'ютерних ресурсів сервера і використання програмного забезпечення як онлайн-сервіса, тобто якщо, є підключення до Інтернету то можна виконувати складні обчислення, опрацьовувати дані використовуючи потужності віддаленого сервера.

«*Хмара*» — це деякий ЦОД (дата-центр, сервер) або їх мережа, де зберігаються дані та програми, що з'єднуються з користувачами через Інтернет.

Найбільш поширеними сховищами даних у хмарах є:

Google Диск надає 15 ГБ (разом з поштою) місця на своєму диску.

Microsoft SkyDrive – 7 ГБ.

Dropbox – 2 ГБ (безкоштовно можна більшувати до 16 ГБ).

Mega – представляє безкоштовно 50 ГБ дискового простору.

eDisk – це доступне з будь-якої точки Землі персональне сховище файлів. В цьому сховищі можна зберігати до 4 ГБайт інформації (близько 40 000 документів). <https://www.youtube.com/watch?v=A4UFeHzX5ko> .

4. Моделі розгортання хмар

Приватна хмара (англ. private cloud) - це хмарна інфраструктура, яка призначена для використання виключно однією організацією, що включає декілька користувачів (наприклад, підрозділів). Приватна хмара може перебувати

у власності, керуванні та експлуатації як самої організації, так і третьої сторони (чи деякої їх комбінації). Така хмара може фізично знаходитись як в, так і поза юрисдикцією власника.

Громадська хмара (англ. community cloud) - це хмарна інфраструктура, яка призначена для використання конкретною спільнотою споживачів із організацій, що мають спільні цілі (наприклад, місію, вимоги щодо безпеки, політику та відповідність різноманітним вимогам). Громадська хмара може перебувати у спільній власності, керуванні та експлуат

ації однієї чи більше організацій зі спільноти або третьої сторони (чи деякої їх комбінації). Така хмара може фізично знаходитись як в, так і поза юрисдикцією власника.

Публічна хмара (англ. public cloud) - це хмарна інфраструктура, яка призначена для вільного використання широким загалом. Публічна хмара може перебувати у власності, керуванні та експлуатації комерційних, академічних (освітніх та наукових) або державних організацій (чи будь-якої їх комбінації). Публічна хмара перебуває в юрисдикції постачальника хмарних послуг.

Гібридна хмара (англ. hybrid cloud) - це хмарна інфраструктура, що складається з двох або більше різних хмарних інфраструктур (приватних, громадських або публічних), які залишаються унікальними сутностями, але з'єднанні між собою стандартизованими або приватними технологіями, що уможливають переносимість даних та прикладних програм (наприклад, використання ресурсів публічної хмари для балансування навантаження між хмарами).

Суспільна хмара (англ. community cloud) — вид інфраструктури, призначений для використання конкретною спільнотою споживачів з організацій, що мають спільні завдання (наприклад, місії, вимоги безпеки, політики, та відповідності різним вимогам). Громадська хмара може перебувати в кооперативній (спільній) власності, управлінні та експлуатації однієї або

більше з організацій співтовариства або третьої сторони (або будь-якої їх комбінації), і вона може фізично існувати як всередині, так і поза юрисдикцією власника.

5. Моделі обслуговування хмар:

Виділяють наступні моделі надання послуг за допомогою хмари:

1. **Програмне забезпечення як послуга (SaaS)** Прикладами програмного забезпечення як послуги, що працює на основі обчислювальної хмари, є сервіси Gmail та Google docs.

2. **Платформа як послуга (PaaS)** Наприклад, Google Apps надає застосунки для бізнесу в режимі онлайн, доступ до яких відбувається за допомогою Інтернет-браузера тоді як ПЗ і дані зберігаються на серверах Google.

3. **Інфраструктура як послуга (IaaS)** Найбільшими гравцями на ринку інфраструктури як послуги є Amazon, Microsoft, VMWare, Rackspace та Red Hat. Хоча деякі з них пропонують більше, ніж просто інфраструктуру, їх об'єднує мета продавати базові обчислювальні ресурси.

Загальною характеристикою компаній, що будують свої продукти на основі хмар, є впевненість у тому, що мережа інтернет в змозі задовольнити потреби користувачів в обробці даних.

Найголовнішою функцією хмарних технологій є задоволення потреб користувачів, що потребують віддаленої обробки даних.

Переваги хмарних технологій:

- забезпечення повсякчасного та зручного мережевого доступу за вимогою;
- не потрібні потужні комп'ютери;
- менше витрат на закупівлю програмного забезпечення і його систематичне оновлення;

- економія коштів на утримання технічних фахівців;
- необмежений обсяг збереження даних;
- доступність з різних пристроїв і відсутня прив'язка до робочого місця;
- забезпечення захисту даних від втрат та виконання багатьох видів навчальної діяльності, контролю і оцінювання, тестування он-лайн, відкритості освітнього середовища;
- робота із інформацією з різних пристроїв;
- легко ділитися інформацією з людьми з будь-якої точки світу за допомогою ПК, планшета, телефона;
- одночасна робота з даними з різних пристроїв декількох людей;
- веб-сервіси працюють з браузерами будь-яких ОС;
- завжди нові версії програм.

Необхідні компоненти для роботи в «хмарах»

Обов'язково:

- Підключення до мережі Інтернет
- Комп'ютер (планшет, смартфон, нетбук)
- Веб-браузер

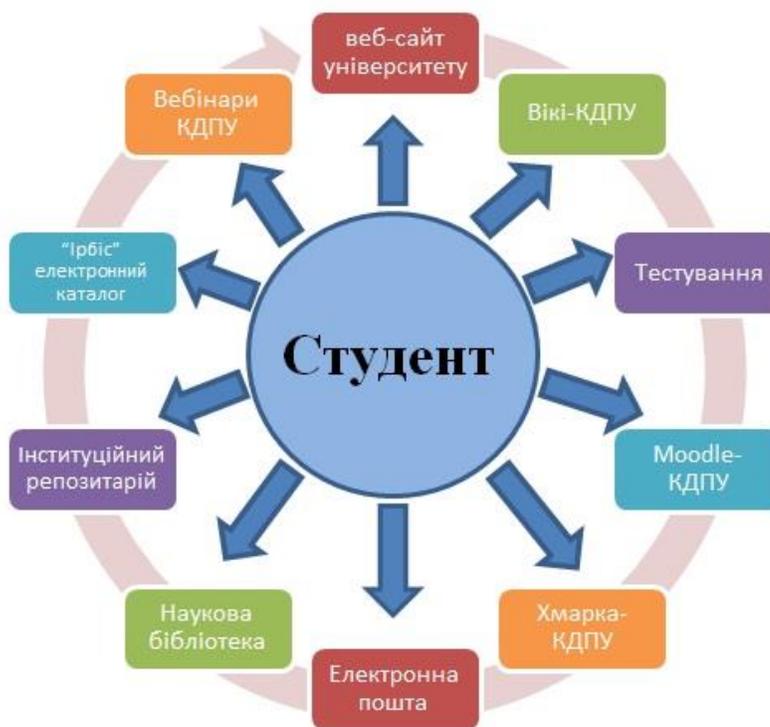
Бажано:

- Компанія, яка надає послуги хмарних технологій
- Навички роботи з Інтернет.

6. Хмарні технології для навчання студентів

Система хмарних технологій навчання предмета, складається з загально-навчальних хмарних технологій (технології он-лайн-розробки та он-лайн-сховищ електронних навчальних матеріалів, технології управління навчанням) та вузькоспеціалізованих хмарних технологій – браузерні системи програмування та моделювання (на підтримку вивчення інформатики), мобільні математичні

середовища (на підтримку вивчення математики), віртуальні он-лайн-лабораторії та системи моделювання (на підтримку вивчення фізики, хімії або біології).



Приклади використання хмарних технологій у навчанні студентів :

- 1) використання Office Web Apps-додатків;
- 2) електронні журнали і щоденники;
- 3) он-лайн сервіси для учбового процесу, спілкування, тестування;
- 4) системи дистанційного навчання, бібліотека, медіатека;
- 5) сховища файлів, спільний доступ;
- 6) спільна робота;
- 7) відеоконференції;
- 8) електронна пошта з доменом вищого навчального закладу.

Microsoft Office 365 — це набір програм, що базується на хмарних технологіях і включає в себе безкоштовну електронну пошту, службу обміну миттєвими повідомленнями, засіб проведення відеоконференцій і здійснення голосових викликів, а також дозволяє створювати і редагувати документи в

онлайн. Хмарний формат означає, що всі дані зберігаються в центрі обробки даних Microsoft, а не на комп'ютері користувача, і це забезпечує користувачам доступ до документів і даних з різних пристроїв через Інтернет з допомогою браузера.

До складу Microsoft Office 365 входять служби Microsoft Lync Online, Microsoft SharePoint Online, а також остання версія набору програм Microsoft Office Professional Plus . Детальніше перегляньте інформацію за посиланням: https://www.vostok.dp.ua/ukr/infa1/program/office_365/



До базових функцій Microsoft Lync належать такі види комунікації, як передавання миттєвих повідомлень, голосова та відеотелефонія. Розширені функції, що їх надає засіб, - це результат інтеграції з іншими програмами Microsoft: використання контактів Microsoft Outlook; доступність контактів служби локальних директорій, наприклад, Microsoft Exchange Server; організація мережних нарад в Microsoft Outlook; можливість бачити усіх людей, що працюють з певним документом Microsoft Office.



Microsoft SharePoint являє собою колекцію програмних продуктів та компонентів, яка є базовою інфраструктурою для спільної роботи, платформою для

документообігу за рахунок створення «маршрутів» руху документів; пошуку інформації в документах та інформаційних системах; аналізу та управління робочими процесами.

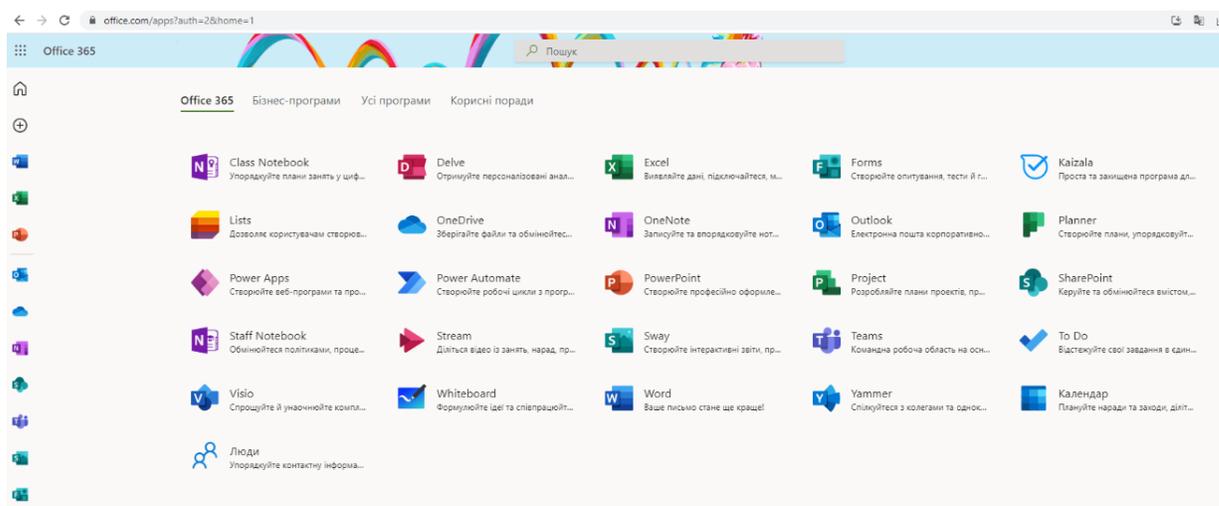


Створювані на платформі SharePoint сайти можуть бути використані в якості сховища інформації, знань і документів, а також використовуватися для виконання веб-зв'язків, що полегшують взаємодію. Користувачі можуть керувати і взаємодіяти з інформацією в списках і бібліотеках документів, використовуючи елементи

управління, які зветься веб-частинами (SharePoint).

Microsoft Office 365 — це набір програм, що базується на хмарних технологіях і включає в себе безкоштовну електронну пошту, службу обміну миттєвими повідомленнями, засіб проведення відеоконференцій і здійснення голосових викликів, а також дозволяє створювати і редагувати документи в онлайн режимі. Хмарний формат означає, що всі дані зберігаються в центрі обробки даних Microsoft, а не на комп'ютері користувача, і це забезпечує користувачам доступ до документів і даних з різних пристроїв через Інтернет з допомогою браузера.

До складу Microsoft Office 365 входять служби Microsoft - SharePoint, Forms, Outlook, Visio, Teams, а також остання версія набору програм Microsoft - Word, Excel, Power Point та ще багато інших додатків.



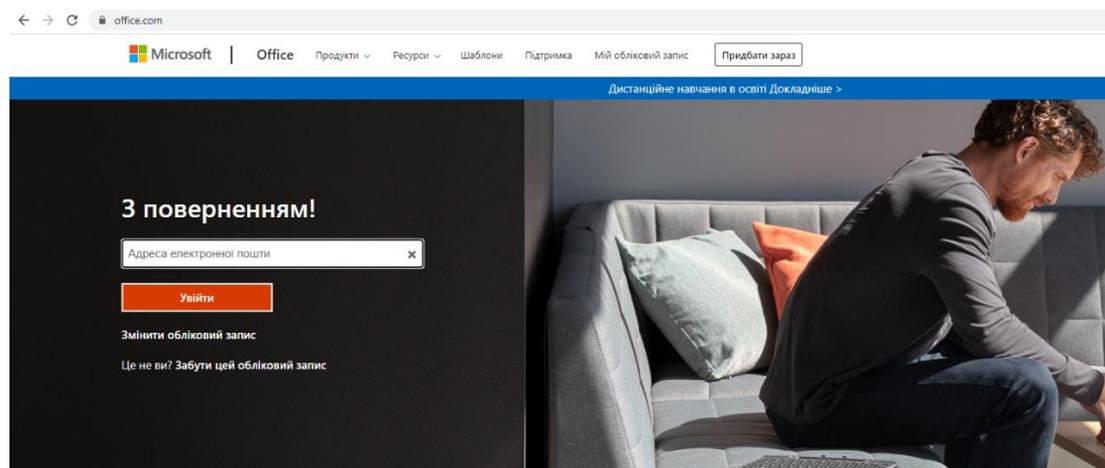
Практичні вправи для знайомства з MS Office 365

В ході виконання практичної роботи студенти повинні навчитися: здійснювати вхід до Office365, використовуючи свій обліковий запис; використовувати пошту для обміну повідомленнями; створювати контакти за інтересами; створювати плани для сумісної роботи із використанням календаря, додавати сповіщення; використовувати можливості OneDrive та SharePoint;

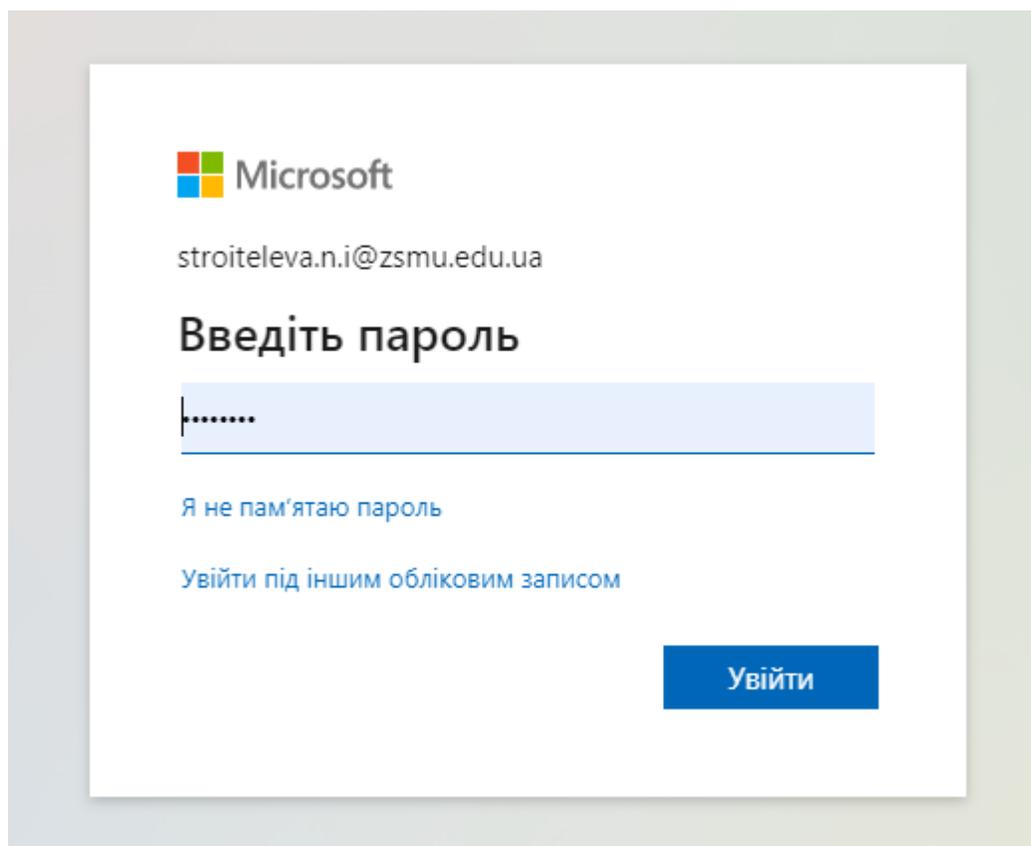
працювати з додатком OneNote.

1. Вхід <https://login.microsoftonline.com>

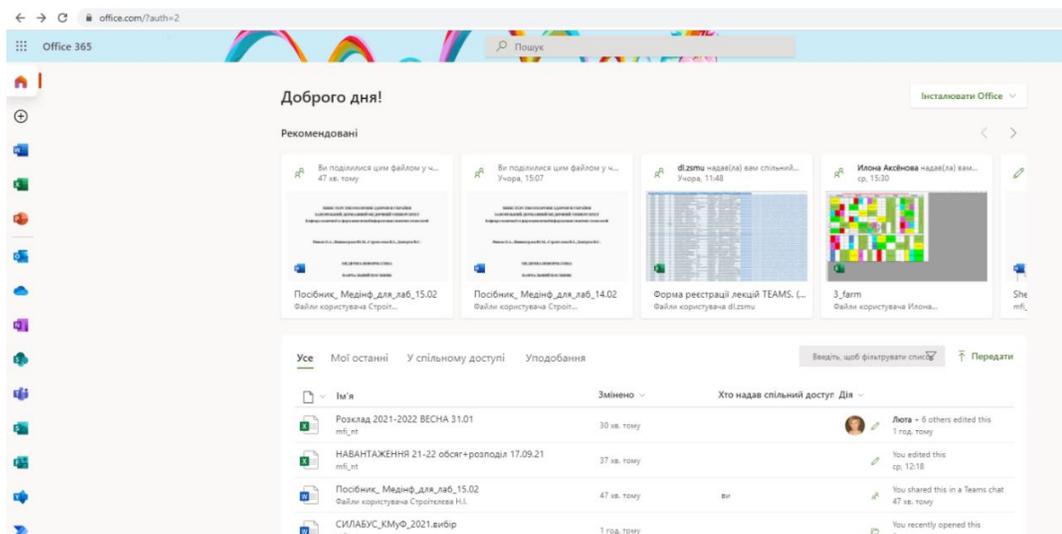
Заповніть, будь ласка, обліковий запис. Якщо Ви забули логін або пароль, зверніться до викладача, який допоможе Вам з цим.



Підписатися на Office [in](#) [f](#) [t](#) [@](#) [✉](#)



При зміні облікового запису Вам необхідно, використовуючи одноразовий пароль, увійти до системи та обрати собі новий пароль. При цьому враховуйте на зауваження, які Вам надає система, якщо пароль не відповідає вимогам конфіденційності або іншим вимогам з позицій системи.



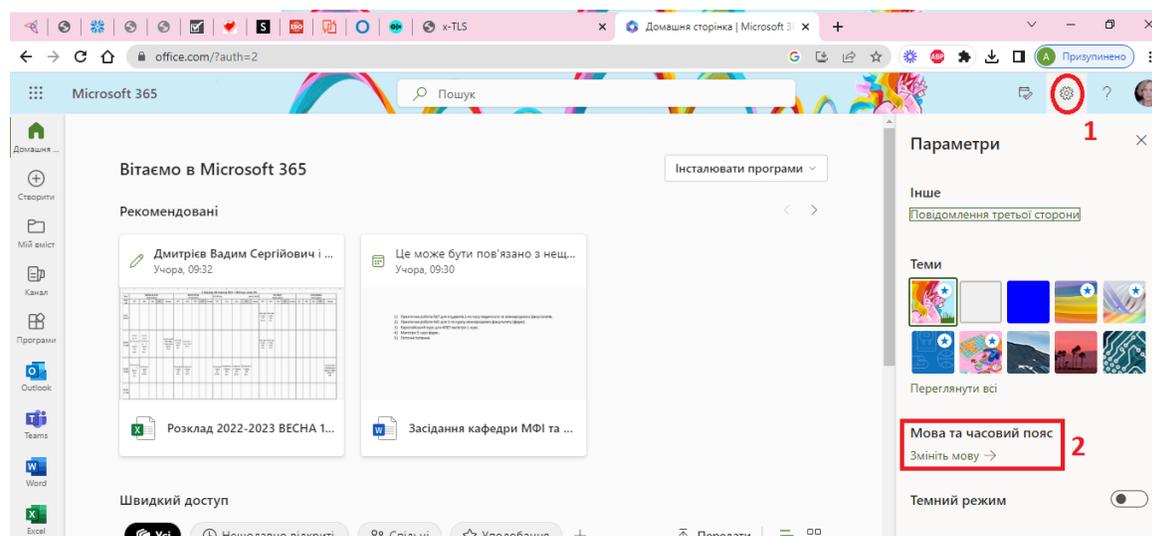
2. Використання Outlook

Однією з переваг Microsoft Office 365 є те, що для роботи з електронною поштою користувачам пропонується програма Outlook — поштовий клієнт корпоративного рівня. Наявність зазначеної програми є відмінною особливістю Microsoft Office 365, що додає цій платформі зручності, оскільки Outlook має більше функцій, ніж звичайний веб-інтерфейс для роботи з електронною поштою.

Продукти MS Office 365 - це набір безкоштовних веб-додатків – текстового і табличного процесора, менеджера презентацій і редактора нотаток. Для зберігання файлів користувачеві видається до 4 Тб безкоштовного дискового простору для необмеженого за часом зберігання створених файлів з можливістю поділу до них спільного доступу, сортування по папках, а також завантаження на локальний комп'ютер. Документи з урахуванням обмеженості функцій інструментів, перенесених у веб-інтерфейси, можна редагувати, а також

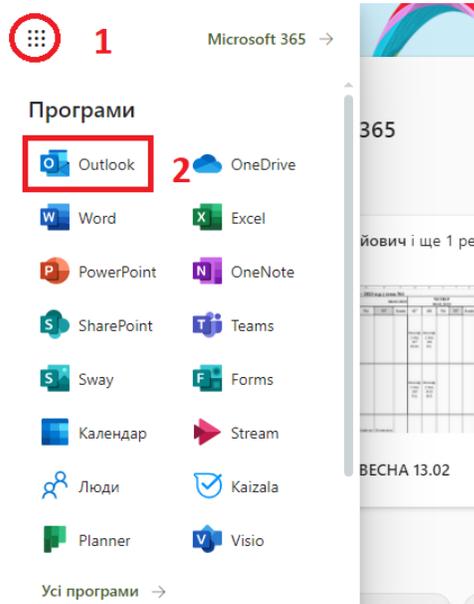
друкувати з браузера. Крім того, для будь-якого з файлів, що знаходиться на Windows Live SkyDrive, можна запустити відповідний додаток у складі настільної версії офісного пакету.

1. Оберіть мову та часовий пояс як це показано на наступному малюнку.

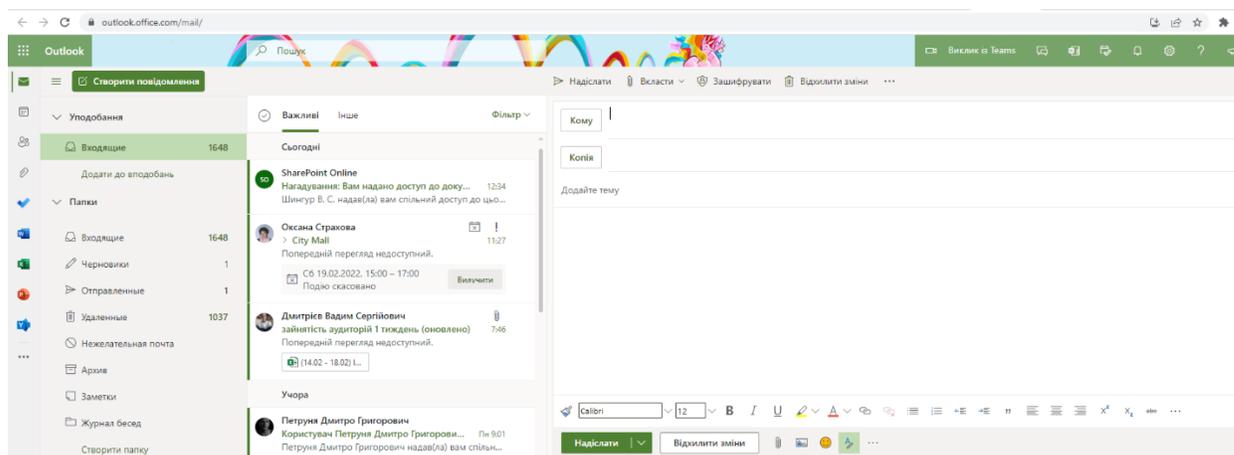


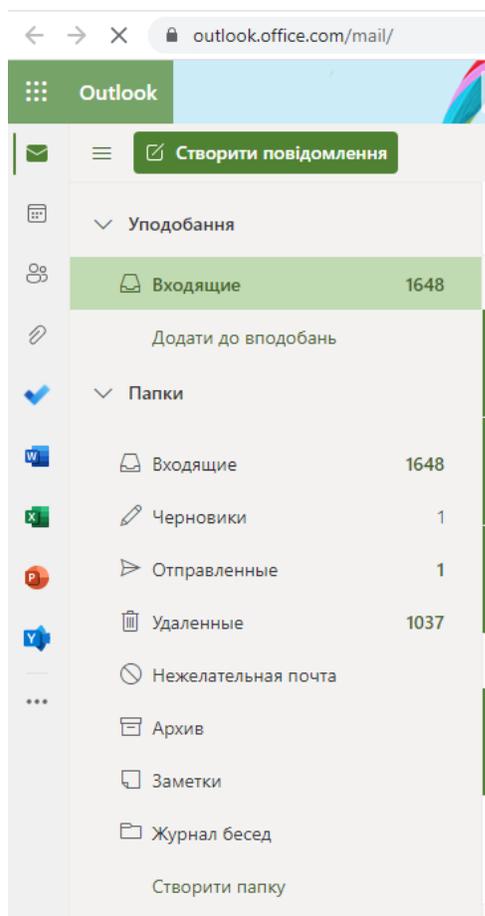
2. Увійдіть до сторінки з Outlook, яка відкривається після клацання посилання Outlook (ПОШТА) на головній сторінці Office 365.

За допомогою цієї програми ви зможете отримати доступ до поштової скриньки через браузер із будь-якого комп'ютера, підключеного до Інтернету, а також читати і надсилати повідомлення електронної пошти, упорядковувати контакти, створювати завдання та керувати календарем.



Ви увійшли до Outlook. Роздивіться уважно інтерфейс.





У програмі Outlook ви можете виконувати всі операції з керування електронною поштою: отримувати і читати листи, надавати відповіді на них, видаляти листи та впорядковувати їх, розміщуючи у відповідних папках. Ви також маєте можливість додавати до листів файли, запитувати сповіщення про їх читання або доставку, призначати для листів категорії, за якими згодом можна буде фільтрувати повідомлення.

Створення і відправка повідомлення

Відкрийте Outlook і перейдіть до розділу Пошта. Виберіть команду Створити>Повідомлення. Клацніть кнопку «Кому» і вкажіть одержувачів повідомлення з списку контактів групи, яку створив викладач вашого навчального закладу.

Для вибору одержувача з власного списку контактів клацніть ліворуч команду Контакти, виберіть контакт і клацніть кнопку Кому. Клацніть кнопку

ОК, щоб вийти з адресної книги.

Уведіть текст повідомлення, за потреби, вкладіть у нього файл і встановіть сповіщення про читання або доставку. Надішліть повідомлення, клацнувши кнопку Надіслати.

Обмін миттєвими повідомленнями

З допомогою Outlook можна спілкуватися з будь-ким із користувачів програми — так само, як у Skype, Windows Messenger та інших службах миттєвих повідомлень.

Щоб уможливити таке спілкування, потрібно додати користувача до свого списку контактів служби миттєвих повідомлень.

Керування контактами

Контакти — це записи, що містять контактну інформацію про користувачів. Кілька контактів можна об'єднати в групу (список розсилки електронної пошти) — у такому разі з'явиться можливість, вказавши ім'я групи, надіслати повідомлення електронної пошти одночасно всім її учасникам.

Зберігаються контакти і групи в адресній книзі. У системі Outlook Web App користувачеві доступні дві адресні книги: одна, глобальна, міститиме усі контакти і групи домену університету (її зазвичай створює адміністратор домену, викладач), а інша — особисті контакти (ви створюєте її власноруч у папці Контакти).

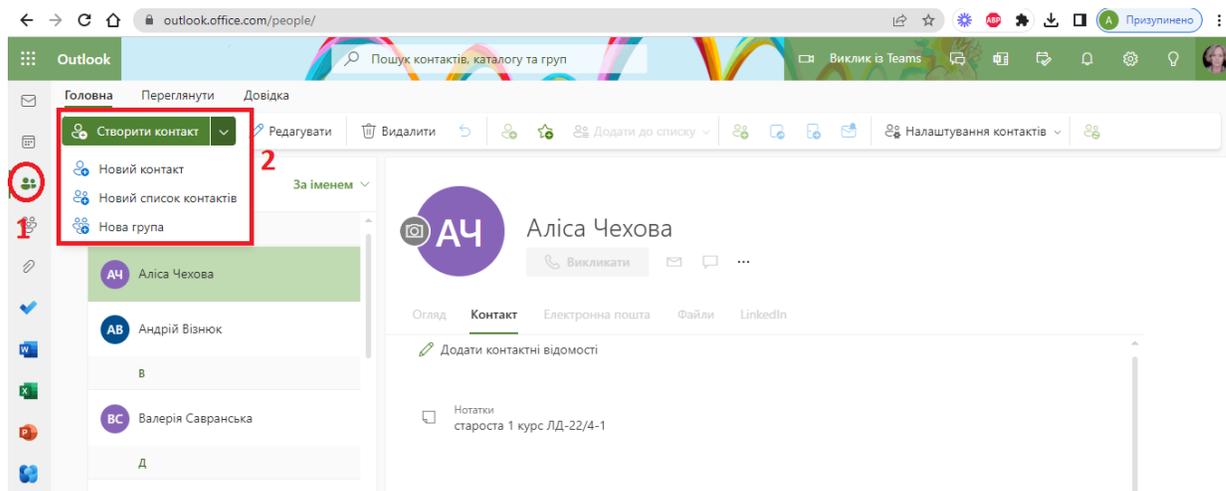
Створення контакту

Відкрийте Outlook. Перейдіть до розділу Контакти і клацніть кнопку, розташовану ліворуч від поля Пошук користувачів. Ви потрапите до адресної книги зі списком контактів усього домену вашого навчального закладу.

Якщо потрібно додати контакт із цього списку у власну адресну книгу, клацніть контакт правою кнопкою миші, виберіть команду Додати до контактів, заповніть поля у формі та клацніть кнопку Зберегти й закрити. Потім клацніть

кнопку закриття адресної книги. Обраний контакт відобразиться серед ваших контактів.

Щоб створити контакт у власній адресній книзі, виберіть команду Створити>Контакт, уведіть усі необхідні дані і клацніть кнопку Зберегти й закрити.



Робота з календарем і завданнями

Програма Outlook— не просто поштовий клієнт, а інструмент, який допоможе вам керувати своїми справами. Для цього у вашому розпорядженні буде календар, що дозволяє планувати зустрічі й зібрання, та окремий засіб для керування завданнями.

Календар Outlook, який відображається після клацання команди Календар у лівому нижньому куті вікна Outlook, більше орієнтований на застосування у робочому процесі, а тому дещо відрізняється від календаря Windows Live.

Зокрема, у ньому немає такого об'єкта, як день народження, і не створюється календар державних свят. У той час як у календарі Windows Live є три подання — день, тиждень і місяць, у календарі Outlook їх чотири — день, робочий тиждень, тиждень і місяць. Для робочого календаря можна задавати відображення тих днів, у які у вас є заняття. У календарі Windows Live можна

створювати події, справи і дні народження, а у календарі Outlook — лише зустрічі та запрошення на зібрання; засоби для керування завданнями містяться не в календарі, а окремо.

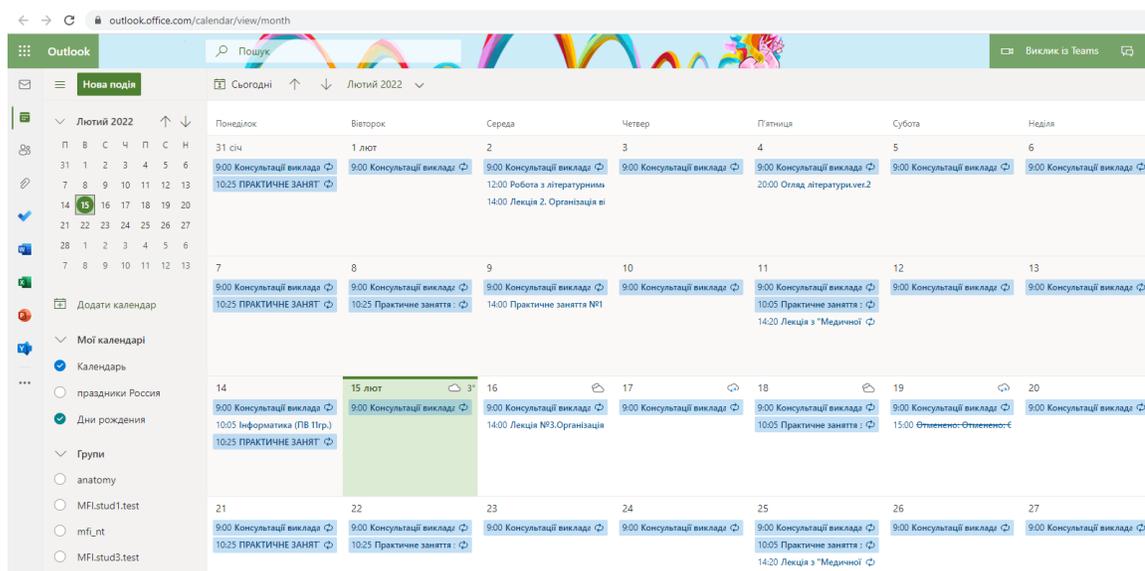
Можна відстежувати різні аспекти завдань, зокрема дати початку і завершення його виконання, важливість, ступінь виконання тощо. Якщо завдання потрібно виконувати по кілька разів, його можна зробити повторюваним.

Надання спільного доступу до свого календаря

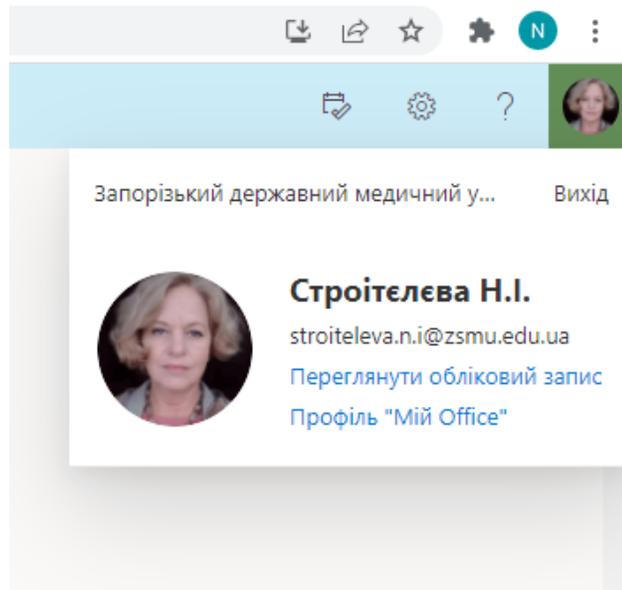
Ви можете надати спільний доступ до календаря будь-якому контакту з глобальної або власної адресної книги. Перейдіть до розділу Календар. Відкрийте меню Спільний доступ і виконайте команду Надати спільний доступ до календаря. Вкажіть контакти, яким хочете надати спільний доступ.

Вкажіть, які дані мають бути доступними іншим користувачам під час перегляду вашого календаря, і клацніть кнопку «Надіслати».

Увійдіть до розділу «Календар» та додайте розклад своїх занять на місяць з вказанням часу та аудиторії.



3. Увійдіть до розділу «Мій обліковий запис» та змініть фото свого профілю, використовуючи фото, яке є на вашому телефоні.

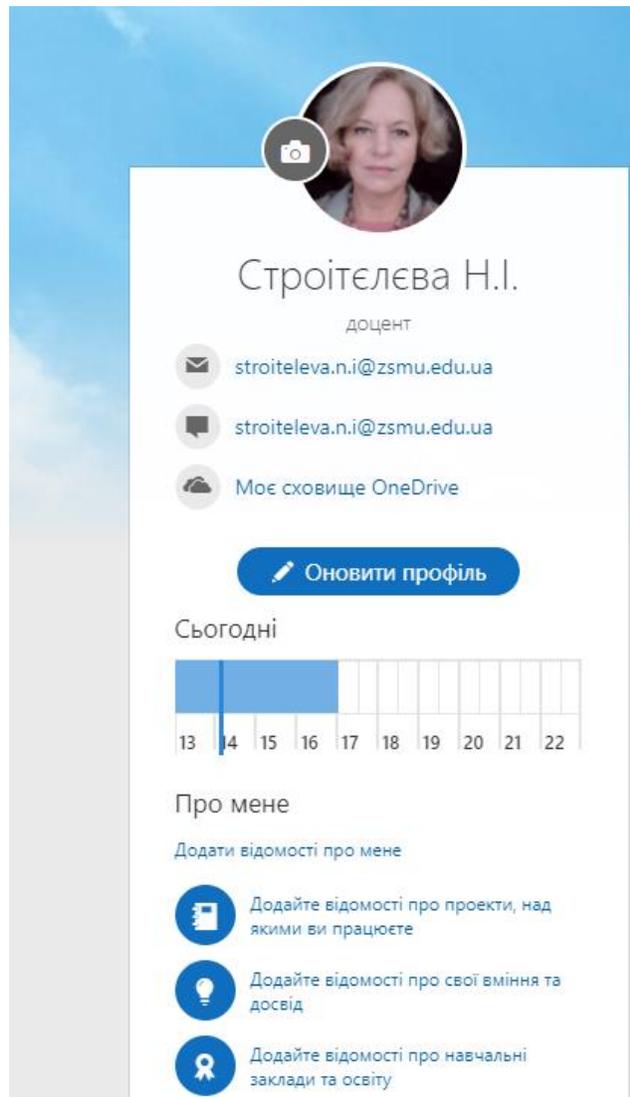


4. Зберігання матеріалів у службі OneDrive

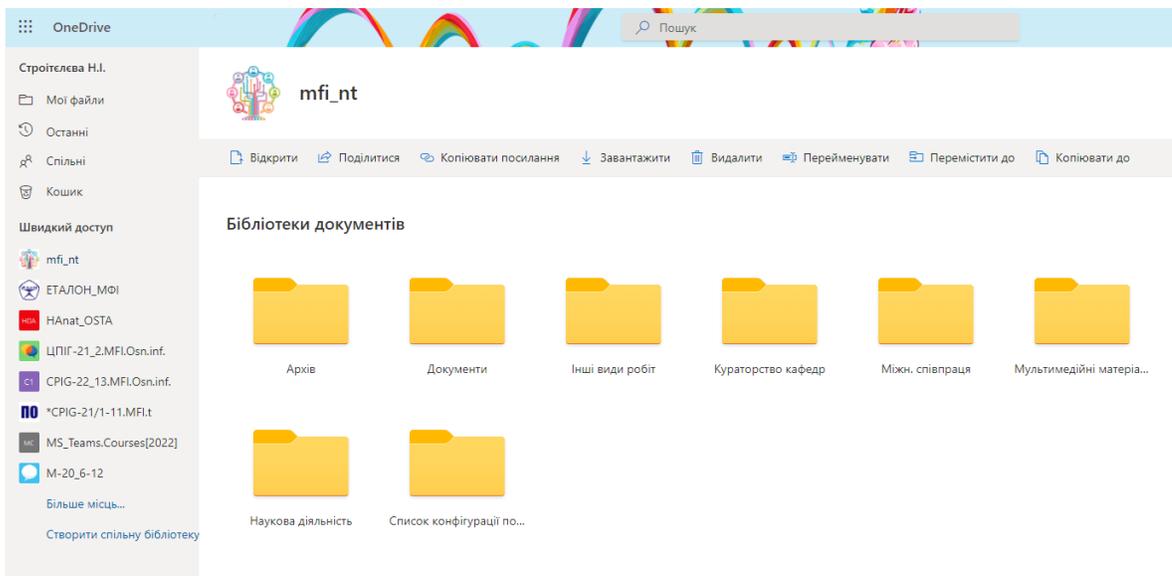
OneDrive — це безкоштовне інтернет-сховище даних. Зберігаючи тут свої файли, ви отримуєте низку переваг:

- відпадає необхідність прив'язуватися до одного комп'ютера;
- не потрібно носити з собою флешки або компакт-диски;
- залишається вільним місце на жорсткому диску комп'ютера;

Для впорядкування файлів можна використовувати папки різних рівнів вкладеності. До кожної з них ви можете призначати різні права доступу.

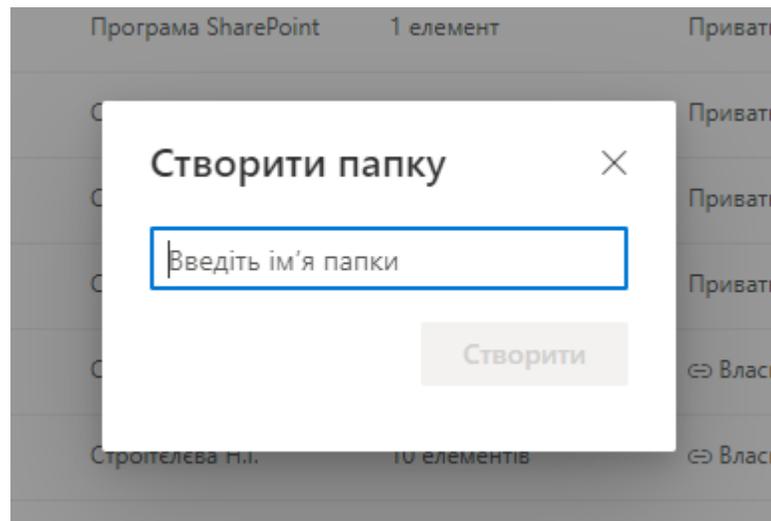


Увійдіть до розділу OneDrive. Клацніть рядок папки, до якої хочете надати спільний доступ, а потім клацніть посилання «Змінити дозволи». Встановіть дозвіл «Для всіх» і клацніть кнопку «Зберегти». Тепер ви можете розмістити посилання на папку у блозі або надіслати його електронною поштою.

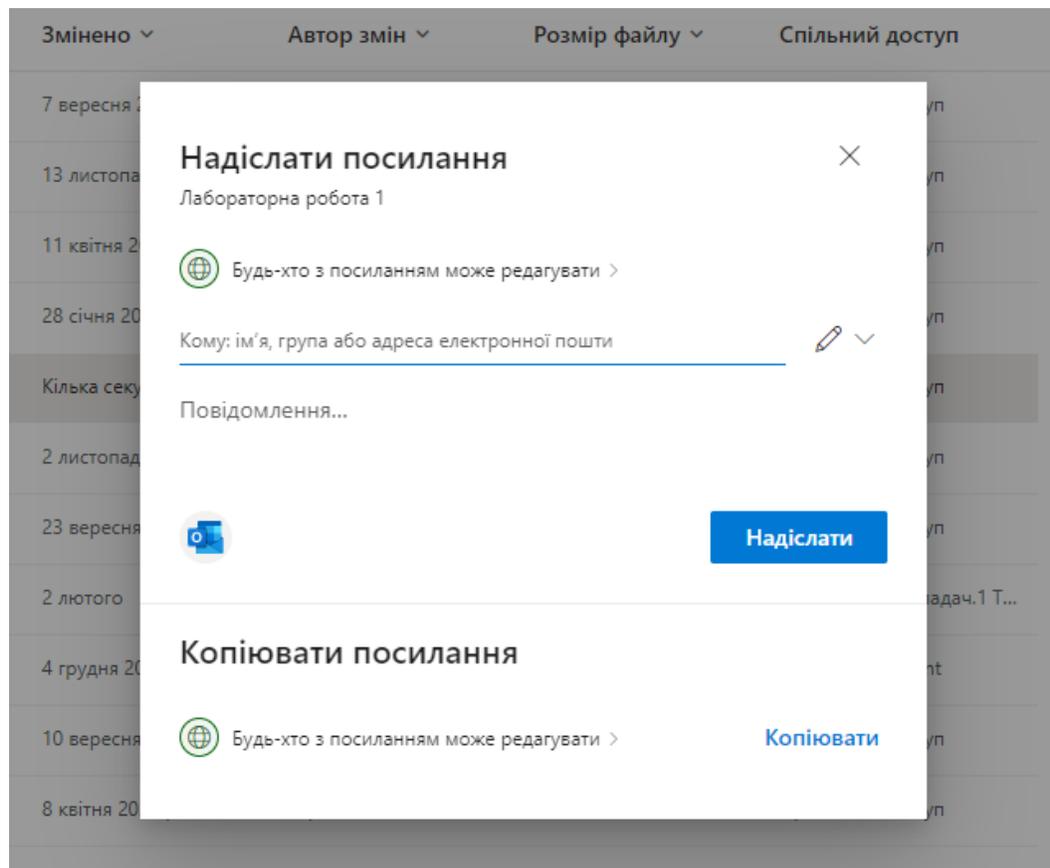


Створіть нову папку та надішліть посилання на неї іншим користувачам.

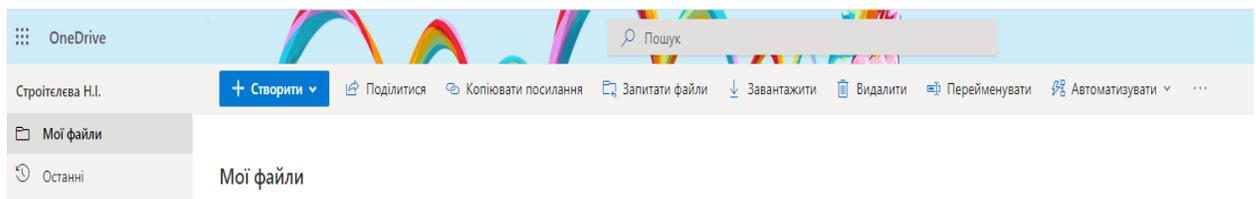
У рядку меню клацніть на «+Створити», оберіть Папку за дайте їй нове ім'я, наприклад, «ПРАКТИЧНЕ ЗАНЯТТЯ1».



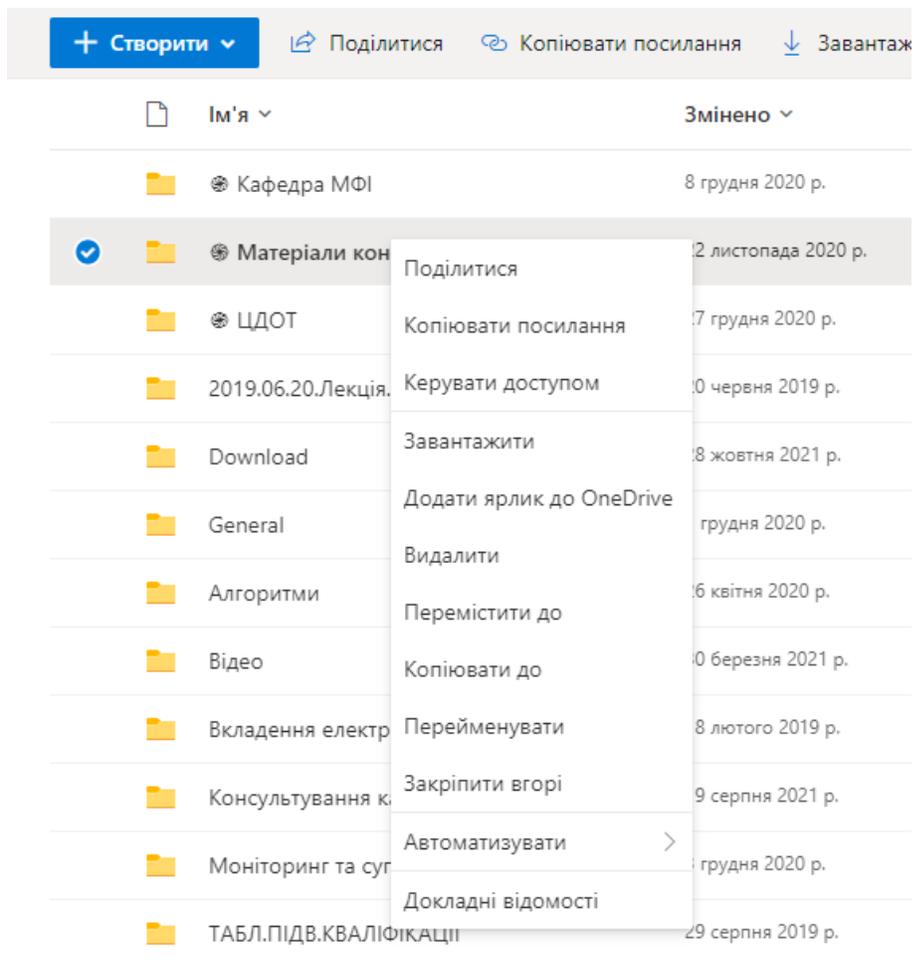
Оберіть з меню «Поділитись» опцію «Надіслати посилання» та вкажіть електронну адресу отримувача, якому ви надасте можливість спільного доступу до папки «Лабораторна робота 1».



В пункті меню «Поділитися» ви можете обрати параметри посилання, тобто вибрати яким саме користувачам відправляти посилання на папку «ПРАКТИЧНЕ ЗАНЯТТЯ1»



За допомогою контекстного меню для папки «ПРАКТИЧНЕ ЗАНЯТТЯ1» ви можете задати загальний доступ до цієї папки.



5. Відкрийте MS Teams та подивіться, яке завдання видав викладач у розділі «Завдання», зверніть увагу на строки виконання завдання.

Порядок виконання роботи:

1. Зайти на сайт Microsoft за адресою <http://portal.office.com> під своїми реєстраційними даними.
2. Ознайомитись з можливостями хмарних сервісів пакету Office365.34. Увійти в параметри системи, обрати українську мову інтерфейсу (зробити скріншот).
3. Зробити селфі та додати фото у свій профіль в MS Office 365. (зробити скріншот)
4. У хмарному додатку Outlook виконати налаштування інтерфейсу електронної

пошти , створити контакти (зробити скріншот) та здійснити обмін повідомленнями з одногрупниками. (зробити скріншот)

5. У хмарному додатку One Drive створити нові каталоги та розмістити в них методичні матеріали з навчальних дисциплін поточного семестру. (зробити скріншот).

6. Зайти у Teams під своїм логіном та паролем.

7. Через мобільний додаток MS Teams створити декілька персональних чатів з одногрупниками чи викладачем (зробити скріншот).

8. У хмарному додатку Календар придумайте на наступні два вихідних дня заплануйте самостійно будь-які події (лекції, практичні заняття, самостійна робота, тощо) із додаванням початку та кінця події, а також місця її проведення. (зробити скріншот)

9. Зроблені скріншоти додати в документ Word, зберегти з ім'ям, що співпадає з прізвищем студента-виконавця та номером практичної роботи (наприклад, Петренко_2.doc)

10. Створений документ WORD відправити викладачеві прикріпивши його до завдання.

Питання для самоконтролю

1. Дайте характеристики різновидам хмарних інфраструктур.
2. Поясніть поняття «хмарні» технології.
3. Визначте поняття «хмарний формат».
4. Назвіть правильну послідовність дій, які потрібно виконати, щоб надати деяким іншим користувачам доступ до Вашого календаря.
5. Назвіть необхідні компоненти для роботи в «хмарах».

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ №2

Тема: Архітектура сучасних комп'ютерних засобів. Основні функціональні вузли (компоненти) та принципи їх взаємодії. Різновиди операційних систем.

Мета роботи: робота з інформаційними ресурсами локальної мережі.

Теоретичні відомості

1. Архітектура сучасних комп'ютерних засобів

Найважливішим вузлом комп'ютера є *системна (system board)* або материнська (motherboard) *плата*. Як справжня мати вона "дає життя" всім іншим компонентам комп'ютера, постійно "годує" їх необхідним живленням, завжди готова "вислухати" кожен з підключених пристроїв - всі ці функції, імовірно, і привели до такої назви. Материнська плата вважається самим головним компонентом будь-якого комп'ютера - вона поєднує всі інші компоненти в єдине ціле й забезпечує їхню погоджену роботу. Отже, материнська плата - це найбільша друкована плата в системному блоці, що несе на собі головні компоненти комп'ютерної системи: центральний процесор, оперативну пам'ять, чіпсет, центральну магістраль (шину), контролер шини й велику кількість різних роз'ємів.

Найсучасніші системні плати містять наступні компоненти:

- гніздо для процесора;
- набір мікросхем системної логіки - чіпсет;
- базову систему уведення-виводу (ROM BIOS);
- гнізда модулів пам'яті;
- перетворювач напруги для центрального процесора;
- роз'єми шин;
- батарею.

Деякі системні плати також включають інтегровані аудіо- і відеоадаптери, мережний і SCSI- інтерфейси, а також інші елементи, залежно від типу системної плати.

Для підключення компонентів комп'ютера до системної плати використовують два способи: з'єднання пайкою та за допомогою роз'ємів. Пайка застосовується для тих компонентів, які є обов'язковими для роботи комп'ютера й жодним чином не призначені для заміни на аналогічні компоненти. Це, наприклад, мікросхеми тактового генератора, портів уведення/виводу і т.д.

"Мозком" персонального комп'ютера є мікропроцесор, або центральний процесор (ЦП). Він виконує обчислення й обробку даних і, як правило, є самою дорогою мікросхемою комп'ютера. У всіх PC-сумісних комп'ютерах використовуються процесори, що підтримують сімейство мікросхем Intel, але випускаються й проектуються вони не тільки самою Intel, але й компаніями AMD, Cyrix, IDT, Rise Technologies та інші. Основні характеристики ЦП:

- фірма - виробник;
- тип (CPU type) і тактова частота (CPU clock);
- напруга живлення (CPU voltage);
- тип корпусу

Чинсет здійснює з'єднання процесора з різними компонентами комп'ютера. Процесор не може взаємодіяти з пам'яттю, платами адаптера й різних пристроїв без допомоги цього набору мікросхем. Якщо скористатися медичною термінологією й зрівняти процесор з головним мозком, то набір мікросхем системної логіки по праву займе місце хребта й центральної нервової системи. Оскільки набір МС управляє інтерфейсом або з'єднаннями процесора з різними компонентами комп'ютера, то він визначає тип і швидкодію використовуваного процесора, робочу частоту шини, швидкість, тип і обсяг пам'яті.

Мікросхема BIOS (Basic Input/Output System, базова система

уведення/виводу) являє собою енергонезалежний постійний запам'ятовувальний пристрій, у якому записані програми, що реалізують функції уведення/виводу, а також програма тестування всіх компонентів комп'ютера в момент включення живлення і ряд інших програм.

Зв'язок між всіма власними пристроями та пристроями, що підключаються до системної плати, виконують шини системної плати та логічні пристрої, розміщені в мікросхемах мікропроцесорного комплекту (чіпсету). Від архітектури цих елементів багато в чому залежить продуктивність комп'ютера. **Шина** уявляє собою набір провідників на системній платі, або у вигляді кабелів. За час, що пройшов після появи першого ПК, особливо за останні роки, було розроблено досить багато варіантів шин вводу-виводу. Пояснюється це просто: для підвищення продуктивності комп'ютера потрібна швидкодіюча шина вводу-виводу.



Рисунок 2.1 – П'ятирівнева організація пам'яті

Ієрархічна структура пам'яті комп'ютера є традиційним розв'язанням проблеми зберігання великої кількості даних. Вона зображена на рис. 2.1. На самому верху ієрархії перебувають регістри процесора. Доступ до регістрів

здійснюється швидше всього, далі йде кеш-пам'ять, об'єм якої зараз становить від 32 Кбайт до декількох мегабайт. Потім впливає оперативна пам'ять, яка в цей час може вміщати від 16 Мбайт до десятків гігабайтів. У міру просування за структурою зверху вниз зростають три параметри. По-перше, збільшується час доступу. По-друге, збільшується об'єм пам'яті.

Кеш-пам'ять – це швидкодіюча пам'ять, призначена для узгодження швидкості роботи повільних пристроїв з швидким процесором шляхом тимчасового зберігання програмного коду і даних. Звертання до вбудованої кеш-пам'яті відбуваються без станів очікування, оскільки її швидкодія відповідає можливостям процесора. Використання кеш-пам'яті зменшує традиційний недолік комп'ютера, який полягає у тому, що оперативна пам'ять працює більш повільно, ніж центральний процесор (так званий ефект "пляшкового горлечка"). Завдяки кеш-пам'яті процесору не доводиться чекати, поки чергова порція програмного коду або даних надійде з відносно повільної основної пам'яті, що приводить до відчутного підвищення продуктивності.

Оперативна пам'ять – це робоча область для процесора комп'ютера. У ній під час роботи зберігаються програми і дані. Оперативна пам'ять часто розглядається як тимчасове сховище, тому що дані і програми в ній зберігаються лише при включеному комп'ютері або до натиснення кнопки перезавантаження. Перед виключенням або натисненням цієї кнопки всі дані, що можуть змінюватися під час роботи, необхідно зберегти на пристрої, що запам'ятовує, який може зберігати інформацію постійно (жорсткий диск). При новому включенні живлення збережена інформація знов може бути завантажена в пам'ять.

Жорсткий диск – пристрій постійної пам'яті для збереження великих обсягів інформації на довгий час. Уявляє собою магнітний диск, основа якого виконана з твердого матеріалу та має феромагнітне покриття. У більшості ЕОМ виконує функцію енергонезалежного носія інформації з довільним доступом.

Ємність пам'яті вінчестера вимірюється у байтах.

Пристрої, що не входять до складу системного блоку, називають *периферійними*. Наведемо сучасні периферійні пристрої, що підключаються до комп'ютера:

- *клавіатура*- один з найважливіших пристроїв комп'ютера, що використовується для введення в систему команд і даних.

- *миша* - це маніпулятор кнопкового типу, який додає зручності в роботі з графічним інтерфейсом комп'ютера.

- *сканер* являє собою електронно-механічний пристрій, призначений для перекладу графічної інформації в цифровий вид, з метою редагування й виводу на друк. Сканери підрозділяють на планшетні, ручні, барабанні, сканери форм, штрих-сканери.

- *монітор* забезпечує інформаційний зв'язок між користувачем і комп'ютером. Можна обійтися без принтера, дисководів і плат розширення, але робота без монітора рівносильна роботі наосліп: не видні ні результати, ні команди, що вводяться із клавіатури. Основний параметр монітора - *частота регенерації*, яку також називають частотою вертикального розгорнення, у більшості моніторів приблизно становить 85 Гц, тобто зображення на екрані поновлюється 85 разів у секунду. Зниження частоти регенерації приводить до мерехтіння зображення, що дуже стомлює очі. Отже, чим вище частота регенерації, тим комфортніше себе почуває користувач. Дуже важливо, щоб частота регенерації, що може забезпечити монітор, відповідала частоті, на яку настроєний відеоадаптер. Якщо такої відповідності нема, зображення на екрані взагалі не з'явиться, а монітор може вийти з ладу.

- *принтер* є найпоширенішим периферійним пристроєм виводу (друку) інформації на паперові носії в комп'ютерній системі *Роздільна здатність або дозвіл монітора* – це розмір мінімальної деталі зображення, яку можна

розрізнити на екрані. Даний параметр характеризується кількістю елементів розкладання - пікселей - по горизонталі й вертикалі екрана. Чим більше кількість пікселей, тим більш детальне зображення формується на екрані.

2. Комп'ютерні мережі

Нині персональні комп'ютери в автономному режимі практично не використовуються, їх, як правило, об'єднують в комп'ютерні мережі. Комп'ютерна мережа - це сукупність комп'ютерів і телекомунікаційного устаткування, що забезпечує інформаційний обмін комп'ютерів в мережі. Основне призначення комп'ютерних мереж - забезпечення доступу до розподілених ресурсів. Комп'ютерні мережі стали логічним результатом еволюції комп'ютерних і телекомунікаційних технологій. З одного боку, вони є часткою випадком розподілених комп'ютерних систем, а з іншого боку, можуть розглядатися як засіб передачі інформації на великі відстані.

Класифікуючи мережі за територіальною ознакою, розрізняють :

- глобальні (WAN),
- локальні (LAN),
- міські (MAN) мережі.

Хронологічно першими з'явилися мережі WAN. Вони об'єднують комп'ютери, розосереджені на відстані сотень і тисяч кілометрів. Перші глобальні комп'ютерні мережі дуже багато що успадкували від телефонних мереж. У них часто використовувалися вже існуючі і не дуже якісні лінії зв'язку, що призводило до низьких швидкостей передачі даних і обмежувало набір послуг, що надавалися, передачею файлів у фоновому режимі і електронною поштою. Мережі LAN обмежені відстанями в декілька кілометрів; вони будуються з використанням високоякісних ліній зв'язки, які дозволяють, застосовуючи простіші методи передачі даних, чим в глобальних мережах, досягати високих швидкостей обміну даними до декількох Гігабітів в секунду. Послуги надаються

в режимі підключення і відрізняються різноманітністю.

Мережі MAN призначені для обслуговування території великого міста. При досить великих відстанях між вузлами (десятки кілометрів) вони мають якісні лінії зв'язку і підтримують високі швидкості обміну. Мережі MAN забезпечують економічне з'єднання локальних мереж між собою, а також доступ до глобальних мереж. В результаті тісної інтеграції LAN, WAN і MAN сталося взаємопроникнення відповідних технологій. Тенденція зближення різних типів мереж характерна не лише для локальних і глобальних комп'ютерних мереж, але і для телекомунікаційних мереж інших типів: телефонних мереж, радіомереж, телевізійних мереж.

Лінії зв'язку або лінії передачі даних - це проміжна апаратура і фізичне середовище, по якому передаються інформаційні сигнали (дані). Для побудови комп'ютерних мереж застосовуються лінії зв'язки, що використовують різне фізичне середовище: кабель "вита пара", коаксіальний кабель, оптичний кабель і навколишній простір.

Залежно від фізичного середовища передачі даних лінії зв'язку можна розділити на:

- дротяні або кабельні, де для передачі сигналів використовуються такі лінії зв'язку як кабелі "вита пара", коаксіальні кабелі або оптоволоконні кабелі;
- безпроводні (радіоканали наземного і супутникового зв'язку), що використовують для передачі сигналів електромагнітні хвилі, які поширюються по ефіру.

Найпоширеніша сучасна бездротова Wi - Fi - технологія дозволяє клієнтам у режимі Ad HOC встановлювати зв'язок безпосередньо один з одним. Встановлюється однорангова взаємодія за типом "крапка-крапка", і комп'ютери взаємодіють безпосередньо без застосування точок доступу. При цьому створюється тільки одна зона обслуговування, що не має інтерфейсу для підключення до дротової локальної мережі. Основна перевага даного режиму -

простота організації : він не вимагає додаткового обладнання (точки доступу). Режим може застосовуватися для створення тимчасових мереж для передачі даних.



Рисунок 2.2 - Інфраструктурний режим Wi-Fi – технології [1]

У інфраструктурному режимі Wi - Fi - технології точки доступу забезпечують зв'язок клієнтських комп'ютерів. Точку доступу можна розглядати як бездротовий комутатор. Точка доступу має порт Ethernet, через який базова зона обслуговування підключається до дротової або змішаної мережі - до мережевої інфраструктури (рис.2.2).

У режимі WDS Wi-Fi - технології точки доступу з'єднуються тільки між собою, утворюючи мостове з'єднання. При цьому кожна точка може з'єднуватися з декількома іншими точками. Всі точки в цьому режимі повинні використовувати однаковий канал, тому кількість точок, що беруть участь в утворенні моста, не повинно бути надмірно великим. Підключення клієнтів

здійснюється тільки по провідній мережі через uplink- порти точок (рис.2.3).

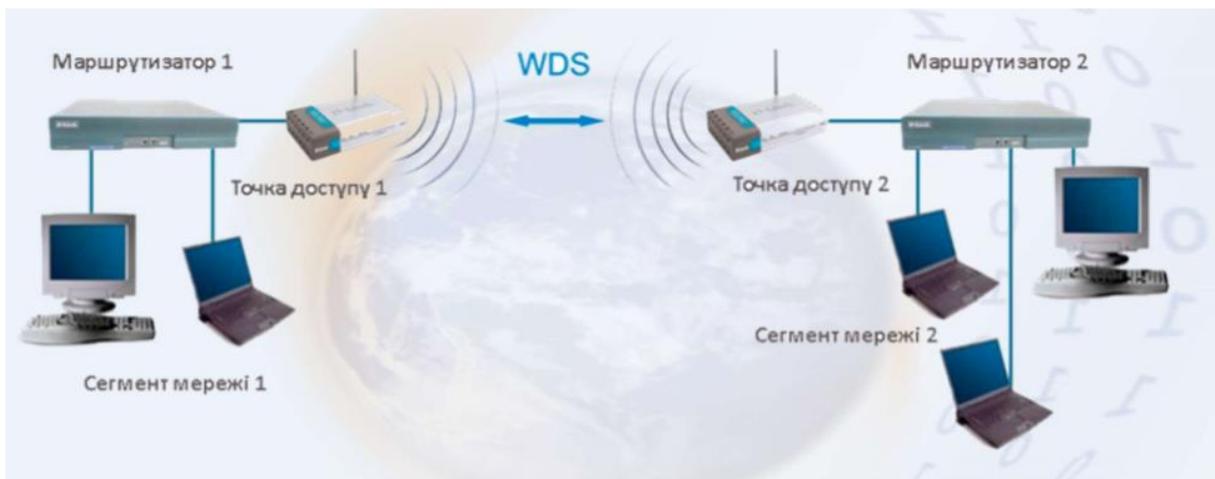


Рисунок 2.3 - Режим WDS Wi-Fi – технології [1]

Проста бездротова мережа для невеликого офісу або домашнього використання (Small Office / Home Office - SOHO) може бути побудована на основі однієї точки доступу (рис.2.4). Для організації мережі адаптери переводяться в режим інфраструктури, а точка доступу - в режим точки доступу. При цьому створюється одна зона обслуговування, в якій знаходяться всі користувачі мережі.

Прикладом бездротової мережі є AirPlay - бездротова потокова передача вмісту з мобільних пристроїв на TV. Службу AirPlay можна використовувати для потокової передачі музики, фотографій і відео на Apple TV або потокової передачі музики на динаміки з підтримкою AirPort Express або AirPlay. Функція відеоповтору AirPlay дозволяє виводити екран iOS на Apple TV.

Пристрої, що використовують технологію Wi-Fi :

- жорсткі диски с Wi-Fi доступом. Жорсткий диск Fuel є зовнішнім накопичувачем сумісним з iPad, iPhone і Mac, на якому зручно зберігати медіабібліотеку, і в будь-який час переглядати вміст з будь-якого пристрою від

Apple за допомогою Wi-Fi. Пристрій самостійно створює точку доступу Wi-Fi з можливістю підключення до п'яти різних пристроїв або трьох, за умови одночасного перегляду фільмів HD-якості. Вбудована батарея здатна до 10 годин підтримувати працездатність Fuel.



Рисунок 2.4 – Бездротова мережа для домашнього використання [1]

- принтери з технологією AirPrint – сучасна технологія і спосіб бездротового друку, здійснюваний за допомогою Wi-Fi і створений компанією Apple. Ця функція дозволяє просто і швидко роздрукувати необхідний файл або документ використовуючи при цьому пристрій на базі iOS. Для друку будь-якого документа з вашого iPhone або iPad вам необхідно лише перебувати в одній Wi-Fi мережі з принтером, що підтримує дану технологію. Не потрібно встановлювати ніяких драйверів або додатків. Просто вибираєте «Друк документа» і відправляєте його на друк.

• web камери Wi-Fi. Logitech Broadcaster Wi-Fi Webcam підтримує запис відео 720p, 3-кратний цифровий зум, панорамування і нахил. Вона оснащена вбудованим мікрофоном і гніздом для зовнішнього мікрофона. Передбачена лампочка підсвічування і гніздо для установки на штативі. Вбудована батарея забезпечує до 2 годин автономної роботи. Згідно із заявою Logitech, за допомогою цієї камери можна створювати «професійно виглядає відео високої чіткості на комп'ютері Apple». Передбачена потокова передача безпосередньо на сервіс Ustream і використання iPad, iPhone або Mac для управління і перегляду. Новинка може працювати другою камерою, додаючи новий ракурс до відео, що знімається вбудованою камерою iMac, MacBook Air або MacBook Pro.

3. Класифікація локальних мереж

Локальна мережа (ЛМ) - це різновид обчислювальної мережі, в якій ПК і комунікаційне устаткування знаходиться на невеликій відстані один від одного. ЛМ зазвичай призначена для збору, зберігання, передачі, обробки і надання користувачам розподіленої інформації в межах підрозділу або фірми. Крім того, ЛМ, як правило, має вихід в Інтернет. По адміністративних стосунках між комп'ютерами можна виділити:

- ЛМ з централізованим управлінням (з виділеними серверами);
- ЛМ без централізованого управління (децентралізовані, рівноправні) або однорангові (однорівневі) мережі.

У локальних мережах з централізованим управлінням один з комп'ютерів є сервером, а інші ПК - робочими станціями. *Сервери* - це високопродуктивні комп'ютери з вінчестерами великої місткості і з високошвидкісною мережевою картою, які відповідають за зберігання даних, організацію доступу до цих даних і передачу даних робочим станціям або клієнтам.

Робочі станції (клієнти) - комп'ютери, з яких здійснюється доступ до

інформації на сервері. У локальних мережах з централізованим управлінням сервер забезпечує взаємодію між робочими станціями, виконує функції зберігання даних загального користування, організовує доступ до цих даних і передає дані клієнтові. Клієнт обробляє отримані дані і надає результати обробки користувачеві. Важливо, що обробка даних може здійснюватися і на сервері. Локальні мережі з централізованим управлінням, в яких сервер призначений тільки зберігання і видачі клієнтам інформації по запитам, називаються мережами з *виділеним файл-сервером*. Системи, в яких на сервері разом зі зберіганням здійснюється і обробка інформації, називаються системами "*клієнт-сервер*". У серверних локальних мережах клієнтові безпосередньо доступні тільки ресурси серверів. Але робочі станції, що входять в ЛМ з централізованим управлінням, можуть одночасно організувати між собою однорангову локальну мережу з усіма її можливостями. Переваги ієрархічної ЛМ:

- вище швидкість обробки даних;
- має надійну систему захисту інформації і забезпечення секретності;
- простіше в управлінні в порівнянні з одноранговими мережами.

Її недоліки: мережа дорожче із-за виділеного сервера; менш гнучка в порівнянні з рівноправною мережею.

У мережах з децентралізованим управлінням немає єдиного центру управління взаємодією робочих станцій і єдиного комп'ютера для зберігання даних. Однорангова локальна мережа - це ЛМ рівноправних комп'ютерів, кожен з яких має унікальне ім'я і, як правило, пароль для входу в нього у момент завантаження операційної системи.

Рівноправність ПК означає, що адміністратор кожного комп'ютера в локальній мережі може перетворити свій локальний ресурс в такий, що розділяється, і встановлювати права доступу до нього і паролі. Він же відповідає за збереження або працездатність цього ресурсу. Локальний ресурс - ресурс, доступний тільки з ПК, на якому він знаходиться. Ресурс ПК, доступний для

інших комп'ютерів, називається таким, що розділяється або спільно використовуваним. Таким чином, однорангова локальна мережа - це ЛМ, в якій кожна робоча станція може розділити усі або деякі з її ресурсів з іншими робочими станціями мережі. Але відсутність виділеного сервера не дозволяє адміністраторові централізовано управляти усіма ресурсами однорангової локальної мережі. Кожна робоча станція може виконувати функції, як клієнта, так і сервера, тобто надавати ресурси іншим робочим станціям і використати ресурси інших робочих станцій. Однорангові ЛМ можуть бути організовані на базі усіх сучасних 32-розрядних операційних систем. Переваги однорангової ЛМ

- це низька вартість і висока надійність, головні її недоліки:
- кількість одночасно працюючих станцій не більше 10;
- слабкий захист інформації;
- складність оновлення і зміни програмного забезпечення робочих станцій.

4. Мережеві топології

Усі комп'ютери в ЛМ сполучені лініями зв'язку. Геометрична прихильність ліній зв'язку відносно вузлів мережі і фізичне підключення вузлів до мережі називається *фізичною топологією*. Нині в локальних мережах використовуються наступні фізичні топології:

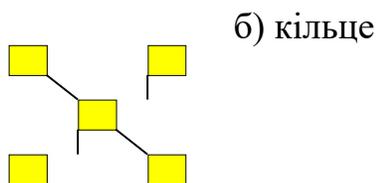
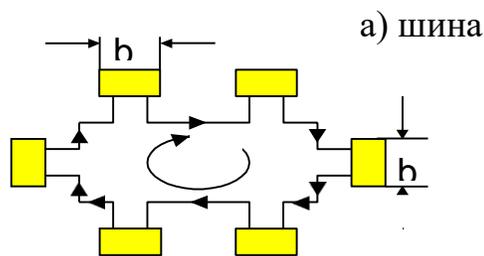
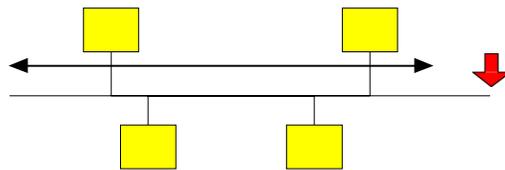
- фізична "шина" (bus);
- фізична "зірка" (star);
- фізичне "кілеце" (ring);

Мережі з шинною топологією (рис.2.5, а) використовують лінійний моноканал (коаксіальний кабель) передачі даних. Дані від передавального вузла мережі передаються по шині в обидві боки, інформація поступає на усі вузли, але приймається тільки тим вузлом, якому вона призначається. У топології логічна шина середовище передачі даних використовуються спільно і одночасно усіма

ПК мережі, а сигнали від ПК поширюються одночасно в усі напрями по середовищу передачі.

Переваги мереж шинної топології :

- відмова одного з вузлів не впливає на роботу мережі в цілому;
- мережу легко налаштувати і конфігурувати;
- мережа стійка до несправностей окремих вузлів.



в) зірка

Рисунок 2.5 – Структури побудови мережі : (а) шина, (б) кільце, (в) зірка.

Недоліки мереж шинної топології :

- розрив кабелю може вплинути на роботу усієї мережі;

- обмежена довжина кабелю і кількість робочих станцій;
- важко визначити дефекти з'єднань.

В мережі, побудованій по топології типу "зірка" (рис.2.5, в), кожна робоча станція з'єднується кабелем (витою парою) з концентратором або хабом (hub). Концентратор забезпечує паралельне з'єднання ПК і, таким чином, усі комп'ютери, підключені до мережі, можуть спілкуватися один з одним. Дані від передавальної станції мережі передаються через хаб по усіх лініях зв'язку усім ПК. Інформація поступає на усі робочі станції, але приймається тільки тими станціями, яким вона призначається. Переваги мереж топології зірка:

- легко підключити новий ПК;
- є можливість централізованого управління;
- мережа стійка до несправностей окремих ПК і до розривів з'єднання окремих ПК.

Недоліки мереж топології зірка – це відмова хаба, яка впливає на роботу усієї мережі, та великі витрати кабелю;

У мережі з топологією "кільце" (рис.2.5, б) усі вузли сполучені каналами зв'язку в нерозривне кільце (необов'язково коло), по якому передаються дані. Вихід одного ПК з'єднується з входом іншого ПК. Почавши рух з однієї точки, дані потрапляють на його початок. Дані в "кільці" завжди рухаються в одному і тому ж напрямі. У кільці, на відміну від інших топологій (зірка, шина), не використовується конкурентний метод посилки даних, комп'ютер в мережі отримує дані від того, що стоїть попереднім в списку адресатів і перенаправляє їх далі, якщо вони адресовані не йому. Пакет передається до тих пір, поки не дістанеться до одержувача. До основного недоліку мереж топології "кільце" відноситься те, що ушкодження лінії зв'язку в одному місці або відмова ПК приводить до непрацездатності усієї мережі. Як правило, в чистому вигляді топологія "кільце" не застосовується із-за своєї ненадійності, тому на практиці застосовуються різні модифікації кільцевої топології.

5. Сучасні операційні системи комп'ютерних засобів

Операційна система - (ОС) - базовий комплекс комп'ютерних програм, що забезпечує управління апаратними засобами комп'ютера, роботу з файлами, введення і виведення даних, а також виконання прикладних програм і утиліт.

Файл - ім'я в файлової системі для доступу до ресурсів. Ресурсами можуть бути:

- область даних (не обов'язково на диску - це може бути і ОЗУ);
- пристрій;
- вхід або вихід іншого процесу;
- мережевий ресурс.

У більшості файлових систем ім'я файлу використовується для вказівки до якого саме файлу проводиться звернення. У різних файлових системах обмеження на ім'я файлу сильно розрізняються.

При включенні комп'ютера операційна система завантажується в пам'ять раніше інших програм і потім служить платформою і середовищем для їх роботи. Крім вищевказаних функцій ОС може здійснювати й інші, наприклад, надання користувальницького інтерфейсу, мережеве взаємодія і т. д. З 1990-х років найбільш поширеними операційними системами є ОС сімейства Microsoft Windows і системи класу UNIX (особливо Linux).

Основні функції ОС наступні:

- Завантаження програм в оперативну пам'ять і їх виконання;
- Стандартизований доступ до периферійних пристроїв (пристрої введення-виведення);
- Управління оперативною пам'яттю (розподіл між процесами, віртуальна пам'ять);
- Управління енергонезалежною пам'яттю (жорсткий диск, компакт-диск і т.д.), як правило, за допомогою файлової системи;

- Користувальницький інтерфейс; Додаткові функції (розвинені сучасні ОС):
- Паралельне або псевдопаралельне виконання завдань (багатозадачність);
- Взаємодія між процесами;
- міжмашинні взаємодії (комп'ютерна мережа);
- Захист самої системи, а також призначених для користувача даних і програм від зловмисних дій користувачів або додатків;
- Розмежування прав доступу і розрахований на багато користувачів режим роботи (аутентифікація, авторизація).

Система UNIX придбала популярність у зв'язку з її успішним використанням на міні-ЕОМ. Цей успіх став поштовхом до того, щоб створити подібну систему і для персональних комп'ютерів. Як правило, різні версії ОС, що відносяться до цієї родини, мають свої назви, але в основних рисах повторюють особливості UNIX. UNIX - операційна система, яка дозволяє здійснити виконання робіт у багатокористувацькому і багатозадачному режимі. Спочатку вона призначалася для великих ЕОМ, щоб замінити MULTICS. UNIX є дуже потужним засобом у руках програміста, але вимагає дуже великого обсягу ОЗП і простору диска. Незважаючи на спроби стандартизувати цю операційну систему, існує велика кількість різних її версій, головним чином тому, що вона була поширена у вигляді програми на мові Сі, яку користувачі стали модифікувати для своїх власних потреб.

Головною відмінною рисою цієї системи є її модульність і великий набір системних програм, які дозволяють створити сприятливу обстановку для користувачів-програмістів. UNIX має "оболонку", з якою користувач безпосередньо взаємодіє, і "ядро", яке, власне, і керує діями комп'ютера. Від UNIX багато інших операційних систем перейняли такі функції, як перепризначення, канал і фільтр, а проте UNIX має безсумнівно перевага в тому, що вона з самого початку розроблялася як багатокористувальницька і

багатозадачна операційна система.

Linux - загальна назва UNIX – подібних ОС на основі однойменного ядра. Це один із найвидатніших прикладів розробки вільного (free) та відкритого (з відкритим кодом, open source) програмного забезпечення (software). На відміну від власницьких операційних систем (Microsoft Windows та MacOS X), їхні вихідні коди доступні всім для використання, зміни та поширення абсолютно вільно (в тому числі безкоштовно).

Linux, спершу розроблений для використання окремими ентузіастами на своїх персональних комп'ютерах, пізніше, завдяки підтримці таких компаній, як IBM, Sun Microsystems, HP, Novell та інших, набув неабиякої популярності як серверна ОС. Linux портовано на велику кількість апаратних платформ. Тепер ця ОС досить успішно використовується як на мейнфреймах та суперкомп'ютерах, так і вбудована в багато інших пристроїв (смартфони, планшетні ПК, маршрутизатори комп'ютерних мереж (роутери), пристрої автоматики, системи керування телевізорами та ігровими консолями тощо). Значна кількість спеціалізованих дистрибутивів Linux, які розробляють та підтримують різні спільноти, надає широкі можливості вибору програмного забезпечення.

Сучасні мобільні телефони стають все більш «розумними», недарма ж їх називають смартфонами (в перекладі з англійської smart phone - розумний телефон). Для нормального функціонування таким телефонам потрібна повноцінна операційна система. Розглянемо більш детально існуючі мобільні операційні системи. Один з «піонерів» серед мобільних операційних систем - це Symbian OS. Розробкою її займається консорціум Symbain, який в 1998 році заснували компанію Nokia, Ericsson, Motorola і Psion, пізніше до нього приєдналися інші виробники мобільних телефонів. Різні версії цієї ОС були адаптовані під певні моделі телефонів. Вона була популярна завдяки своїй легкості і стабільності, низьким вимогам до начинки» телефону, а також частому виходу нових версій і швидкому виправленню нестабільностей. Ну, і звичайно,

важливу роль відіграло те, що платформа Symbian активно підтримувалася одним з провідних виробників телефонів, Nokia, а також довгий час практично не мала конкурентів. Але коли почали активно розвиватися інші мобільні операційні системи (зокрема, Windows Mobile, Google Android, Apple iOS), Symbian OS здала свої позиції, хоча до цих пір залишається досить популярною.

Система Google Android буквально «дихає в спину» Symbian OS. Незважаючи на свою молодість, вона дуже популярна. Google Android заснована на ядрі Linux, вона використовується не тільки в мобільних телефонах, а й комунікаторах, нетбуках, планшетних комп'ютерах та інших пристроях. Ця система приваблює своєю гнучкістю: завдяки відкритому вихідному коду кожен розробник може змінювати її «під себе». Для користувачів ця відкритість виражається у великій кількості різноманітних додатків. Вона багатозадачна, відрізняється високою швидкістю і зручною інтеграцією з сервісами Google. Але відкритість - також причина і деяких недоліків цієї ОС. Наприклад, вона часто вимагає доопрацювань.

Деякі виробники телефонів вважають за краще використовувати власні мобільні операційні системи. Скажімо, iPhone від Apple працюють під управлінням операційної системи Apple iOS. Пристрої від компанії Research In Motion Limited (RIM) - смартфони BlackBerry - оснащені однойменною операційною системою. А серія тачфонів (сенсорних телефонів) Wave від Samsung працює на платформі Bada (їх так і називають - «бадафони»). Перевага цих систем в тому, що вони «заточені» під апарати конкретного виробника з урахуванням всіх їх особливостей. Зазвичай такі системи надійні, регулярно оновлюються і можуть похвалитися хорошою службою підтримки. Але є у них і недоліки, скажімо, користувачі BlackBerry OS скаржаться на не надто зручний браузер, а на iOS не можна встановити неофіційні програми. Але якщо ви вирішили неодмінно купити собі iPhone, Blackberry або Samsung Wave, з вадами доведеться миритися.

Шосте місце за популярністю після Symbian OS, Google Android, Apple iOS, BlackBerry OS і Bada займає операційна система Windows Mobile, заснована на платформі Windows CE. Вона зручна тим, що її інтерфейс схожий з інтерфейсом звичної нам операційної системи Windows для персональних комп'ютерів. Вона багатозадачна, може похвалитися зручною синхронізацією. Але при цьому ОС Windows Mobile пред'являє досить високі вимоги до «начинки» смартфона, вразлива для вірусів і не завжди працює стабільно.

Порядок виконання роботи

1. Створити папку на робочому столі. Для створення нової папки зробити клацання правою кнопкою миші на порожньому місці робочого столу і в меню, що випадає, вибрати «Папка» (рис 2.6).
2. Перейменувати папку (своє ім'я) шляхом натискання клавіші F2.
3. Відкрити цю папку (подвійне клацання на ній) і створити текстовий документ з ім'ям «Конфігурація комп'ютера» як показано на рис.2.7.
4. Відкрити щойно створений текстовий документ і записати в нього конфігурацію Вашого комп'ютера. Для цього відкрити меню «Пуск» і обрати «Налаштування», відкрити вікно, що наведено на рис.2.8.
5. Обрати значок «Система» і пункт «Про програму», як показано на рис. 2.9.
6. Нижче, справа обрати пункт «Диспетчер пристроїв». При цьому відкриється вікно (див. рис.2.10).

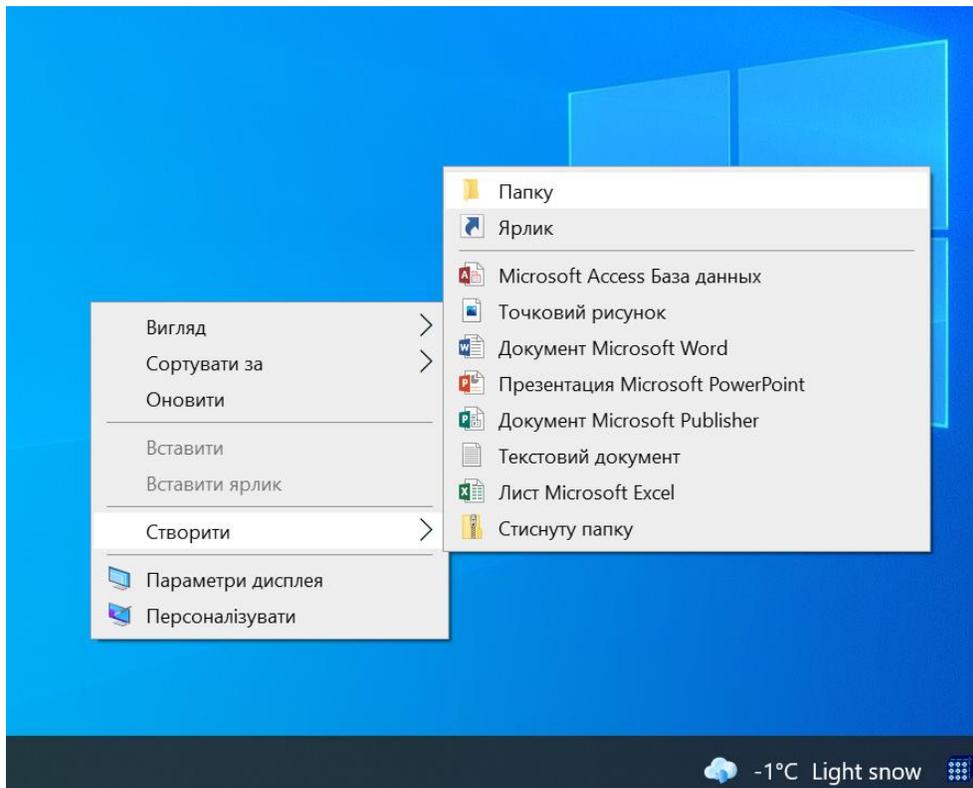


Рисунок 2.6 - Створення нової папки на робочому столі.

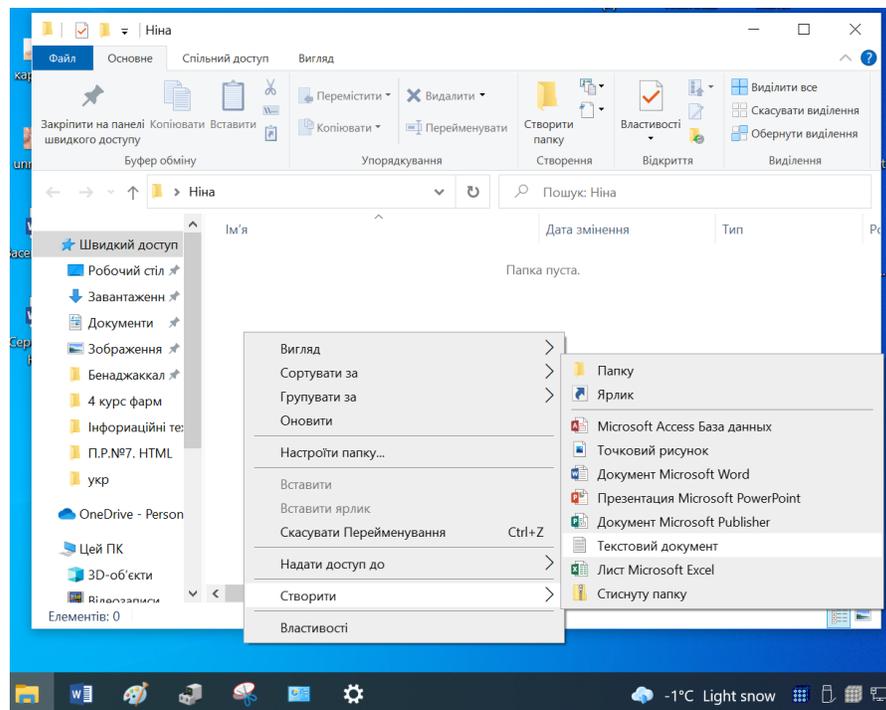


Рисунок 2.7 - Створення нового текстового документа.

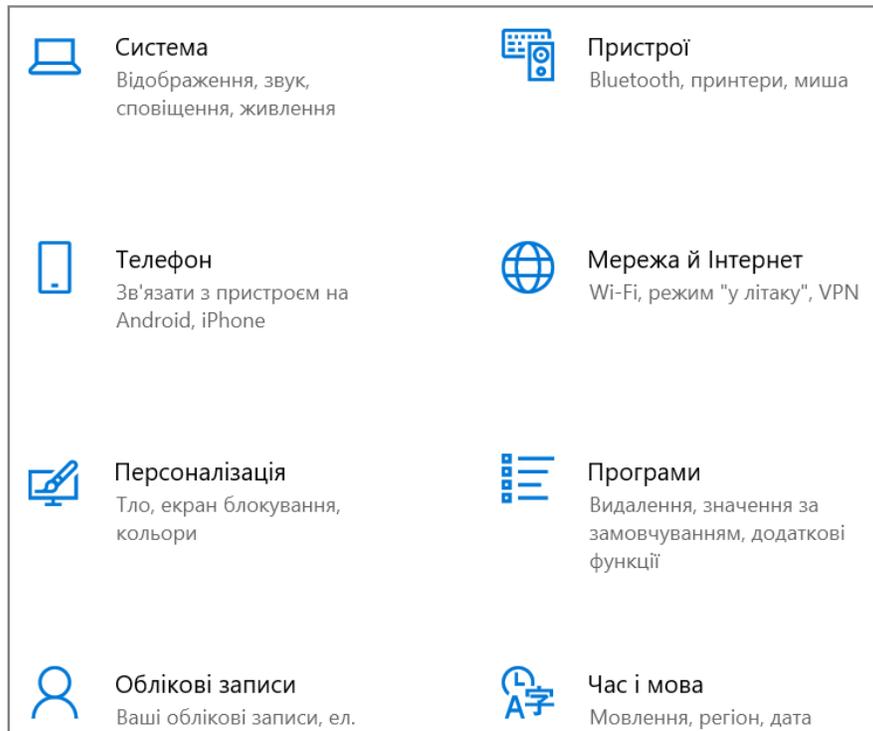


Рисунок 2.8 - Панель управління комп'ютера.

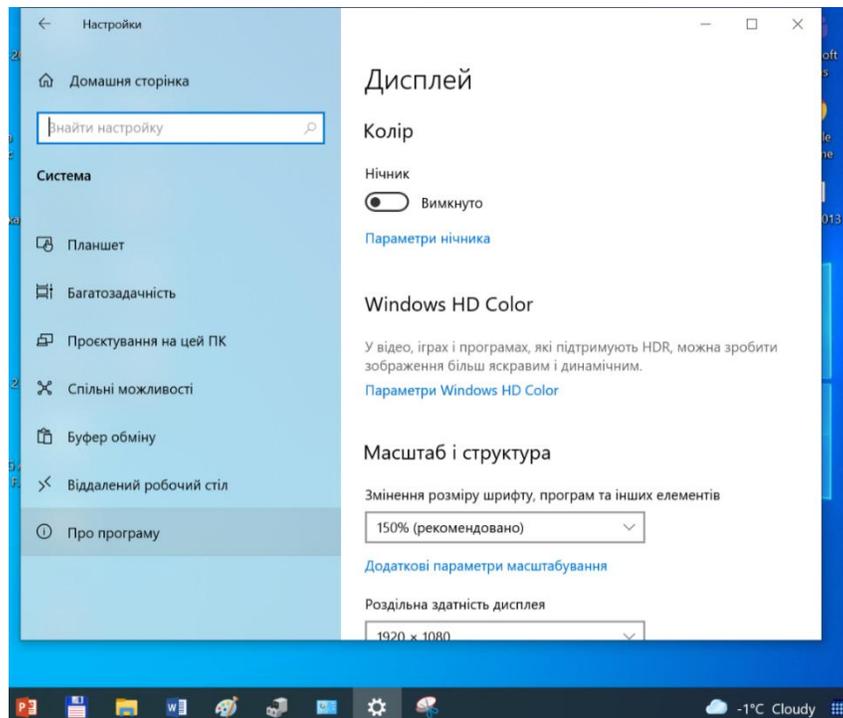


Рисунок 2.9 – Властивості системи

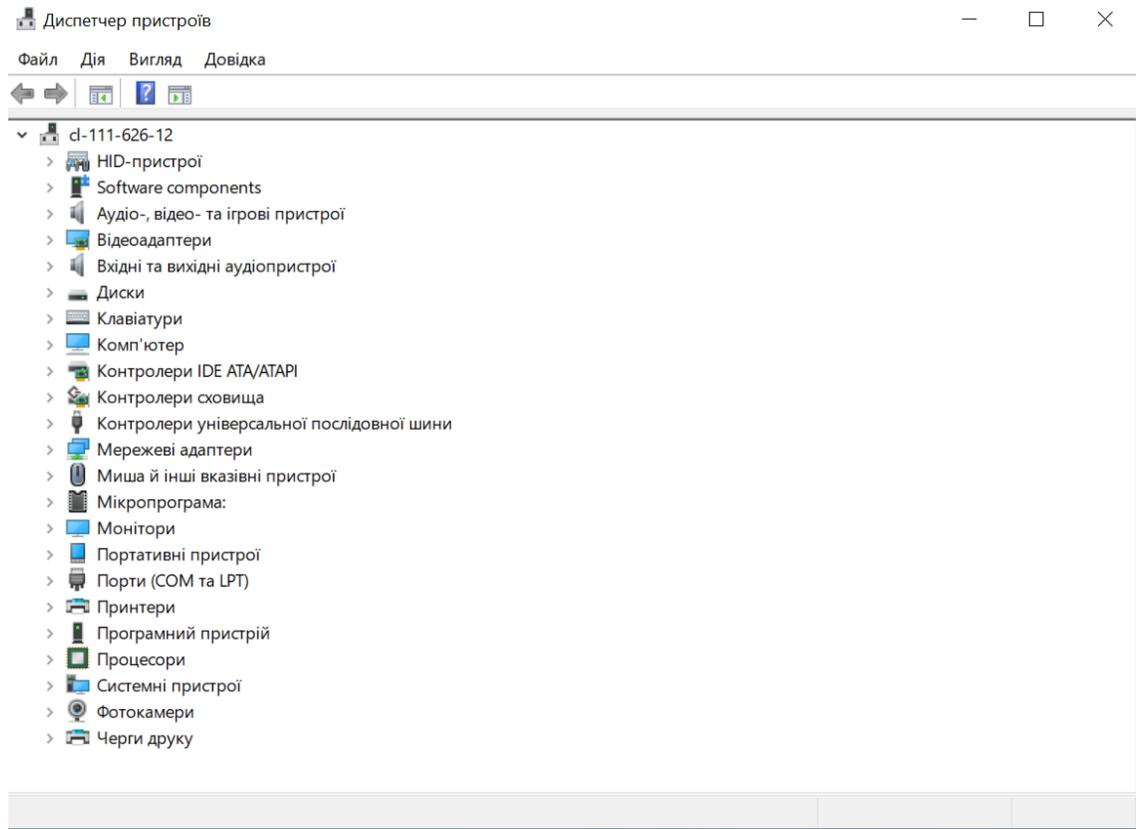


Рисунок 2.10 – Диспетчер пристроїв

7. У створений Вами файл «Конфігурація комп'ютера» записуєте (рисунок 2.10):
- a) Тип процесора;
 - b) Інформацію про жорсткий диск;
 - c) Інформацію про DVD / CD - дисководи;
 - d) Тип монітора;
 - e) Інформацію про мережеву плату;
 - f) Інформацію про відеоадаптер;
 - g) Інформацію про оперативну пам'ять.
8. Збережіть ці дані, натиснувши поєднання клавіш Ctrl + S. Закрийте текстовий документ.

9. Створіть файл з ім'ям «Конфігурація телефону» та занесіть в нього аналогічні параметри свого мобільного телефону.
10. Додайте обидва документи в архів з ім'ям «Конфігурація». Для цього виділіть мишею документи і виберіть «Додати до архіву Конфігурація. 7-Zip» (див. рис.2.11). Відкрийте щойно створений архів та розпакуйте його.
11. Перетворіть архів в такий, що розпаковується самостійно. Для цього натисніть на піктограму SFX в верхньому правому куті програми.
12. Зробіть висновки, порівнюючи конфігурації двох комп'ютерних засобів.

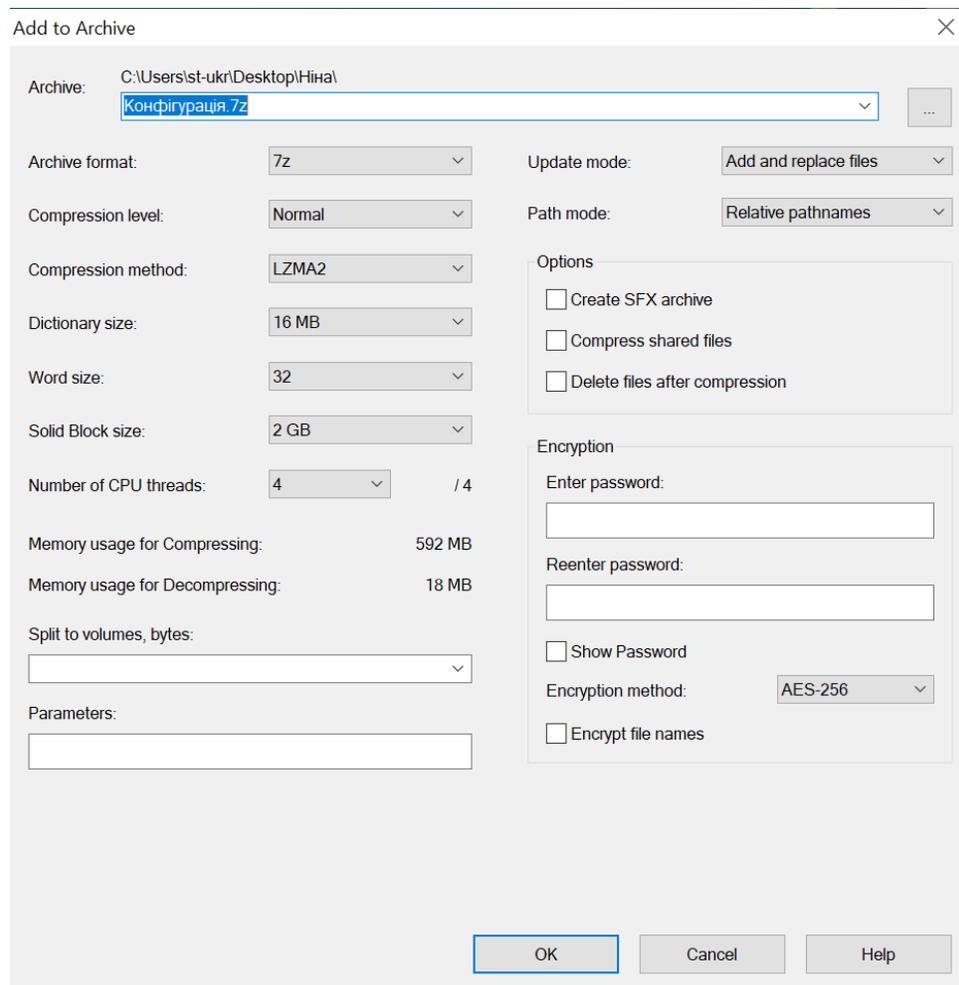


Рисунок 2.12 - Вікно архіватора WinRAR.

Питання для самоконтролю

1. Назвіть основні компоненти апаратної частини персонального комп'ютера
2. Дайте характеристику видам комп'ютерних мереж
3. Назвіть переваги та недоліки кожної топології комп'ютерної мережі
4. Дайте визначення операційній системі, назвіть сучасні операційні системи
5. Визначте ієрархію в організації пам'яті в комп'ютері

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ №3

Тема: Основи проектування навчальних елементів для онлайн курсів edX. Обробка медичних зображень у програмі ACD ChemSketch. 3D – моделювання хімічних сполук

Мета роботи: Опанувати навичками проектування навчальних елементів для використання їх при розробці онлайн курсів edX.

Теоретичні відомості

Електронне навчання (англ. E-learning, скорочення від англ. Electronic Learning) — система навчання за допомогою інформаційних, електронних технологій. За визначенням фахівців ЮНЕСКО: «e-Learning — навчання за допомогою інтернету і мультимедіа».

1. Що це таке - онлайн навчання?

Процес навчання є взаємодією педагога, студента і засобів навчання. Можливості сучасних комп'ютерних засобів і інформаційних технологій дозволяють покласти на засоби навчання частину функцій викладача і частину функцій навчаного, прийнятого в класичній формі навчання.

Інформаційні технології в онлайн навчанні грають роль інструментів, які:

- забезпечують учням віддалений доступ не тільки до навчального матеріалу, а й до великої кількості довідкової інформації, в додатковій, що супроводжує формі;
- надають учням засоби спілкування з онлайн репетитором, а також між собою; сьогодні є технічні можливості для того, щоб учень, знаходиться на великій відстані від учбового центру (школи, внз, коледжу, ліцею), прослухав і проглянув лекцію, взяв участь у відеоконференції або отримав консультацію,

виконав комп'ютерний лабораторний експеримент і т. ч. здійснюють управління і контроль за процесом навчання;

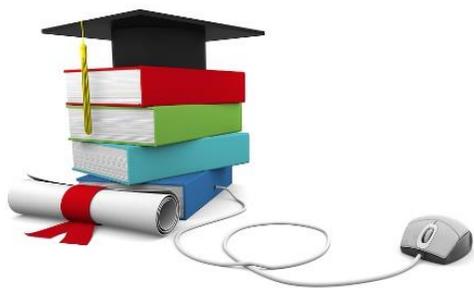
- навчаний повинен переконатися, перед усім сам в тому, що розібрався в учбовому матеріалі, який вивчався, зрозумів його, запам'ятав основні положення, навчився застосовувати їх на практиці для вирішення практичних завдань. З іншого боку, активна роль онлайн викладача не менш істотна, оскільки його завдання не лише переконатися в знаннях підопічного, але і - як при очному навчанні - прийняти рішення по коригуванню програми навчання з тим, щоб добитися найкращого засвоєння пройденого матеріалу
- надають можливість створення ефективних тренажерів, засобів візуалізації, максимальне використання різних способів представлення інформації : тексту, графіки, відео, звукового супроводу, анімації, т. е. те, що дістало назву " мультимедіа";

Ключовим елементом побудови навчання з використанням інтернет-технологій є забезпечення віддаленого доступу до навчального контенту. Широке поширення отримують такі різновиди онлайн навчання як віртуальна школа, дистанційні курси і т.п.



- **Віртуальний навчальний заклад** - освітній заклад, в якому педагогічний процес і навчання здійснюються через Інтернет. Матеріали з навчальних курсів у віртуальному навчальному закладі представлені в електронному вигляді і викладаються на веб-сайті. Ці матеріали зазвичай включають в себе тексти лекцій з предмету, інтерактивні тести і тренажери, словники і т. д. Ознайомившись з матеріалами віртуального уроку, студент виконує ряд завдань, які автоматично перевіряються системою, з виставленням оцінки. Студент може вступати у взаємодію з мережевими викладачами, консультуючись з предметів. Мережеві викладачі

можуть також здійснювати контроль і оцінку знань, спілкуючись зі студентами по електронній пошті, по телефону, в форумі або за допомогою інших технічних засобів зв'язку. Атестація зазвичай здійснюється у формі іспиту (очного або заочного).



Дистанційний курс - особлива, заснована на використанні сучасних інформаційних технологій, форма подання змісту навчального курсу. Дистанційний курс є основним елементом побудови навчання з використанням технологій дистанційного навчання

Використання Internet дозволяє організовувати наступні форми занять:

Чат-заняття - навчальні заняття, здійснювані з використанням чат-технологій. Чат-заняття проводяться синхронно, тобто всі учасники мають одночасний доступ до чату.

Веб-заняття - дистанційні уроки, конференції, семінари, ділові ігри, лабораторні роботи, практикуми та інші форми навчальних занять, що проводяться за допомогою засобів телекомунікацій та інших можливостей Internet. Для веб-занять використовуються спеціалізовані освітні веб-форуми - форма роботи користувачів з певної теми або проблеми за допомогою записів, що залишаються на одному з сайтів з встановленою на ньому



відповідною програмою.

Від чат-занять веб-форуми відрізняються можливістю більш тривалої (багатоденної) роботи і асинхронним характером взаємодії учнів і педагогів. Чат заняття проходять у синхронному режимі.

Телеконференції- проводяться, як правило, на основі списків розсилки з використанням електронної пошти. Для навчальних телеконференцій характерно досягнення освітніх завдань. Також існують форми дистанційного навчання, при якому навчальні матеріали висилаються поштою в регіони.

Онлайн-семінар-різновид веб-конференції, проведення онлайн-зустрічей або презентацій через Інтернет в режимі реального часу. Під час веб-конференції кожен із учасників знаходиться у свого комп'ютера, а зв'язок між ними підтримується через Інтернет за допомогою завантаженого додатки, встановленого на комп'ютері кожного учасника, або через веб-додаток.

Звичайно, система онлайн навчання також не ідеальна. По-перше, виникають складнощі у визначенні особистості учня. Тобто, поки неможливо точно перевірити, чи здає іспит студент самостійно або хтось за нього. По-друге, буває, що якості Інтернету недостатньо, щоб налагодити безперебійний зв'язок між студентами та викладачами. По-третє, курси дистанційного навчання сприяють тому, що безпосередній контакт між студентами і викладачем втрачається.

Онлайн навчання прекрасно підходить для тих, хто живе у віддалених районах, а також для тих, хто в силу певних причин не може відвідувати очну форму навчання. Крім того, безперечною перевагою дистанційних курсів навчання через Інтернет є те, що студент може сам обирати, в який час доби йому зручніше займатися, а також визначити для себе індивідуальну тривалість занять.

Дистанційне навчання не має жорсткого розкладу занять, а всі нюанси завжди можуть зважитися найбільш швидким чином за допомогою електронної

пошти, скайпу або ICQ. Крім того, з'являється можливість поговорити з викладачем on-line і поставити всі питання з того чи іншого предмету.

2. Що таке відкриті онлайн-курси?

Масові відкриті онлайн-курси, (МООС) (з англ.англ. Massive open online course — масивні, масові, широкодоступні, публічні, відкриті дистанційні онлайн курси) — це інтернет-курс з великомасштабною інтерактивною участю та відкритим доступом через інтернет. На додаток до традиційних матеріалів навчального курсу, такі як відео, читання, і домашніх завдань, МООС надає можливість використання інтерактивного форуму користувачів, які допомагають створити спільноту студентів, викладачів та асистентів (TAS).

Відеозаписи лекцій різних навчальних закладів почали з'являтися у мережі Інтернет ще наприкінці 1990-х років, однак лише масові відкриті онлайн-курси надали змогу інтерактивного спілкування студентів та викладачів, а також прийому іспитів в режимі онлайн. Це одна із найновіших форм дистанційного навчання, яка активно розвивається у світовій освіті. Подібні сайти розраховані на студентів різних попередніх рівнів підготовки — як новачків, так і досвідчених фахівців.

Найбільш поширені сервіси для навчання онлайн

Coursera

Пропонує своїм слухачам сотні безкоштовних онлайн-курсів з різних дисциплін, у разі успішного закінчення яких користувач отримує сертифікат про проходження курсу. Coursera співпрацює з університетами з різних країн світу для викладання курсів цих навчальних закладів онлайн. Курси є безкоштовними, але для тих, хто хоче отримати сертифікат із персональною верифікацією, передбачені платні сертифікати для окремих курсів. Протягом навчання студент повинен переглядати відеолекції, які надсилаються йому щотижня, читати

рекомендовані статті та виконувати домашні завдання. Деякі курси мають українські субтитри.

Khan Academy

Освітня організація, яка була створена ще у 2006 році. Мета організації – надання якісної освіти кожному і всюди. На сайті є кілька тисяч безкоштовних мікролекцій з математики, фізики, хімії, історії, фінансів, економіки, біології, мистецтва, комп'ютерних наук та інші. Всі курси розділені на уроки, з можливістю перегляду незалежно від курсу, який ви вибрали. Проект підтримується за рахунок пожертвувань.

Udacity

Більшість курсів Udacity – технічної спрямованості. В кожному курсі є кілька уроків, які містять короткі відео. В кінці заняття є завдання, щоб перевірити наскільки ви засвоїли матеріал. В середньому курс триває до 2 місяців, а щотижня на нього треба витратити орієнтовно 6 годин на навчання.

edX

Спільний проект Масачусетського технологічного та Гарвардського університетів. Кількість представлених іменитих вузів тут просто вражаюча. Більше 200 університетів зі всього світу виявили бажання співпрацювати з проектом і надають свої матеріали. Курси безкоштовні, але якщо захочете сертифікат – доведеться заплатити.

EdX— безкоштовна інтернет платформа масових відкритих інтерактивних курсів. EdX проводить онлайн-курси університетського рівня в широкому діапазоні дисциплін для слухачів зі всього світу на безоплатній основі, а також проводить дослідження в галузі навчання. Зараз є 53 школи, некомерційні організації, корпорації та міжнародні організації, які пропонують або планують

пропонувати курси на ресурсі EdX.

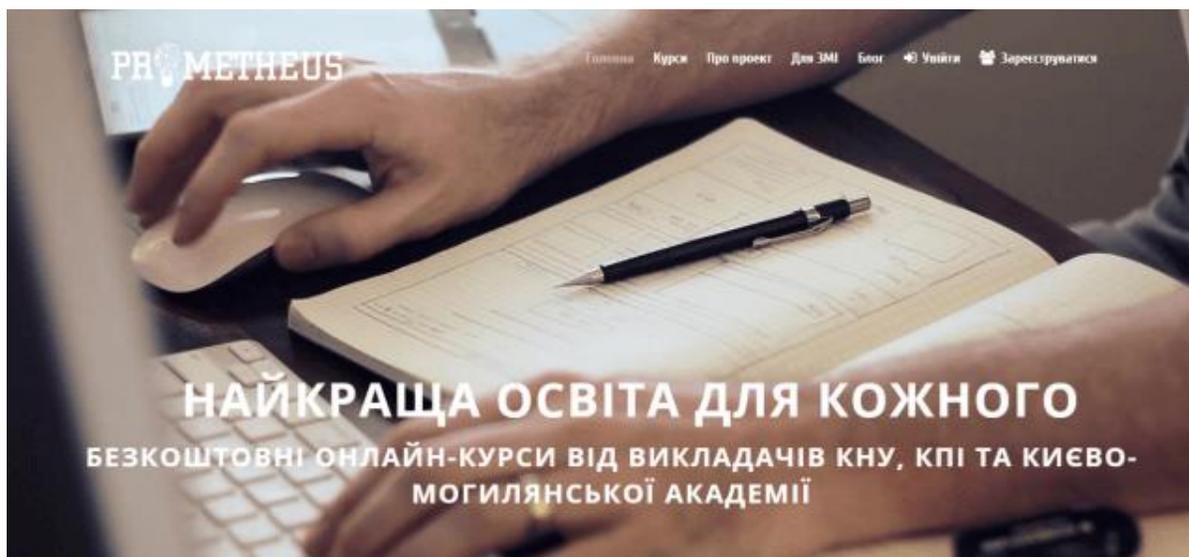
Udemy

Тут представлені найрізноманітніші курси, серед яких продуктивність, стиль життя чи, навіть, музика. Є як безкоштовні курси, так і платні. Навчальні матеріали представлені у вигляді відео, аудіо, презентаціями та текстом. Udemy також пропонує можливість організаціям створювати власні навчальні проекти для корпоративного навчання.

iTunes U

Ця платформа підійде для власників Apple-пристроїв. Список запропонованих дисциплін дуже великий, як і список навчальних закладів, що надають лекції. Часто в уроці є допоміжні файли, де пояснюються деякі аспекти лекцій, або ж наводяться приклади застосування пройденого уроку.

Prometheus



Громадський проект масових відкритих онлайн-курсів. Цей український сервіс стартував лише нещодавно, але вже набув великої популярності. Зараз

доступні чотири курси: Prometheus надає безкоштовну можливість університетам, провідним викладачам та компаніям-лідерам у своїй галузі публікувати та розповсюджувати курси на цій платформі.

Кожен курс складається із відеолекцій, інтерактивних завдань, а також форуму, на якому студенти мають змогу поставити питання викладачу та поспілкуватись один з одним. Успішне завершення курсу дасть змогу отримати електронний сертифікат, який підтверджуватиме здобуті знання.

3. Основні рекомендації, що до створення on line – курсів

. Використовуйте інтерактивні елементи в створенні курсу, а саме: відео та аудіо ролики, кейси, фонова музика і все, що допоможе зробити електронне навчання більш ефективним і результативним. Головною метою для розробника є - домогтися емоційної прихильності від користувача. Емоційні реакції допомагають учням краще засвоювати нову інформацію і навички. Відео-ролики або зображення, які можуть викликати емоційний відгук, залишаються в пам'яті надовго і змушують думати про себе навіть після закінчення навчання.

Тут, мабуть, самим потужним і доступним інструментом стане інтегрування в курс навчання прикладів з реального життя, які студент може застосувати в своєму окремому випадку. Це продемонструє учням з перших рук, як набуті знання можуть бути застосовані поза навчального середовища.

Дуже важливо в процесі навчання *організувати групову взаємодію*. Забезпечте учнів спілкуванням один з одним. Використовуйте групові дискусії на інтернет-форумах і всіляко заохочуйте їх, щоб вирішувати проблеми колективно в режимі онлайн.

Дуже потужним інструментом є *соціальні мережі*. Можна інтегрувати соціальні медіа в стратегії електронного навчання. Співпраця у групах дозволяє увімкнути людський фактор в e-learning курс, не дивлячись на деяку обмеженість

віртуального зв'язку. Живе спілкування надасть аудиторії можливість навчатись один у одного, і тим самим вивчити тему глибше, обмінюючись досвідом та своїми враженнями у спеціально створених групах.

Порядок виконання роботи

Необхідно розробити фрагмент on line – курсу, що присвячений біосинтезу білка. Цей процес складається з 4 етапів (див.табл.3.1).

Викладач розподіляє студентську групу на команди (по 2-3 людини). Кожна команда отримує своє завдання від викладача (номер етапу біосинтезу білка), яке виконує за допомогою програм ACD / ChemSketchi, PowerPoint. Сценарій завдання має бути реалізований в документі Microsoft Word.

Завдання, яке потрібно виконати:

1. Увійдіть до Office 365, використовуючи свій обліковий запис.
2. Відкрийте сервіс OneDrive.
3. Створіть папку з назвою «ПРАКТИЧНЕ ЗАНЯТТЯ№3».
4. Відкрийте програму Power Point.
5. Використовуючи Internet, підберіть матеріал у різних форматах (текст, малюнки, таблиці, схеми, відео) для представлення обраного навчального елемента з подальшим розміщенням цього матеріалу в онлайн курсі.
6. Використовуючи матеріал, який ви підібрали, створіть презентацію у PowerPoint, яка презентує **ваше уявлення** про сутність матеріалу щодо обраного навчального елемента та логіку його подачі.
7. Якщо ви, за певних обставин, не встигли зробити презентацію до кінця, ви маєте змогу допрацювати її вдома. Обов'язкова умова: ви повинні надіслати вашому викладачу на пошту посилання на вашу завершену презентацію, яка знаходиться у вашому віртуальному просторі (Ваш OneDrive). Це необхідно зробити до наступного заняття.

Таблиця 3.1 – Етапи біосинтезу білка

Етапи	Характеристика
I. Транскрипція	Передача інформації про структуру білка з молекули ДНК на ІРНК. Ділянка ДНК є матрицею для відповідної ІРНК. Синтезовані молекули ІРНК переходять із ядра в цитоплазму, а ДНК відновлює свою структуру
II. Активація амінокислот	Відбувається в цитоплазмі. Активовані молекули амінокислот з'єднуються з відповідними молекулами тРНК. У молекулі тРНК є дві ділянки: акцепторна, до якої прикріплюється відповідна амінокислота, та ділянка, що містить антикодон — триплет нуклеотидів, який комплементарний кодону ІРНК даної амінокислоти. Активовані амінокислоти, сполучені з тРНК, надходять до рибосом
III. Трансляція	Синтез поліпептидних ланцюгів. Молекула ІРНК рухається між двома субодиницями рибосом, до неї послідовно приєднуються молекули тРНК з амінокислотами. За принципом комплементарності кодони ІРНК вступають у зв'язок з антикодонами тРНК. Про завершення синтезу сигналізує термінуючий кодон ІРНК (УАА, УАГ, УГА)
IV. Утворення вторинної і третинної структур білкової молекули	Здійснюється в цитоплазмі шляхом скручування, згортання поліпептидного ланцюга. Потім до нього приєднуються органічні молекули — вуглеводи, жирні кислоти тощо. Цей процес відбувається в ЕПС та комплексі Гольдж

4. Робота у програмі ACD / ChemSketch

Для роботи у програмі ACD / ChemSketch треба виконати наступні дії:

1. Відкрити редактор ACD / ChemSketch і створити новий документ. Вікно редактора має вигляд, див. рис. 1. Ознайомитися з інтерфейсом програми.

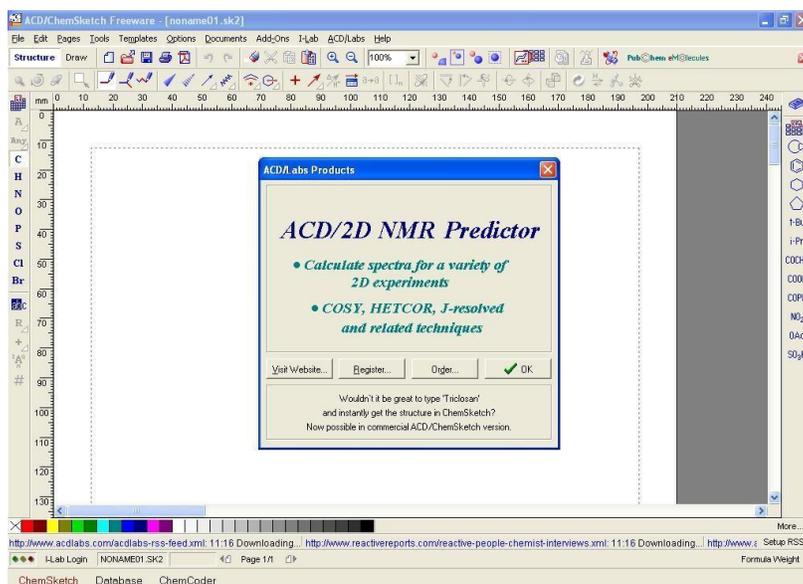


Рисунок 3.1. Вікно редактора ACD / ChemSketch

2. Дотримуючись інструкцій, опишимо нижче створити плоску модель хімічної формули:

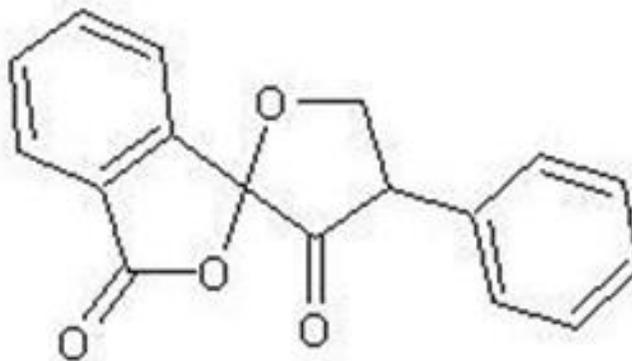


Рисунок 3.2. Структурна формула $C_{17}H_{10}O_4$

3. Відкрити таблицю радикалів, що знаходиться правому верхньому куті редактора:



Рисунок 3.3. Відкриття таблиці радикалів

4. З таблиці радикалів обрати «циклопентан», див. рис. 3.4 і помістити його на робочий лист, див. рис. 3.5.
5. Далі використовуючи цей же елемент, з'єднати його з вже встановленими, див. рис. 3.6.
6. Діючи аналогічним чином нанести на лист та об'єднати з наявними елементами «бензин».

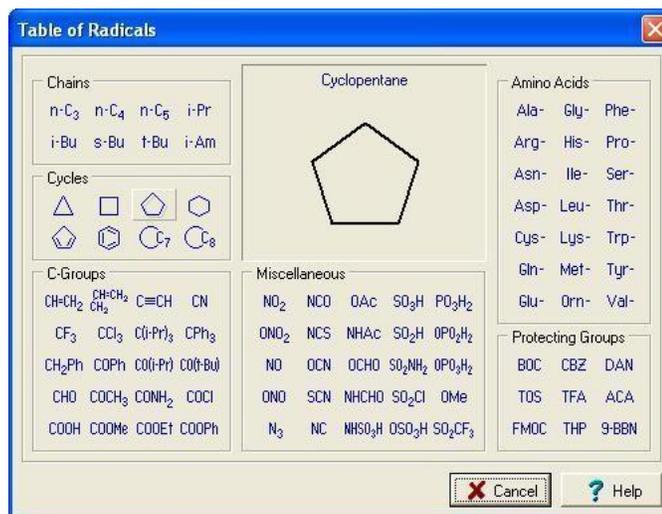


Рисунок 3.4. Таблица радикалів

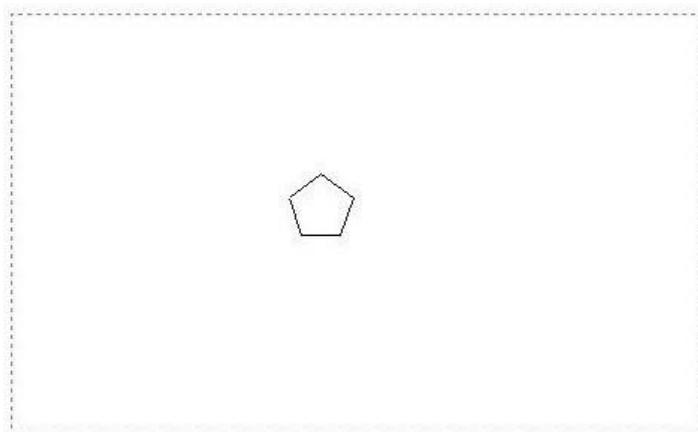


Рисунок 3.5. На формі розміщено циклопентан

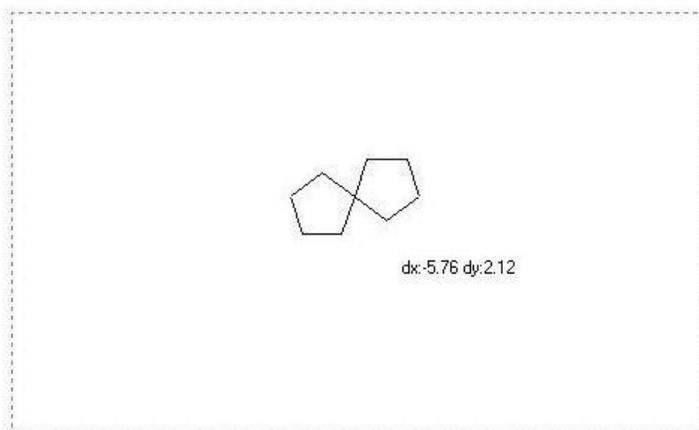


Рисунок 3.6. Об'єднання елементів

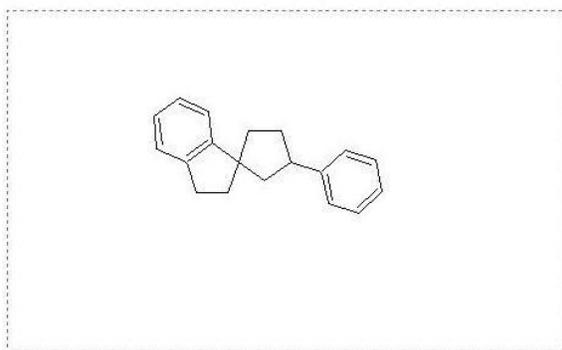


Рисунок 3.7. Додавання бензольного кільця

7. Далі вибрати «кисень» зі списку хімічних елементів, що знаходиться зліва у вікні редактора і помістити його в вузли, щоб вийшла структура.

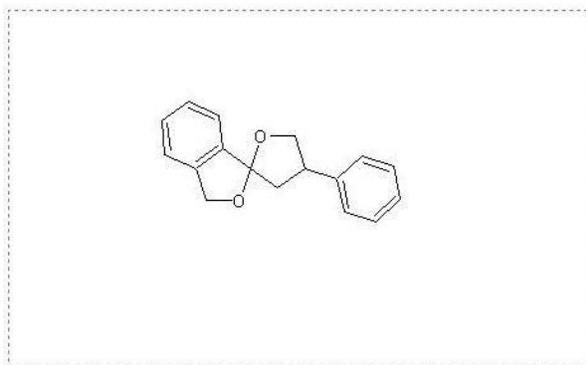


Рисунок 3.8. Додавання атомів кисню

8. Для додавання двохвалентного зв'язку з киснем можна спочатку додати групу CH_2 і потім змінити її на кисень, див. рис. 3.9 і 3.10.

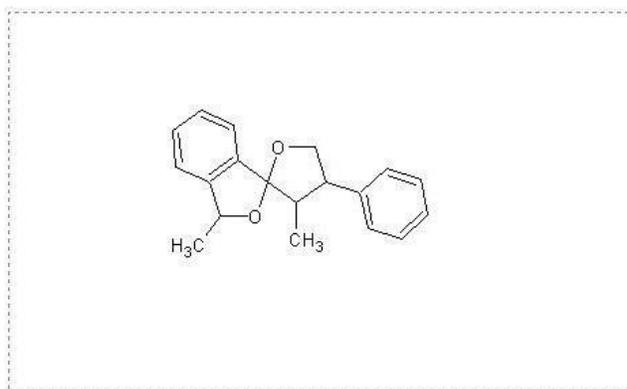


Рисунок 3.9. Додавання зв'язків CH_3

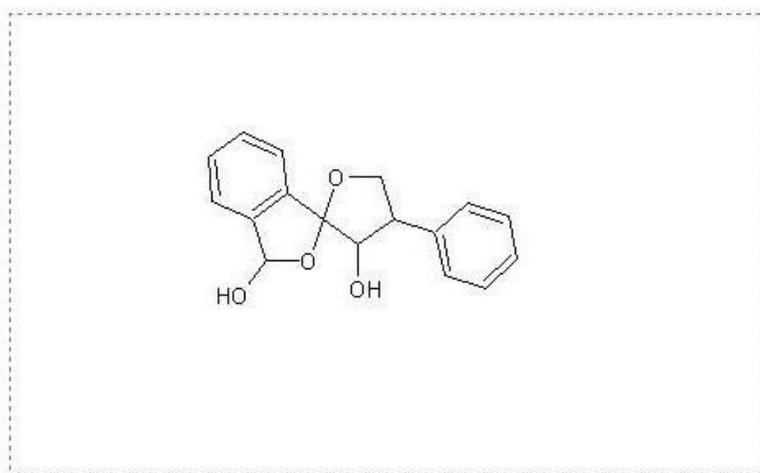


Рисунок 3.10. Зміна на OH - групу

9. Для отримання з гідроксильної групи двохвалентного зв'язку клацнути ще раз по зв'язку, див. рис. 3.11.

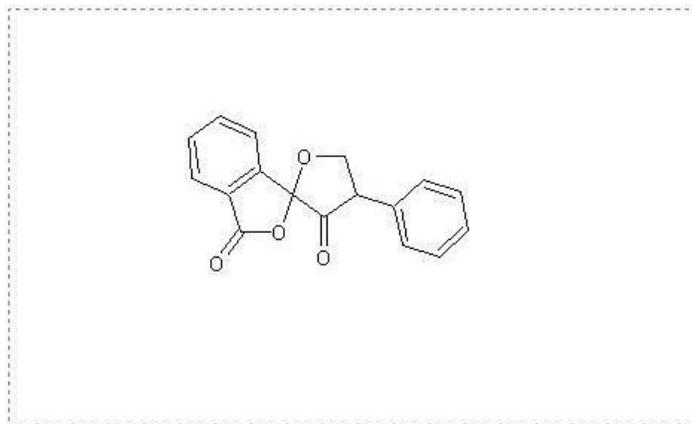


Рисунок 3.11- Остаточна 2D - модель структурної формули

10. Побудувати 3D – модель отриманої структурної формули. Для цього зберегти 2D - модель на диск у файл з розширенням *.mol. За допомогою редактора 3D Viewer відкрити збережену 2D - модель, див. рис. 3.12.

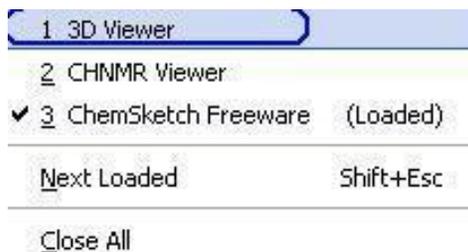


Рисунок 3.12 - Відкриття редактора 3D Viewer

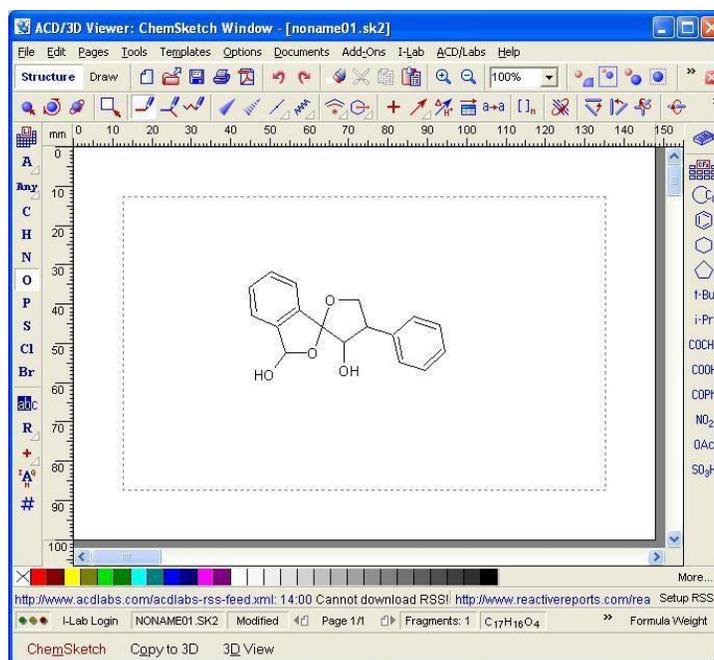


Рисунок 3.13. Копіювання структурної формули в редактор 3D - моделей

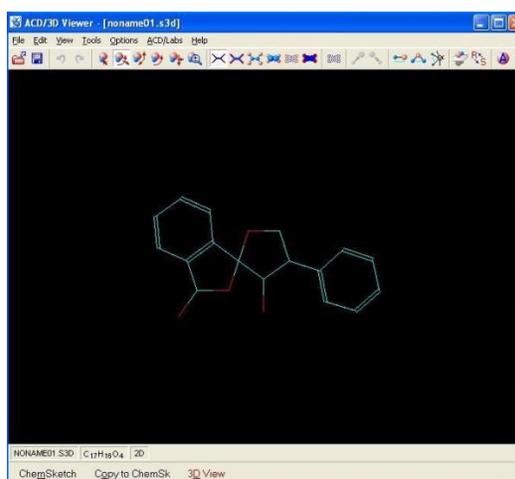


Рисунок 3.14. Тривимірна модель хімічної сполуки

11. Змінюючи вид тривимірної моделі уявити просторову структуру в різних видах, наприклад та зображення на рис. 3.15 і 3.16.

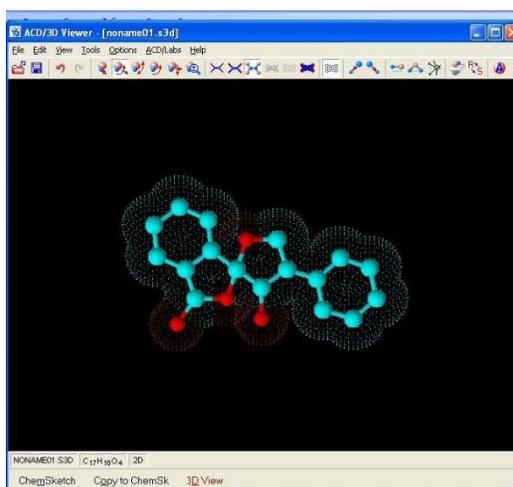


Рисунок 3.15. Модифікація

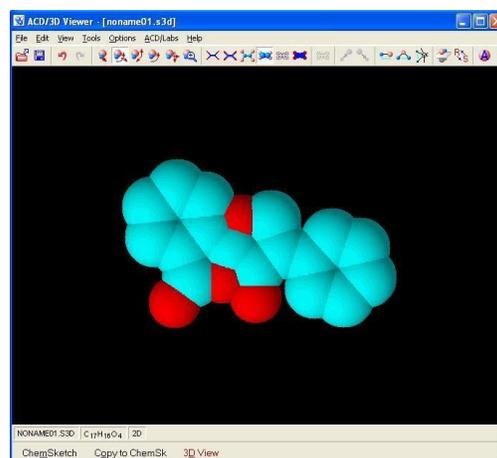


Рисунок 3.16. Модифікація

12. За допомогою інструменту 3D Rotate за допомогою миші повернути модель на деякий кут, див. рис. 3.17.

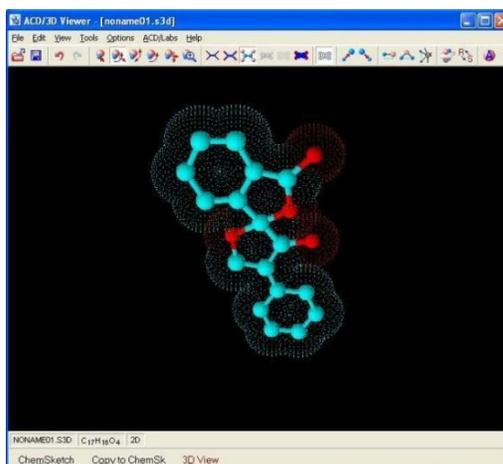


Рисунок 3.17. Поворот на довільний кут

13. За допомогою інструменту Select Atoms помітити атоми кисню зеленим кольором, що знаходяться у вузлах циклопентана, оптимізувати модель за допомогою 3D Optimization і домогтися автоматичного обертання моделі за допомогою інструменту Auto Rotate and Change Style, див. рис. 3.18.

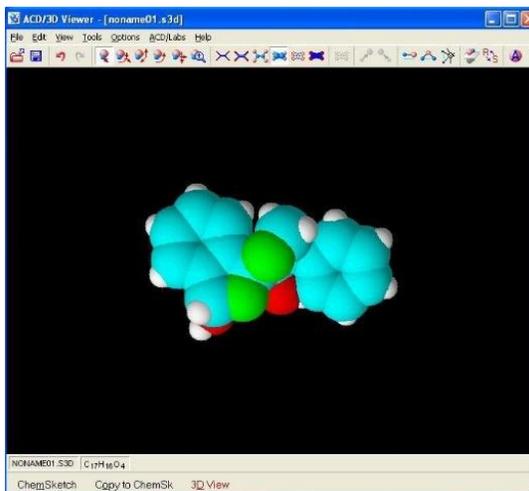


Рис.3.18. Автоматичне обертання з'єднання з виділеними атомами кисню, що знаходяться у вузлах циклопентаном

14. За допомогою інструменту Auto Add Frames додати фрейми і зберегти файл з розширенням gif у вигляді анімації. При цьому задати параметри:

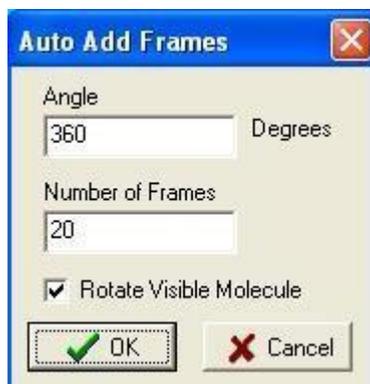


Рисунок 3.19. Автоматичні параметри

15. Змінити значення кута повороту (Angle) і кількість фреймів (Number of Frames) і зберегти під іншим ім'ям. Порівняти отримані моделі.

16. Скопіювати структурну формулу в графічний редактор Paint. Та зберегти файл з розширенням JPG.

При створенні хімічних формул треба керуватися наступними правилами:

- Операції по переміщенню, оптимізації і т.д. можна виробляти і з частиною фігури, попередньо її вибравши. Для виділення частини формули виберіть інструмент Select / Move.

- Щоб «перетворити» одинарну зв'язок в подвійну, виберіть інструмент "Draw normal" і просто клацніть по вибраній зв'язку, так само можна змінити вид зв'язку за допомогою інструментів «Up stereo bond» і «Down stereo bond».

- Для зміни типу атома (наприклад, C на N) просто виберіть новий тип з панелі зліва і клацніть по об'єкту зміни. Уважно вивчивши панель праворуч, таким же чином можна змінити заряд і інші властивості атома.

- Якщо вся ваша структура або її частина виглядають «криво», то використовуйте кнопку "Clean structure" для того щоб привести кути і довжини зв'язків до стандартних значень.

- Активно користуйтеся заготовками, список заготовок можна викликати, натиснувши клавішу F5.

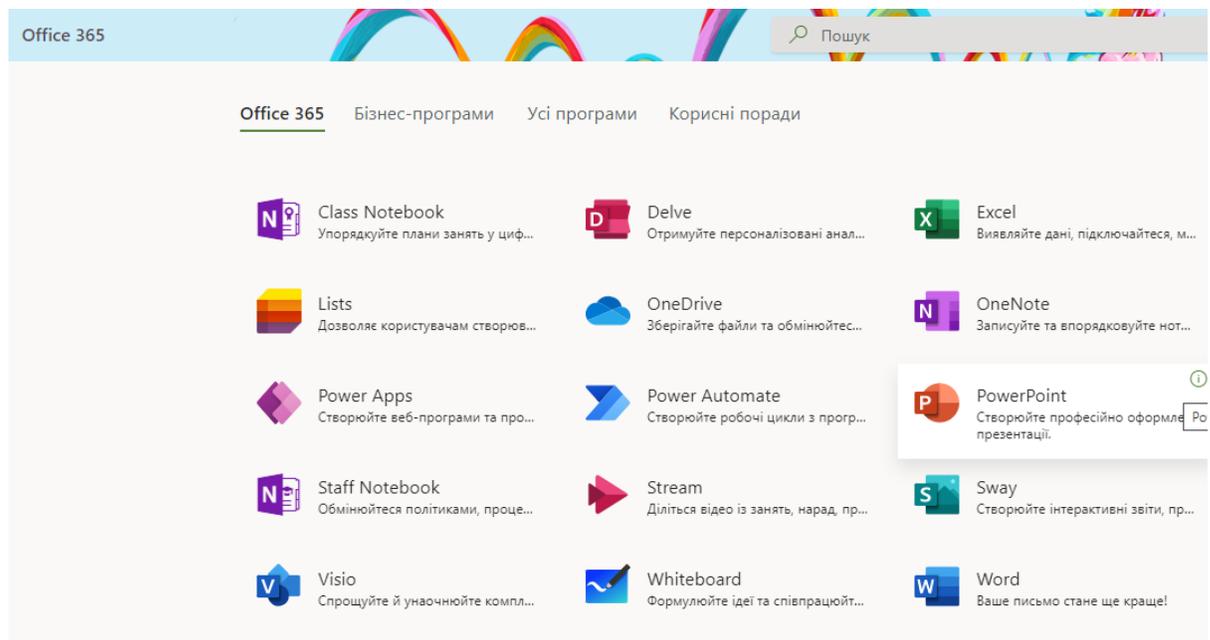
- Для зміни зовнішнього вигляду ковалентних зв'язків в програмі Chemsketch спочатку треба виділити зв'язку. Для цього зручно використовувати інструмент Select / Move при цьому утримуючи клавішу SHIFT. Після того як виділення закінчено, клацніть 2 рази по одній з виділених ковалентних зв'язків. У вікні, що з'явилося зробіть необхідні зміни (Atom style, Bond style) і застосуєте їх (Apply).

- Щоб зробити підписи перейдіть в режим DRAW, виберіть інструмент TEXT (праворуч), для редагування тексту натисніть на білий куточок і виберіть T.

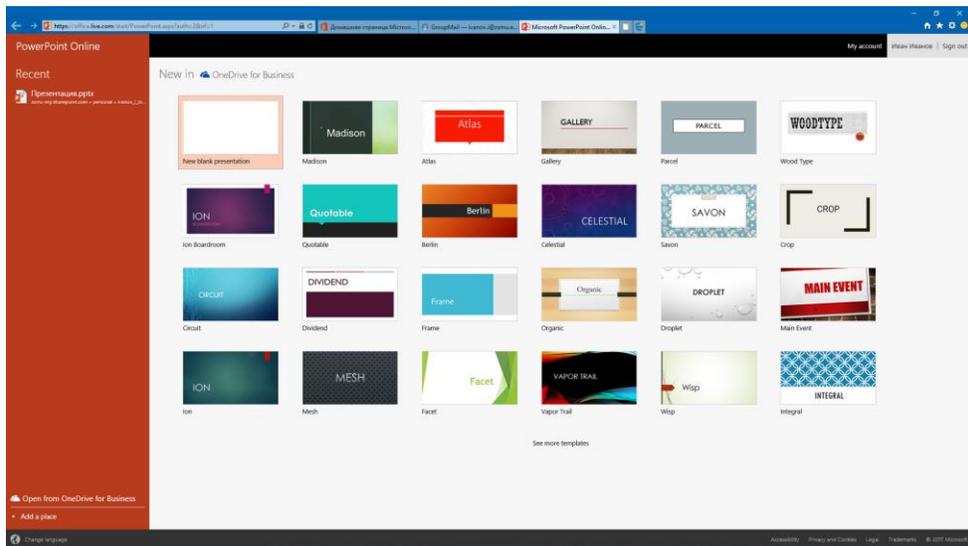
- Для створення нової сторінки виберіть пункт меню Pages-New.
- Для переходу в 3D режим виберіть пункт меню ACD / Labs - 3D Viewer.
- Атоми вибираються з лівого меню (і вставляються клацанням миші), зв'язку простягаються при утриманні лівої кнопки миші.
- Щоб перетворити одинарну зв'язок в подвійну, клацніть по цій зв'язку; такі клацання повернути її в одинарну.

Створення презентації за допомогою програми PowerPoint

Для подальшого виконання завдання відкрийте вкладинку «Головна» та виберіть «PowerPoint».



1. Оберіть дизайн презентації.



2. Створить декілька слайдів.

У презентацію необхідно включити малюнки, графіки, таблиці і т. д., використувані в лекції.

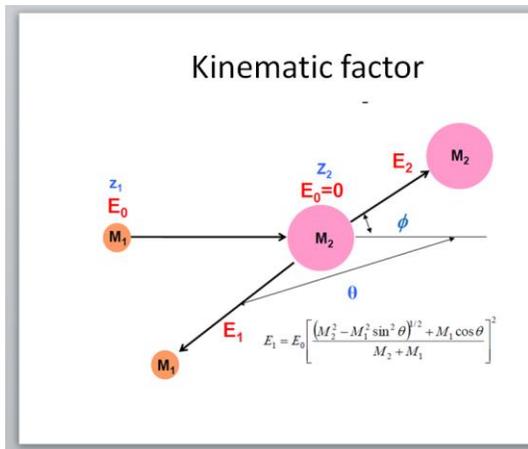


Рис.3.21 Приклад матеріалів у форматі презентації PowerPoint

Рис. 3.22 Приклад матеріалів у форматі презентації PowerPoint

Technologies	Necessary pressure,
Electron beam processing:	
Welding	10^{-3}
Dimension processing, cutting	10^{-3}
Melting, zone-refined	10^{-3}
Electron lithography	10^{-5}
Ion treatment:	
Cleaning	10^{-2}
Etching	10^{-2}
Implantation	10^{-3}
Ion lithography	10^{-5}
Growing of single crystals	
	10^{-4}
Deposition of thin films:	
Deposition of passive elements	10^{-3}
Deposition of optical elements	10^{-6}
Surface quality control:	
Scanning microscopy	$10^{-8} - 10^{-10}$
Atomic force microscope	$10^{-8} - 10^{-10}$
Assembly of photoelectronic devices (night-vision device)	10^{-10}

Рис.3.23 Приклад матеріалів в форматі презентації PowerPoint

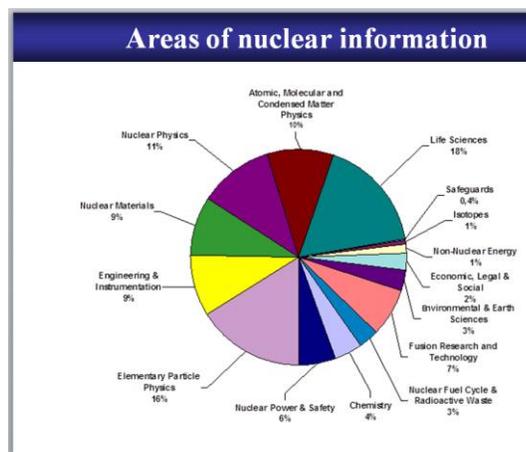


Рис.3.24 Приклад матеріалів в форматі презентації PowerPoint

3. Надішліть посилання на свою презентацію викладачеві у команду академічної групи, що створена у додатку Teams від Office 365. При виконанні презентації враховуйте критерії оцінювання створеної презентації, що наведені викладачем у завданні Teams.

Питання для самоконтролю

6. Дайте характеристику різновидів навчальних занять, що проводяться за допомогою засобів телекомунікацій та інших можливостей Internet
7. Назвіть переваги дистанційного навчання
8. Назвіть недоліки дистанційного навчання
9. Назвіть можливі компоненти презентації MS PowerPoint?
10. Основні елементи інтерфейсу програми ACD/ChemSketch

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ №4

Тема: статистичні методи обробки результатів фармацевтичних досліджень за допомогою електронних таблиць MS Excel

Мета роботи: придбання навичок обробки та узагальнення хіміко-фармацевтичних даних, а також навичок роботи зі статистичними функціями Excel.

Теоретичні відомості

Основні відомості про програму MS Excel

MS Excel- програма для розрахунків за допомогою електронних таблиць. Дана програма працює з таблицями, що складаються з осередків (полів), упорядкованих по рядках і стовпцях (колонок), в яких може міститися різноманітна інформація. Основним об'єктом, з яким працює редактор, є робоча книга. Робоча книга складається з листів. Кількість листів в книзі обмежена обсягом доступної оперативної пам'яті (за замовчуванням 3 листа). Робочий простір листа представлено у вигляді таблиці, що складається з стовпців і рядків. Кожному стовпцю присвоєно назву у вигляді латинських букв або їх поєднань (A, B, .. Z, AB ... AZ). Кожному рядку присвоєно номер від 1 до максимального значення (в різних версіях - по-різному). На перетині рядків і стовпців розташовані осередки. Максимальний розмір листа в версії 2010 дорівнює 1 048 576 рядків і 16 384 стовпчика. Адресація до осередків здійснюється шляхом вказівки заголовка стовпця і рядка. Наприклад, адреса рядка, що знаходиться в стовпці M і рядку 15 буде мати вигляд: M15.

Діапазон- це набір з двох або більше комірок робочого аркуша, які можуть бути як суміжні, так і несуміжні. Адреса діапазону задається адресами лівої верхньої комірки діапазону і правої нижньої комірки, розділених знаком «:» (двокрапка). Якщо ж діапазон має довільну форму, то його адреса задається

адресами складових його прямокутних піддіапазонів, розділених знаками ";" (крапка з комою). Наприклад, «B6: D10» або «H7: J8; G10: I12 ». Діапазони в Excel дуже важливі, так як багато операцій, такі як копіювання, переміщення, форматування та ін., Можуть виконуватися не тільки з окремими осередками, й з діапазонами. Виділенням називається маркування однієї або декількох осередків з метою подальшого виконання операцій над ними. Виділена область відрізняється іншим кольором осередків

Для виділення діапазону досить позиціонувати курсор миші над кутовим осередком (не має значення, якого кута) виділяється діапазону і при натиснутій лівій кнопці миші перемістити його до протилежної (по діагоналі) осередку цього діапазону. Щоб виділити діапазон довільного виду, необхідно, утримуючи клавішу Ctrl виділяти за допомогою миші (як було описано вище) прямокутні діапазони і окремі осередки, з яких буде складатися цей діапазон. Формули - це основа обчислень за допомогою MS Excel. Будь-яка формула в електронних таблицях починається зі знака рівності «=». Якщо не ввести знак рівності, то вводяться дані будуть інтерпретуватися програмою як текст або число.

Після введення формули, в рядку формул буде відображатися вираз формули, а в осередку - результат. Оператором називають знак або символ, що задає тип обчислення у формулі. Існують математичні, логічні оператори, оператори порівняння та посилань. Формули можна вводити з використанням клавіатури і миші. З використанням клавіатури вводять оператори (знаки дій), константи (переважно числа) і, іноді, функції. З використанням миші виділяють осередки, що включаються в формулу. Адреси осередків (посилання) також можна вводити з клавіатури, обов'язково в англійській розкладці.

Обчислення у формулі виконуються в певному порядку. Для зміни порядку проходження операцій потрібно використовувати круглі дужки. Дужки завжди повинні бути парними, тобто кожної відкриває (лівої дужки) у формулі мала відповідати закриває права дужка. Щоб скористатися вмістом осередки або групи

осередків у формулі вводять їх адреси. Адреса комірки (посилання) визначає осередок або групу осередків в межах робочої книги. Результат обчислення формули, що містить адреси осередків, залежить від їх вмісту. При зміні вмісту змінюється і результат. Ті осередки, адреси яких містяться у формулі, в свою чергу можуть містити формули або адреси інших осередків. Таким чином, за допомогою електронних таблиць можна побудувати складні ієрархічні обчислення.

Посилання бувають декількох типів:

1. *Відносні*. Посилання такого виду визначають адреси осередків по відношенню до активної осередку, наприклад «A1»;
2. *Абсолютні*. Ці посилання задають адреси осередків у відповідність з їх становищем на робочому аркуші, наприклад «\$ A \$ 1» (тут для фіксації рядки і стовпці використовується знак «\$»);
3. *Змішані*. У таких посилань є як абсолютна, так і відносна частина адреси, наприклад «A \$ 1» або «\$ A1».

Тип наявної посилання можна швидко змінювати за допомогою клавіші [F4]. Різниця між відносними та абсолютними посиланнями істотно, особливо при копіюванні формул в межах робочого листа. При зміні позиції комірки, що містить формулу, абсолютне посилання не змінюється. При копіюванні формули вздовж рядків і вздовж стовпців абсолютне посилання не коригується. Посилання на осередки інших робочих листів активної робочої книги вводяться аналогічно посиланням в межах поточного робочого аркуша. Назва робочого листа відділяється від адреси комірки знаком оклику, наприклад, «Аркуш2! A1». За замовчуванням посилання на осередки в формулах відносні, тобто адреса комірки визначається на основі розташування цього осередку щодо осередку з формулою. При копіюванні комірки з формулою відносна посилання автоматично змінюється.

У редакторі електронних таблиць є можливість вводити тривимірні

посилання. Тривимірні посилання використовуються при необхідності виконання дій з даними з однієї і тієї ж осередки або діапазону комірок на декількох аркушах однієї книги. Тривимірні посилання включає в себе посилання на клітинку або діапазон, перед якою ставляться імена аркушів.

При цьому в формулу включаються всі листи, що зберігаються між початковим і кінцевим іменами, зазначеними в заголовку. Для того щоб ввести тривимірні посилання, потрібно виділити область осередків на першому аркуші, натисніть клавішу [Shift], і не відпускаючи її, вкажіть останній лист тривимірного діапазону. Тривимірні посилання зазвичай використовуються для створення посилань на осередки на інших аркушах, визначення імен і створення формул з використанням наступних функцій: SUM, AVERAGE, CHECK, , MAX, MIN, MODA, IF, MEDIANA, IF, DEVIATION; DISPERSION; SQUARE DEVIATION.

Наприклад, «=СУММ (Лист1: Лист3! C1: C16)».

Також можна використовувати стиль посилань, в якому нумеруються як рядки, так і стовпці (Стиль посилань R1C1). Цей стиль корисний при обчисленні положення стовпців і рядків в макросах. У стилі посилань R1C1 Excel вказує положення осередку буквою «R», за якою йде номер рядка, і буквою «C», за якою йде номер стовпчика. Наприклад, R [-2] C - відносна посилання на осередок, розташований на два рядки вище і в тому ж стовпці, R C [2] - відносна посилання на осередок, розташований на два рядки нижче і на два стовпці правіше, R2C2 - абсолютне посилання на комірку, розташовану в другому рядку і в другому стовпці, R [-1] відносна посилання на рядок, розташований вище поточної комірки, R - абсолютне посилання на поточний рядок.

Функції Excel- це спеціальні, заздалегідь створені формули, які дозволяють легко і швидко виконувати складні обчислення. Їх можна порівняти зі спеціальними клавішами на калькуляторах, призначених для обчислення квадратних коренів, логарифмів і ін. Excel має кілька сотень вбудованих функцій, які виконують широкий спектр різних обчислень. Деякі функції є еквівалентами

довгих математичних формул, які можна зробити самому. А деякі функції у вигляді формул реалізувати неможливо. Функції складаються з двох частин: імені функції і одного або декількох аргументів. Ім'я функції, наприклад, СУММ, - описує операцію, яку ця функція виконує. Аргументи задають значення або осередки, які використовуються функцією. У формулі, наведеній нижче: СУМ - ім'я функції; В 1: В5 - аргумент. Дана формула підсумовує числа в осередках В1, В2, В3, В4, В5.

= **SUM (B1: B5)**

аргумент функції укладений в круглі дужки. Відкрита дужка відзначає початок аргументу і ставиться відразу після імені функції. У разі введення пробілу або іншого символу між ім'ям і відкриває дужкою в осередку буде відображено помилкове значення # ИМЯ? Деякі функції не мають аргументів. Навіть в цьому випадку функція повинна містити круглі дужки:

= **PI ()**

При використанні в функції декількох аргументів вони відокремлюються один від одного крапкою з комою. Наприклад, наступна формула вказує, що необхідно перемножити числа в осередках А1, А3, А6:

= **MULTIPL (A1; A3; A6)**

У функції можна використовувати до 30 аргументів, якщо при цьому загальна довжина формули не перевищує 1024 символів. Будь аргумент може бути діапазоном, що містить довільне число осередків листа. наприклад:

= **СУММ (A2: A5; B4: B8)**

Посилання на засланні осередки в свою чергу можуть містити формули, які посилаються на інші осередки або діапазони. Використовуючи аргументи, можна легко створювати довгі ланцюжки формул для виконання складних операцій. У наведених раніше прикладах всі аргументи були посиланнями на комірки або діапазони. Але в якості аргументів можна також використовувати числові,

текстові та логічні значення, імена діапазонів, масиви і помилкові значення. Деякі функції повертають значення цих типів, і їх надалі можна використовувати в якості аргументів в інших функціях. Аргументи функції можуть бути числовими. Наприклад, функція СУММ в такій формулі підсумовує числа 24, 987, 49:

=SUM (24; 987; 49)

Статистичні функції MS Excel

Застосування статистичних функцій полегшує користувачеві статистичний аналіз даних. Кількість доступних статистичних функцій в сьомій версії програми збільшилася, і можна стверджувати, що за спектром доступних функцій Excel сьогодні майже не поступається спеціальними програмами обробки статистичних даних. Для того щоб мати можливість використовувати всі статистичні функції, слід завантажити надбудову Пакет аналізу. Основу статистичного аналізу становить дослідження сукупностей і вибірок. Вибірка представляє собою підмножина сукупності. Досліджуючи вибірки за допомогою обчислення відхилень і відстеження взаємозв'язків з генеральною сукупністю, можна простежити, наскільки репрезентативна вибірка. Цілий ряд статистичних функцій Excel призначений для аналізу ймовірностей. Нижче наведено опис деяких найбільш поширених функцій. Інформацію про інші функції користувач може знайти в довідковій підсистемі.

1. AVERAGE

синтаксис: СРЗНАЧ (число1, число2, ...)

результат: Середнє значення (середнє арифметичне) аргументів.

аргументи: число1, число2, ... - числа або імена, масиви або адресні посилання на діапазон комірок, що містить посилання. Функція СРЗНАЧ дозволяє задавати від 1 до 30 аргументів;

2. CHECK

синтаксис: СЧЕТЗ (значення1, значення2, ...)

результат: Кількість всіх значень (будь-якого типу), наведених в якості аргументів.

аргументи: значення1, значення2, ... - не більше 30 аргументів; в матрицях і адресованих діапазонах порожні поля ігноруються.

3. СЧЕСКИФ

синтаксис: СЧЁТЕСЛИ (інтервал; критерій)

результат: Підраховує кількість осередків всередині діапазону, що задовольняють заданому критерію.

Аргументи: інтервал - це одна або кілька осередків посліпль, що включають числа або імена, масиви або посилання, що містять числа. Порожні осередки і текстові значення не враховуються, критерій - критерій у формі числа, виразу, тексту або посилання на осередок, який визначає, які комірки потрібно підраховувати. Наприклад, критерій може бути виражений наступним чином: 32, "32", "> 32", "яблука" або В4. *Зауваження:* в цьому аргументі можна використовувати символи узагальнення: знак питання (?) і зірочку (*). Знак питання відповідає будь-якому символу, зірочка - будь-якій послідовності символів. Для пошуку самого знаку або зірочки слід поставити перед ними знак ~ (тільда).

4. ІФ

синтаксис: ЕСЛИ (лог_вираз; значення якщо істина; значення якщо брехня)

результат: Повертає одне значення, якщо задана умова при обчисленні дає значення ІСТИНА, й інше значення, якщо БРЕХНЯ. Функція ІФ використовується при перевірці умов для значень і формул.

аргументи: Лог_вираз - будь-яке значення або вираз, що приймає значення ІСТИНА або БРЕХНЯ. Наприклад, А10 = 100 - логічне вираження; якщо значення в клітинці А10 дорівнює 100, це вираз приймає значення ІСТИНА, а в іншому

випадку - значення БРЕХНЯ. Цей аргумент може використовуватися в будь-якому операторі порівняння. Значення якщо істина - значення, яке повертається, якщо аргумент «лог_вираз» має значення ІСТИНА. Наприклад, якщо даний аргумент - рядок «В межах бюджету», а аргумент «лог_вираз» має значення ІСТИНА, то функція IF відобразить текст «У межах бюджету». Якщо аргумент «лог_вираз» має значення ІСТИНА, а аргумент «значення якщо істина» не заданий, повертається значення 0 (нуль). Щоб відобразити слово ІСТИНА, необхідно використовувати логічне значення ІСТИНА для цього аргументу. Аргумент «значення якщо істина» може бути формулою. Значення якщо брехня - значення, яке повертається, якщо «лог_вираз» має значення БРЕХНЯ. Наприклад, якщо даний аргумент - рядок «Перевищення бюджету», а аргумент «лог_вираз» має значення БРЕХНЯ, то функція IF відобразить текст «Перевищення бюджету». Якщо аргумент «лог_вираз» має значення БРЕХНЯ, а аргумент «значення якщо брехня» опущений (Після аргументу «значення якщо істина» відсутня крапка з комою), то повертається логічне значення БРЕХНЯ.

Якщо аргумент «лог_вираз» має значення БРЕХНЯ, а аргумент «значення якщо брехня» порожній (Після аргументу «значення якщо істина» стоїть крапка з комою, а за нею - закриває дужка), то повертається значення 0 (нуль). Аргумент «значення якщо брехня» може бути формулою. яке повертається, якщо «лог_вираз» має значення БРЕХНЯ. Наприклад, якщо даний аргумент - рядок «Перевищення бюджету», а аргумент «лог_вираз» має значення БРЕХНЯ, то функція IF відобразить текст «Перевищення бюджету». Якщо аргумент «лог_вираз» має значення БРЕХНЯ, а аргумент «значення якщо істина» опущений (Після аргументу «значення якщо істина» відсутня крапка з комою), то повертається логічне значення БРЕХНЯ. Якщо аргумент «лог_вираз» має значення БРЕХНЯ, а аргумент «значення якщо брехня» порожній (Після аргументу «значення якщо істина» стоїть крапка з

комою, а за нею - закриває дужка), то повертається значення 0 (нуль).

Зауваження: як значення аргументів «значення якщо істина» і «значення якщо брехня» можна для побудови більш складних перевірок використовувати до 64 вкладених один в одного функцій ЯКЩО. Після обчислення аргументів «значення якщо істина» і «значення якщо брехня», функція ЯКЩО повертає отримане значення. Якщо один з аргументів функції ЯКЩО є масивом (Масив. Об'єкт, який використовується для отримання кількох значень в результаті обчислення однієї формули або для роботи з набором аргументів, розташованих в різних осередках і згрупованих по рядках або стовпцях. Діапазон масиву використовує загальну формулу; константа масиву представляє собою групу констант, використовуваних як аргументи.), при виконанні функції ІФ обчислюються всі елементи масиву. Excel містить додаткові функції, які можна застосовувати для аналізу даних з використанням умов. Наприклад, для підрахунку числа входжень текстового рядка або числа в діапазоні осередків використовуйте функції СЧЕСК.

6. MAX

синтаксис: МАКС (число1, число2, ...)

результат: Найбільше значення в списку аргументів.

*аргументи:*число1, число2, ... - від 1 до 30 чисел, серед яких шукається максимальне значення. Можна задавати аргументи, які є числами, порожніми осередками, логічними значеннями або текстовими уявленнями чисел; аргументи, які є значеннями помилки або текстами, що не перетворюваними в числа, призводять до появи значень помилки. Якщо аргумент є масивом або посиланням, то в ньому враховуються тільки числа. Порожні осередки, логічні значення, тексти або значення помилок в масиві або посиланні ігноруються. Якщо аргументи не містять чисел, то функція МАКС повертає 0.

7. MIN

синтаксис: МИН (число1, число2, ...)

результат: Найменше значення в списку аргументів.

аргументи: число1, число2, ... - не більше 30 аргументів; ігноруються тільки значення помилки і текст, який не може бути перетворений в числа; якщо жоден аргумент не містить чисел, функція МІН повертає 0.

8. DISPERSION

синтаксис: ДИСП (число1, число2, ...)

результат: Дисперсія вибірки. Аргументи розглядаються як вибірка з генеральної сукупності.

аргументи: число1, число2, ... - не більше 30 аргументів; текстові, логічні і порожні поля призводять до помилки.

9. MEDIAN

синтаксис: МЕДІАНА (число1, число2, ...)

результат: Медіана заданої множини чисел (число, яке є серединою безлічі чисел: половина чисел більше, ніж медіана, а половина чисел менше, ніж медіана).

аргументи: Число1, число2, ... - числа або імена, масиви або адресні посилання на діапазон комірок, що містить посилання.

10. MODA

синтаксис: МОДА (число1, число2, ...)

результат: Найбільш часто зустрічається значення в масиві або інтервалі даних. Так само, як і функція МЕДІАНА, функція МОДА є мірою взаємного розташування значень.

аргументи: число1, число2, ... - від 1 до 30 аргументів, для яких обчислюється функція МОДА; в функції МОДА можна використовувати замість аргументів масив або посилання на масив.

11. DEVIATION

синтаксис: СТАНДОТКЛОН (число1, число2, ...)

результат: Оцінка стандартного відхилення за вибіркою. Стандартне відхилення - це міра того, наскільки широко розкидані точки даних відносно свого середнього.

аргументи: число1, число2, ... - від 1 до 30 числових аргументів, відповідних вибірці з генеральної сукупності.

Примітка: Використовуйте цю функцію, щоб обчислити стандартне відхилення генеральної сукупності на підставі вибірки.

12. T.TEST

синтаксис: T.TEST (масив1; масив2)

результат: Повертає результат F-тесту. F-тест повертає двосторонню ймовірність того, що різниця між дисперсіями аргументів «масив1» і «масив2» несуттєва. Ця функція дозволяє визначити, чи мають дві вибірки різні дисперсії. Наприклад, якщо дані результати тестування для приватних і громадських шкіл, можна визначити, чи мають ці школи різні рівні розкиду результатів тестування.

аргументи: Масив1 - перший масив або інтервал даних. Масив2 - другий масив або інтервал даних.

Зауваження : Аргументи можуть бути або числами, або містять числа іменами, масивами або посиланнями. Якщо аргумент, який є масивом або посиланням, містить текст, логічні значення або порожні клітинки, ці значення ігноруються; осередки, які містять нульові значення, враховуються. Якщо кількість точок даних в аргументі «масив1» або «масив2» менше 2 або якщо дисперсія аргументу «масив1» або «масив2» має нульове значення, функція ФТЕСТ повертає значення помилки # СПРАВ / 0!

13.T.TEST

синтаксис: T.TEST (массив1; массив2; хвосты; тип)

результат: Повертає ймовірність, відповідну критерієм Стьюдента. Функція TTEST дозволяє визначити, ймовірність того, що дві вибірки взяті з генеральних сукупностей, які мають один і той же середнє.

аргументи: Массив1 - перша безліч даних. Массив2 - друга безліч даних. Хвосты - число хвостів розподілу. Якщо хвосты = 1, то функція TTEST використовує односторонній розподіл. Якщо хвосты = 2, то функція TTEST використовує двостороннє розподіл. Тип - вид виконуваного t-тесту. Здійснюється тест: парний, двухвиборочний з рівними дисперсіями (гомоскедастическій), двухвиборочний з нерівними дисперсіями (гетероскедастичних).

зауваження: Якщо аргументи «массив1» і «массив2» мають різне число точок даних, а тип = 1 (парний), то функція TTEST повертає значення помилки # Н / Д. Аргументи «хвосты» і «тип» усикаються до цілих значень.

Якщо аргумент «хвосты» або «тип» не є числом, то функція TTEST повертає значення помилки # значить !. Якщо аргумент «хвосты» має значення, відмінне від 1 і 2, то функція TTEST повертає значення помилки # ЧИСЛО !. Функція TTEST використовує дані аргументів «массив1» і «массив2» для обчислення невід'ємної t-статистики. Якщо хвосты = 1, TTEST повертає ймовірність більш високого значення t- статистики, виходячи з припущення, що «массив1» і «массив2» є вибірками, що належать генеральної сукупності з одним і тим же середнім. Значення, що повертається функцією TTEST в разі, коли хвосты 2, вдвічі більше значення, що повертається, коли хвосты = 1, і відповідає ймовірності вищого абсолютного значення t-статистики, виходячи з припущення, що «массив1» і «массив2» є вибірками,

Якщо формула або функція містить помилку, яка не дозволяє обчислити результат, Excel відобразить значення помилки. Кожен вид помилки

викликається різними причинами, і вирішуються такі помилки різними шляхами.

Нижче представлені наступні типи помилок:

- Помилка #####.
Дана помилка виникає, якщо стовпець недостатньо широкий, або якщо дата або час є негативними числами;
- Помилка # СПРАВ / 0!
Дана помилка виникає при розподілі числа на 0;
- Помилка # Н / Д
Дана помилка виникає, якщо значення недоступно функції або формулою;
- Помилка # ІМ'Я?
Дана помилка виникає, якщо Excel не може розпізнати текст у формулі;
- Помилка # ЧИСЛО!
Дана помилка виникає при введенні неприпустимих числових значень в формулу або функцію;
- Помилка # ПОСИЛАННЯ!
Дана помилка виникає, якщо посилання на осередок вказана невірно.
- Помилка # значить!
Дана помилка виникає при використанні неприпустимого типу аргументу або операнда.

Порядок виконання роботи

1. **Робота з формулами.** На новому аркуші редактора MS Excel виконати розрахунок обсягу вибірки для проведення вибіркових досліджень.

Завдання (А): визначити, яку частку становить певний препарат в роздрібному продажі (n). Формула для розрахунку має вигляд:

$$n = \frac{\frac{t_{n,a}^2 P Q}{d^2}}{1 + \frac{1}{N} \left(\frac{t_{n,a}^2 P Q}{d^2} - 1 \right)}$$

Вихідні дані для розрахунку взяти з табл. 4.2.

Дані з табл. 4.2 ввести в незаповнені клітинки робочого листа, побудувати формулу з урахуванням особливостей синтаксису Excel (див. Табл. 1.) для обчислення обсягу вибірки. У формулі використовувати відносні посилання на клітинки з вихідними даними.

Таблиця 4.2 - Вихідні дані завдання (А).

позначення	значення	величина
P	0,4	Можливий розмір оцінюваної частки
Q	1-P	
N	10000	Загальна кількість аптек
d	0,05	Абсолютна гранично допустима помилка у визначенні значення частки
$t_{n,a}^2$	≈ 2	Критичне значення розподілу Стьюдента

Завдання (Б) : визначити середній розмір продажів певного препарату в аптеці, при цьому граничний абсолютне відхилення не повинно перевищувати 2000 грн. Формула для розрахунку має вигляд:

$$n = \frac{\frac{t_{n,a}^2 S^2}{d^2}}{1 + \frac{1}{N} \left(\frac{t_{n,a}^2 S^2}{d^2} - 1 \right)}$$

Вихідні дані для розрахунку взяти з табл. 4.3:

Таблиця 4.3 - Вихідні дані завдання (Б).

позначення	значення	величина
N	1000	Загальна кількість аптек
S ²	30000000	Оцінка вибіркової дисперсії
d	0,05	Абсолютна гранично допустима помилка у визначенні значення
t_n^2 ,	≈2	Критичне значення розподілу Стюдента

Дані з табл. 3 ввести в незаповнені клітинки робочого листа, побудувати формулу з урахуванням особливостей синтаксису Excel (див. Табл. 1.) для обчислення обсягу вибірки. У формулі використовувати відносні посилання на клітинки з вихідними даними.

- Робота зі статистичними функціями Excel для визначення параметрів вибірки.** На новому аркуші редактора MS Excel виконати розрахунок характеристик вибірки: середнє значення, медіану, моду, середнє лінійне відхилення, дисперсію, середньоквадратичне відхилення, мінімальне і максимальне значення. Вихідні дані для задачі дано в табл. 4.4.

Таблиця 4. 4 - Показники фізичного розвитку школярів

№	Маса тіла, кг	Зріст, см	ОГК, см
1	41	150	80
2	45	145	75
3	39	155	81

4	42	151	76
5	38	161	77
6	41	159	79
7	42	153	73
8	43	152	61
9	41	150	81
10	40	154	80

Для обчислення параметрів вибірки використовуйте вбудовані статистичні функції Excel. Для вставки вбудованих функцій можна використовувати Майстер функцій.

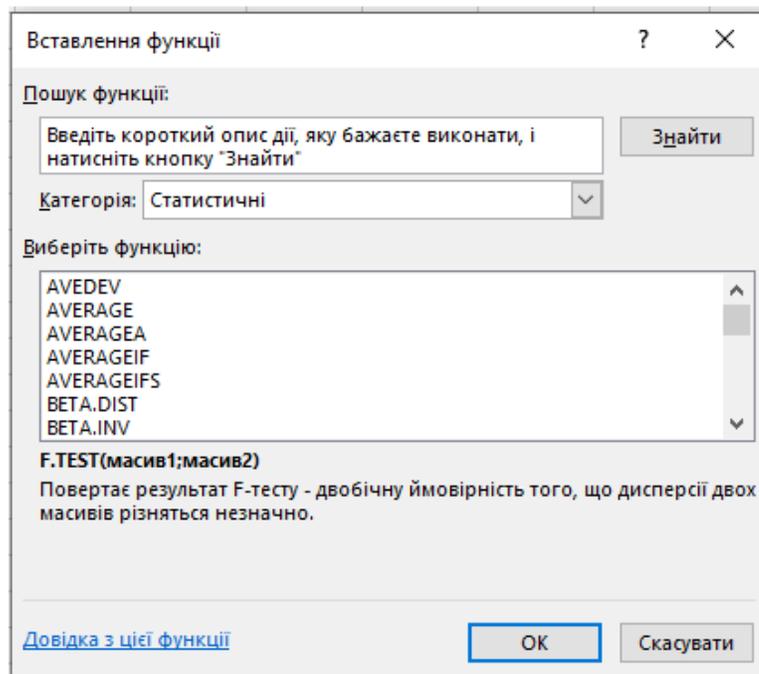


Рис. 4.1. Майстер функцій Крок 1 (вибір категорії функцій і конкретної функції).

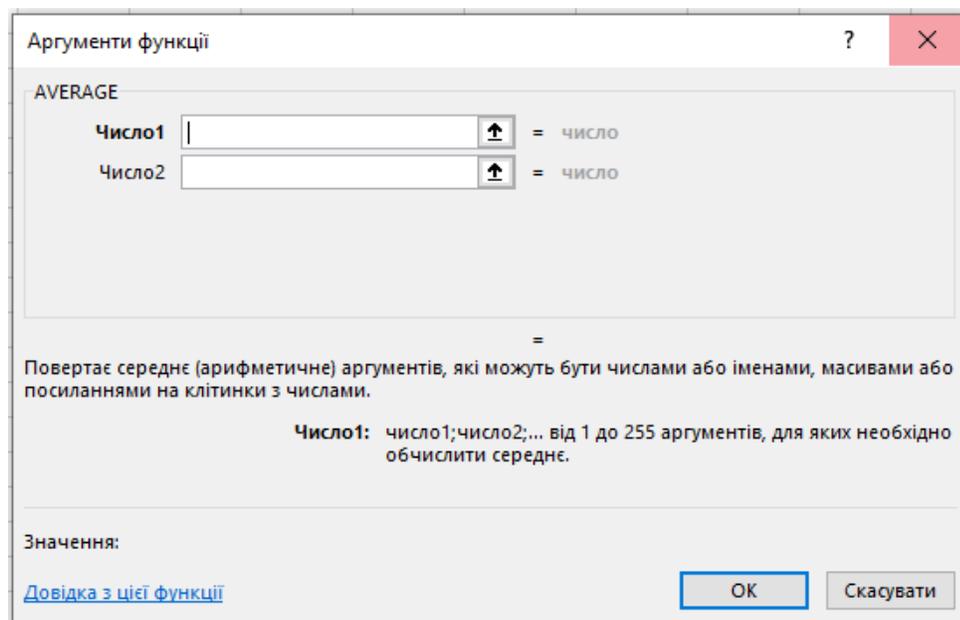


Рис. 4.2. Вставка функції AVERAGE

3. **Параметричні критерії перевірки статистичних гіпотез про середні і дисперсіях.**

Завдання. Досліджували динаміку продажів лікарського препарату для лікування хронічного гепатиту, як з використанням рекламної акції, так і без неї. Для визначення ефективності реклами при продажу лікарського препарату ГЕПТРАЛ® в аптеках області взято дані:

- 15-ти аптек області, які не проводили рекламну акцію (контрольна група);
- 10-ти аптек мережі «ЄВРОАПТЕКА» на 1, 3 і 9-му тижні від проведення рекламної акції (дослідна група).

Дані про динаміку продажів наведені в табл. 4.5.

Необхідно визначити, з точки зору статистики, чи ефективна реклама? Для цього необхідно перевірити середні значення для представлених вибірок і оцінити значимість відмінностей. Перш ніж перевіряти гіпотезу про рівність середніх цих вибірок, необхідно перевірити гіпотезу рівності дисперсій, щоб знати який з критеріїв вибрати для її перевірки (існує кілька варіантів, з рівними

і нерівними дисперсіями, а також без припущень про дисперсії.

Таблиця 4.5 - Вихідні данні

№ п / п	Контрольна група, X0	досвідчена група		
		1-й тиждень, X1	3-й тиждень, X2	9-й тиждень, X3
1	2	26	14	5
2	5	35	12	1
3	4	42	19	5
4	0	25	5	0
5	1	33	18	2
6	7	37	18	2
7	3	35	5	2
8	2	10	10	10
9	5	28	16	1
10	1	31	15	5
11	0			
12	6			
13	3			

Перед перевіркою рівності дисперсій потрібно переконатися, в тому, що дані розподілені по нормальному закону. Визначимо чи можна вважати закон розподілу вибірок (X0, X1, X2, X3) нормальним. Якщо немає, хоча б для однієї пари вибірок, то необхідно використовувати непараметричний критерій. Слід

пам'ятати, що непараметричні критерії мають для випадку нормального розподілу меншу потужність, ніж відповідні параметричні критерії.

Для перевірки гіпотези про нормальний розподіл можна скористатися правилом 3-х сігм, яке говорить про те, що практично всі значення нормально розподіленої випадкової величини знаходяться в інтервалі

$$x - 3\sigma ; x + 3\sigma,$$

де σ - середньоквадратичне відхилення. Більш строго - не менше ніж з 99,7% достовірністю, значення нормально розподіленої випадкової величини лежить в зазначеному інтервалі.

Висновок з цього правила є критерій перевірки на нормальність: якщо майже всі відхилення від середнього (99,7%) менше 3-х сигм ($\epsilon_i < 3\sigma$), то дані можна вважати розподіленими за нормальним законом. Для більш суворої перевірки, необхідно щоб 2/3 відхилень від середнього були менше, ніж σ , і щоб половина відхилень від середнього було менше, ніж $0,625\sigma$.

4. На новому аркуші Excel ввести вихідні дані для завдання (табл. 5.) і перевірити чи нормально розподілена кожна з вибірок. Для цього для кожної з вибірок розрахувати AVERAGE і DEVIATION. Далі для кожного значення кожної вибірки розрахувати абсолютне значення (модуль) відхилення від середнього (ABS). Майже всі (15 значень для контрольний групи і по 10 значень для дослідної групи) відхилення від середнього повинні бути менше потроєного значення СКО. Заповнити таблицю:

Умова	необхідно	0	1	2	3
Майже всі (99,7%) відхилення від середнього менше 3 сигм.	$\approx 15 (X_0)$ і ≈ 10 (X_1, X_2, X_3)				

Для автоматизації розрахунків побудуйте стовпець з різницями $x_i - x$. Далі

скористайтеся функцією IF для ідентифікації тих значень різниці, які менші трьох сигм. Далі скористаєтеся функцією СHECKIF для підрахунку кількості значень, що не суперечать умові. Впишіть результат в таблицю.

Після підтвердження гіпотези про нормальність розподілу досліджуваних вибірок потрібно перевірити рівність дисперсій для пов'язаних вибірок (вибірки X0 і X1, X0 і X2, X0 і X3). Скористаємося критерієм Фішера для кожної пари вибірок. Сформулюємо гіпотези H0 і H1.

1. Нульова гіпотеза (H0): $S12 = S22$ (дисперсії вибірок рівні);
2. Альтернативна гіпотеза (H1): $S12 \neq S22$ (дисперсії вибірок нерівні, двостороння гіпотеза).

Для перевірки гіпотез скористаємося функцією ФТЕСТ, яка повертає двосторонню ймовірність того, що різниця між дисперсіями аргументів «Виборка1» і «Виборка2» несуттєва. Ця функція дозволяє визначити, чи мають дві вибірки різні дисперсії. Заповніть таблицю:

	X0 і X1	X0 і X2	X0 і X3
Результат F-тесту (%)			
Висновок: (Так, рівні чи ні, не рівні)			

При цьому інтерпретувати отриманий результат потрібно наступним чином: якщо результат F-тесту не менше 95%, тоді приймається нульова гіпотеза (H0) (різниця між дисперсіями несуттєва, тобто вони рівні), в іншому випадку приймається альтернативна гіпотеза (H1) (дисперсії нерівні, і ймовірність цього твердження статистично значуща, тобто більше 95%, що досить для медико-біологічних досліджень).

Далі слід перевірити рівність середніх значень пов'язаних вибірок, щоб визначити ефективність реклами після рекламної акції. Скористаємося критерієм Стьюдента (при рівних або нерівних дисперсіях см. Вище).

Сформулюємо гіпотези H_0 і H_1 .

1. Нульова гіпотеза (H_0): $\bar{X}_1 = \bar{X}_2$ (Середні значення вибірок рівні);
2. Альтернативна гіпотеза (H_1): $\bar{X}_1 \neq \bar{X}_2$ (Середні вибірок нерівні, двостороння гіпотеза).

Для перевірки гіпотез скористаємося функцією TTEST, яка повертає ймовірність, відповідну критерієм Стьюдента. Функція TTEST дозволяє визначити, ймовірність того, що дві вибірки взяті з генеральних сукупностей, які мають один і той же середній. Заповніть таблицю:

	X0 і X1	X0 і X2	X0 і X3
Результат Т-тесту (%)			
Висновок: (Так, рівні чи ні, не рівні)			

Для розрахунків за допомогою функції TTEST, потрібно задати два додаткові параметри: хвости і тип тесту. Задати параметр «хвости», рівним 2 (двостороннє розподіл), а параметр «тип»: 2 - тест двохвибірковий з рівними дисперсіями (гомоскедастическими), 3 - тест двохвибірковий з нерівними дисперсіями (гетероскедастичними).

При цьому інтерпретувати отриманий результат потрібно наступним чином: якщо результат Т-тесту не менше 95%, тоді приймається нульова гіпотеза (H_0) (середні рівні), в іншому випадку приймається альтернативна гіпотеза (H_1)

(середні нерівні, і ймовірність цього твердження статистично значуща, тобто більше 95%, що досить для медико-біологічних досліджень).

Загальний результат повинен виглядати наступним чином: вибірки (їх середні і дисперсії) взяті після проведення рекламної акції повинні належати іншій ГС (тобто сукупності мережі аптек в яких проведена рекламна акція, на відміну від вихідної ГС, де рекламної акції поки ще не було) і чим більше часу пройшло після рекламної акції, тим менше повинні відрізнятися вибірки: вихідна (взята з ГС, без проведення рекламної акції) і експериментальна (взята ГС аптек з проведеною рекламною акцією). Т.обр. ефективність рекламної акції повинна підтвердитися в найближчі дні після її проведення, і далі йти на спад, що повинні підтверджувати цифри, отримані в результаті Ф і Т-тестів.

Тобто ймовірність рівності середніх значень (тобто приналежності вибірок однієї ГС) вихідної (X0) і останньої (X3) вибірки повинна бути вже досить високою.

Питання для самопідготовки

1. Для чого використовується редактор електронних таблиць? Особливості проведення розрахунків за допомогою Excel.
2. Яким чином здійснюється адресація до осередків робочого аркуша Excel?
3. Що таке діапазон? Як виділити діапазон комірок в Excel?
4. Перерахувати основні арифметичні і логічні оператори Excel.
5. Чим абсолютне посилання відрізняється від відносного? Від мішаного?
6. Що таке формула в Excel? Як здійснити введення формули?
7. Що таке функція в Excel? Що називається аргументом функції?
8. Перерахувати основні статистичні функції в Excel.
9. Які типи помилок при воді формули бувають?
10. Як інтерпретувати результати роботи функцій ТТЕСТ і ФТЕСТ?

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ № 5

Тема: Комп'ютерні дані: типи даних, обробка та управління. Cache ObjectScript. Робота в термінальному режимі. Оператори введення-виведення, арифметичні оператори, оператори вибору, масиви і рядкові змінні.

Мета роботи: Освоїти основні принципи алгоритмізації. Ознайомитися з правилами написання програм на основі лінійних і розгалужених алгоритмів.

Необхідне програмне забезпечення: Cache Object Script.

Теоретичні відомості

1. Основні елементи Cache Object Script

Cache Object Script - це потужний об'єктно-орієнтований редактор, вбудований в Cache. Синтаксис мови близький до синтаксиса широко відомих мов (Delphi, C ++ і ін.) Хоча є деякі особливості. Налагодження коду може бути виконано за допомогою спеціальної програми «Термінал», яка моментально інтерпретує будь-яку команду Cache Object Script.

У мові розрізняють *локальні і глобальні змінні*. Глобальні змінні (global) зберігаються в базі даних, їх імена починаються з символу «^», а в іншому, робота з ними однакова. У Cache Object Script відсутня декларування змінних: єдиний тип даних - це рядок символів змінної довжини, хоча внутрішніми засобами системи результат арифметичної або логічної операції інтерпретується як число або як логічне значення.

В силу цього, великого значення набувають функції, які створені і створюються для виконання певних дій, в залежності від того, що повинен робити конкретний метод. У Cache Object Script це функції, що маніпулюють з датою, часом, числами, списками, рядками і функції \$ ZUTIL, які виконують конфігурація системи і завдання управління.

2. Основні команди та оператори Caché Object Script:

- команда DO;
- команда SET;
- команда WRITE;
- команда READ;
- команда QUIT;
- команда NEW;
- команда IF;
- команда ELSE;
- команда FOR.

Команди керують виконанням вирішуваних завдань. За всіма командами може слідувати двокрапка (:) і вираз. Такий запис називається постусловієм. Якщо постусловіє істинно, команда виконується; якщо постусловіє помилково, то командане виконується, а управління передається наступній команді.

Наступні команди дозволяють виконувати основні прикладні задачі:

- команда DO;
- команда SET;
- команда WRITE;
- команда READ;
- команда QUIT;
- команда NEW;
- команда IF;
- команда ELSE;
- команда FOR.

Команда DO

Команда DO змушує Caché запустити наступний за нею рядок або сукупність рядків, укладених в {}. Якщо метод не повертає значень, або ці

значення не важливі, можна використовувати команду DO для запуску методів.

Команда SET

Команда SET використовується для присвоєння значень змінним. Вона може привласнювати значення однієї або декількох змінним одночасно. Щоб привласнити значення однієї змінної, використовуйте наступний синтаксис:

SET variable = <expression>

Присвоїти значення відразу декільком змінним можна наступним чином:

SET (X1, x2, ..., xN) = <expression>

Команда WRITE

Команда WRITE може бути використана як з аргументом, так і без нього. Без аргументу команда WRITE виводить на використовуваний пристрій виведення імена всіх змінних локального середовища і їх значень. Якщо в якості аргументу команди WRITE задано правильне вираження, то на пристрій виведення буде виведено значення цього виразу.

Наприклад, щоб відобразити значення змінної x, необхідно записати:

WRITE x

Таблиця 5.1 - Оператори форматного виведення.

код форматування	ефект
!	Починає новий рядок
#	Починає нову сторінку. На терміналі очищує поточний екран і позиціонує курсор у верхній лівий кут екрану
? n	Переводить курсор на n колонку, де n - натуральне число

Команда READ

Команда READ дозволяє приймати і зберігати дані, введені користувачем з поточного пристрою введення. Команда READ повинна виглядати так:

READ формат, рядок, змінна

де формат визначає, чи буде вводиться користувачем інформація, яка відображується на екрані, рядок з'явиться на екрані перед запрошенням введення, а змінна збереже введені дані. Щоб визначити область введення для користувача, використовуються наступні коди.

Команда QUIT

Команда QUIT завершує процес, запущений командами DO або FOR або завершує виконання внутрішньої функції або методу.

Команда QUIT без аргументу завершує виконання команд DO або FOR, функцій або методів, які не повертають значення, і передає управління наступному рядку коду.

Команда QUIT з аргументом працює тільки всередині функцій і методів і повертає значення аргументу.

Примітка: В Cache Object Script немає старшинства операцій. Вираз обчислюється зліва направо.

Команда NEW

Команда NEW ініціалізує змінні. Всі змінні, використовувани в методі, слід формувати за допомогою цієї команди.

Команда IF

Команда IF визначає, істинно або хибно значення виразу, яке йде безпосередньо за нею. Якщо вираз істинний, то виконується залишок рядка. Якщо вираз помилковий, залишок рядка не виконується, а управління передається наступному рядку коду.

Наступний приклад попереджає спробу поділу на 0:

IF y = 0 WRITE "Не можна ділити на 0".

У цьому прикладі команда WRITE виконується, тільки якщо $y = 0$.

Команда ELSE

Команда ELSE визначає код, який виконується в разі, якщо вираз команди IF є помилковим.

Наприклад, можна використовувати команди IF і ELSE для визначення нахилу(Slope) лінії:

IF x = 0 SET Slope = "не визначена чоловіча"

ELSE SET Slope = y / x

Зверніть увагу, що між ELSE і SET два пробіли!

3. Оператори

Оператори визначають дію, яка виконується над одним або декількома виразами. В системі Caché є безліч операторів, що виконують арифметичні операції.

Основні оператори:

- арифметичні оператори (+, -, =, /, \, **, #);
- оператори порівняння чисел (>, <);
- оператор конкатенації рядків (_);
- оператор «НІ» (`);
- оператор відповідності шаблону (?).

Список операторів наведений у таблиці 5.2.

Таблиця 5.2 - Арифметичні оператори

оператор	значення
+	Якщо використовується з одним операндом, то операнд обчислюється як числове вираження. Якщо вираз не може бути обчислено, повертає 0. При використанні з двома операндами складає їх числові значення.
-	Якщо використовується з одним операндом, то операнд обчислюється як числове вираження, потім множиться на -1. Якщо вираз не може бути обчислено, повертає 0. При використанні з двома операндами віднімає їх числові значення
=	Порівнює 2 операнда і повертає True, якщо вони рівні, і False в іншому випадку. Оператор порівняння може бути використаний як для чисел, так і для рядків
*	Повертає добуток двох операндів. Якщо один з операндів НЕ обчислюється як число, його значення приймається рівним 0, в результаті також виходить 0
/	Поділ першого операнду на другий. Повертає помилку, якщо другий операнд дорівнює нулю.
\	Цілочисельний поділ першого операнда на другий. Повертає помилку, якщо другий операнд дорівнює нулю.
**	Зводить перший операнд в ступінь другого операнда.
#	Повертає залишок від ділення першого операнда на другий.

У Caché є оператори, які порівнюють числові значення.

4. **Cache Terminal**

Cache Terminal - це програма емуляції ASCII терміналу, що поставляється разом з Caché. Ця програма може бути з успіхом використана для вивчення моделей даних, підтримуваних Caché, особливостей використання мови Caché Object Script, тому що вона негайно виконує будь-яку команду Caché Object Script, записану в командному рядку:

USER>

де USER - це ім'я області Caché (Name Space), в якій ви працюєте.

Ви можете змінити ім'я області, використовуючи команду Caché Object Script: ZNSPACE "вираз" або ZN "вираз"

Наприклад, команда

USER>ZN "SAMPLES"

виконується в області USER, негайно призводить до зміни області:

SAMPLES>

тобто ви тепер будете працювати в області SAMPLES.

Можна запустити на виконання фрагмент з декількох команд Caché Object Script, але всі вони повинні бути записані в одному рядку. Редагувати команди у вікні терміналу досить складно. Тому записувати команди в рядку довші ніж 80 символів не рекомендується.

Якщо ви хочете виконати якусь послідовність команд, то краще оформити їх як програму, використовуючи засіб Caché Studio.

Клавіатурний інтерфейс Caché Terminal

Клавіатурний інтерфейс програми Caché Terminal дуже примітивний. У режимі емуляції реального терміналу VT 320 працюють такі клавіші:

Таблиця 5.4 - Корисні поєднання клавіш для роботи в програмі «Terminal»

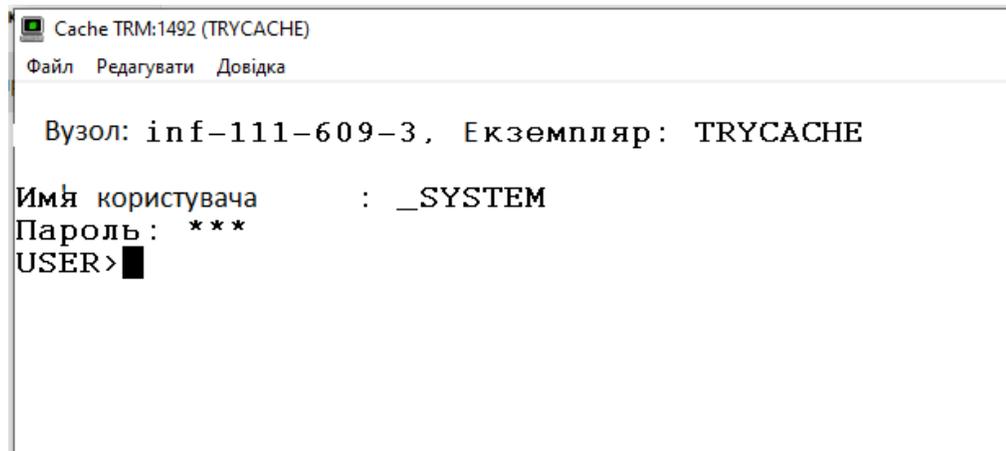
Клавіша	Дія
<Insert>	insert / override
<Delete>	Знищити літеру під курсором
<Home>	Перейти на початок рядка
<End>	Перейти в кінець рядка
<Up Arrow>	Вниз по стеку команд сеансу
<Down Arrow>	Вгору по стеку команд сеансу
<Left Arrow>	Вліво на символ
<Right Arrow>	Вправо на символ
<Ctrl>, I	додати пробіл
<Ctrl>, L	Знищити від курсора до кінця рядка
<Ctrl>, U	Знищити від курсора до початку рядка
<Ctrl>, W	Знищити від курсора до кінця слова
<Ctrl>, G	Знищити від курсора до початку слова
<Ctrl>, X	Очистити рядок
<Ctrl>, S	Заблокувати висновок на екран
<Ctrl>, Q	Розблокувати висновок на екран

Порядок виконання роботи

1. Відкрити термінал Cache з меню Cache Куба, розташованого в правому нижньому кутку на панелі задач. Він має вигляд, показаний на рис. 1.

Програма Caché Terminal буде використовуватись нами у подальшому для налагоджування програм, процедур, функцій та перевірки роботи окремих операторів. Робота в Terminal проводиться в режимі командного рядка, після запрошення треба набрати команду та натиснути ENTER. Тут же виводиться

результат виконання програми.



```
Cache TRM:1492 (TRYCACHE)
Файл Редагувати Довідка

Вузол: inf-111-609-3, Екземпляр: TRYCACHE

Имя користувача      : _SYSTEM
Пароль: ***
USER>
```

Рисунок 5.1 - Зовнішній вигляд програми «Термінал»

2. На прикладах, показаних нижче, вивчити основні команди і конструкції мови Cache.

2.1. Набрати команду Write і переконатися, що в даний момент в пам'яті не існує змінних.

```
USER> Write
```

```
USER>
```

2.2. Присвоїти змінним a і b відповідні значення:

```
USER> Set a = 125.1, b = "Текстова величина"
```

Зверніть увагу на те, що при клавіатурному вводі числових значень відокремлення цілої частини числа від дробової здійснюється за допомогою КРАПКИ, а не коми.

2.3. Переконатися в тому, що значення змінним присвоєно із використанням команди Write:

```
USER> Write a
```

```
125.1
```

```
USER> Write b
```

```
Текстова величина
```

2.4. Вивести на екран значення всіх змінних:

```
USER> Write
```

a = 125.1

b = "Текстова величина"

2.5. Поміняти місцями значення змінних a і b:

```
USER> Set c = a, a = b, b = c
```

2.6. Переконайтеся в тому, що значення змінних дійсно змінилися:

```
USER> Write a
```

Текстова величина

```
USER> Write b
```

125.1

2.7. Попередня дія виконувалася з використанням допоміжної змінної «c», яка в даний момент вже не потрібна. Видаліть її і переконайтеся, що вона дійсно вилучена.

```
USER> Kill c
```

```
USER> Write
```

a = "Текстова величина"

b = 125.1

2.8. Експериментуйте з функціями форматного виведення, використовуючі коди з таблиці 2:

```
USER> Write !, "Список ліків",!, "Анальгін",!, "Парацетамол",!, "Валідол"
```

Список ліків

Анальгін

Парацетамол

Валідол

2.9. Повторіть попередній пункт, але зробіть вивід списку препаратів на новому аркуші, використовуючі коди з таблиці 2:

USER> Write #, "Список ліків",!, "Анальгін",!, "Парацетамол",!, "Валідол"

На новій сторінці ви отримаєте наступне:

Список ліків

Анальгін

Парацетамол

Валідол

2.10. Змінити попередній приклад наступним чином:

USER> Write #, "Список ліків",!,?10, "Анальгін",!,?10, "Парацетамол"

Список ліків

-----Анальгін

-----Парацетамол

-----Валідол

2.11. Побудуйте елементи таблиці (всього 30 позицій)

USER> set gorizontline=" _____"

USER>set name="! Sidorenko ! Ivanenko ! Petrov !"

USER>set vertline="! ! ! !"

USER>write gorizontline, !, name, !, gorizontline, !, vertline, !, vertline

Результатом такого коду буде побудована на екрані таблиця (див. рис.5.2).

```
!  sidorenko  !  Ivanenko  !  Petrov!  
-----  
!  
!  
!  
!  
USER>
```

Рисунок 5.2 – Таблиця, побудована з окремих елементів COS

2.12. Попрацюйте з командою Read:

USER> Read "Введіть найменування препарату:", C

Введіть найменування препарату: *Валідол*

USER> Write C

Валідол

Таким чином змінна C містить значення, яке було введено користувачем.

2.13. Задайте час очікування 10 сек для введення користувачем одного символу:

*USER> Read "Введіть один символ", * d: 10*

Надрукуйте будь-який символ самостійно.

2.14. Дайте змінним наступні значення: $a = 5$, $b = 7$, $c = 10$ і зробіть обчислення, наведені у таблиці 6, користуючись операторами SET і WRITE.

Таблиця 5.6– Завдання для обчислення за допомогою математичних операторів

Вираз, який треба обчислити
$a + b$
$a * b - c$
c / a
$a * b + 2$
$2 + a * b$
$(a+b)/c+2$
$a+(b/(c+12))$
Залишок від ділення 3700 на 3600
Залишок від ділення c на a
Цілочисельне ділення 9,121 на 1

Цілочисельне ділення 3700 на 3600
2^5
9 0,5
16 0,25
4^{-2}

Примітка: тут вперше використаний скорочений синтаксис COS, який дозволяє використовувати команди, що починаються з їхньої першої букви, наприклад замість «*Write*» можна друкувати «*w*» і т.д.

Під час виконання цього завдання пам'ятайте про існування пріоритету виконання математичних дій у складних математичних виразах. Використовуйте круглі дужки () для розставлення пріоритету дій у складних виразах.

2.15. Робота з символічними змінними. **Символьна змінна** - це змінна, значенням якої є рядок символів або будь-нечислове значення, укладене в подвійні лапки.

USER> s x = "Символьна змінна"

USER> w x

Символьна змінна

2.16.Збережіть на жорсткий диск значення змінних. Для цього використовується знак циркумфлекс «^»:

USER> s ^ User.x = "Іванов Олексій"

2.17.Переконайтеся в успішному збереженні змінної за допомогою порталу управління системою. Відкрийте пункт «Портал управління» з меню Cache – куба та виконайте покрокові дії, наведені на рис.5.3 – 5.7.

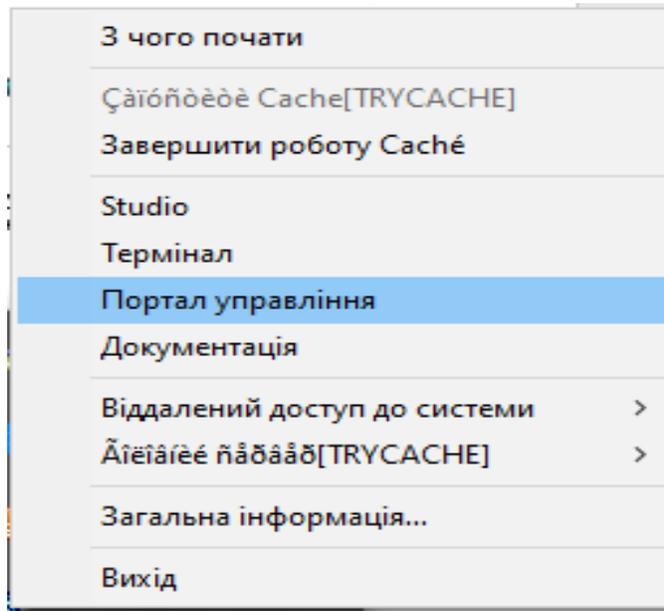


Рисунок 5.3 - Портал управління системою

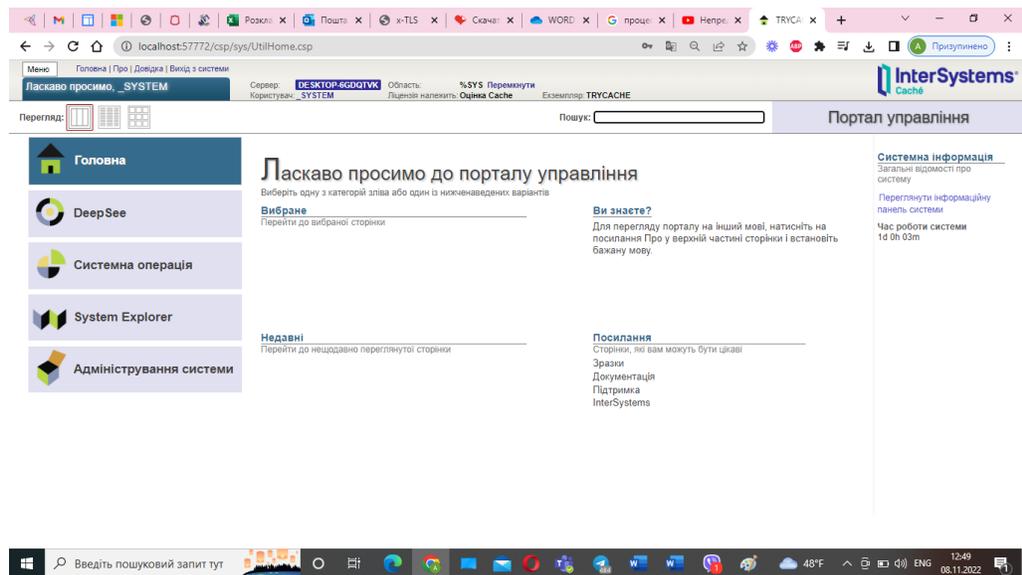


Рисунок 5.4 – Зовнішній вигляд порталу управління системою

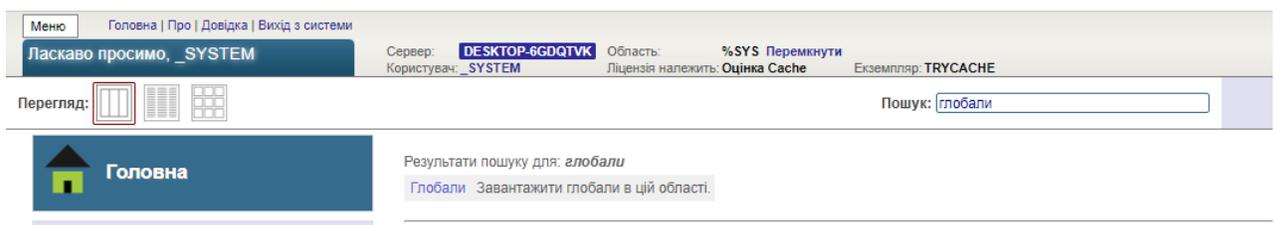


Рисунок 5.5 – Пошук глобалів

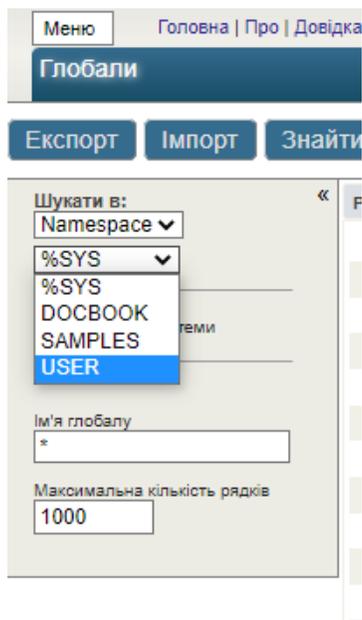


Рисунок 5.6 - Вибір робочої області USER

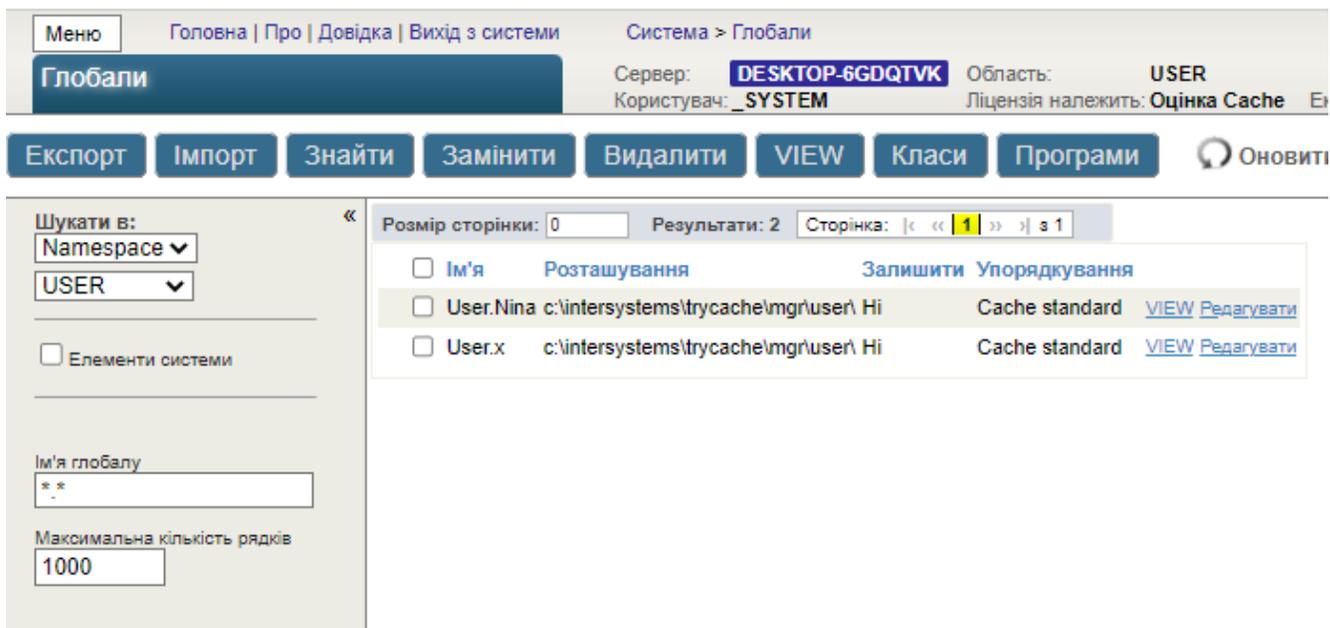


Рисунок 5.7 - Перегляд глобальної змінної.

3. Експериментуйте зі строковими функціями, вбудованими в ядро Cache.

3.1. **Функція \$ Length ()** - повертає довжину рядка або кількість підстрок рядків, якщо вказати другий параметр.

```
USER> w $ L (^ User.x)14
```

3.2. **Функція \$ Extract ()** - повертає підрядок з рядка, початок і кінець якої визначається другим і третім аргументом.

```
USER> w $ E (^ User.x, 7,14)Олексій
```

3.3. **Функція \$ Find** - шукає підрядок у рядку і повертає позицію символу наступного за підрядком, третій аргумент задає початкову позицію пошуку.

```
USER> w $ F (^ User.x, "Олексій")15
```

3.4. **Функція \$ Justify** - вирівнювання числа, яке може бути задано у вигляді рядка заданої довжини із зазначенням потрібної кількості цифр після точки.

```
USER> w $ J ("125", 1,3)125.000
```

3.5. Оператор конкатенації (об'єднання). Має вигляд: "_".

```
USER> s a = "Олексій"
```

```
USER> s b = "Іванович"
```

```
USER> w a_b Олексій Іванович
```

4. Попрацюйте з масивами. **Масив** - це безліч однотипних елементів. Масив має розмірність - кількість індексів і якість значення.

```
USER> s a (1) = "Парацетамол"
```

```
USER> w a (1)
```

```
Парацетамол
```

```
USER> s b (23, "холерик") = "Іванов А."
```

```
USER> w b (23, "холерик")
```

```
Іванов А.
```

Питання для самопідготовки

1. Дайте визначення змінної в Cache Object Script.
2. Для чого використовується програма Terminal?
3. Що таке Cache-куб? і для чого він потрібен?
4. Чим глобальна змінна відрізняється від локальної змінної?
5. Як зберегти масив на диск? Що таке масив?
6. Опишіть функції для роботи з текстовими змінними.
7. Опишіть синтаксис оператора варіанта.
8. Чим лінійний алгоритм відрізняється від розгалуженого алгоритму?
9. Розповісти про скорочений синтаксис Cache.
10. Форматне введення і виведення інформації

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ №6

Тема: Алгоритмізація циклічних процесів. Формальна логіка. Масиви та вбудовані функції навігації в масивах. Інструментальне середовище Studio Intersystem.

Мета роботи: освоїти основні операції над глобалами.

Теоретичні відомості

1. Створення програм за допомогою Cache Studio

Cache Studio - це один з додатків, що викликаються з меню Cache-куба. Для розробки програм пропонується графічний редактор, що дозволяє комфортно створювати класи, CSP - сторінки, програми на макромові, що включають файли, програми на проміжному коді, а також компілювати їх в виконуваний об'єктний код. Cache Studio надає розробникові такий допоміжний засіб, як автоматичний синтаксичний контроль. Різні елементи мови підсвічуються різними кольорами. І в разі порушення синтаксичних правил виводиться повідомлення про помилки. Cache Studio має власний відладник, який може бути використаний для пошуку помилок в додатках.

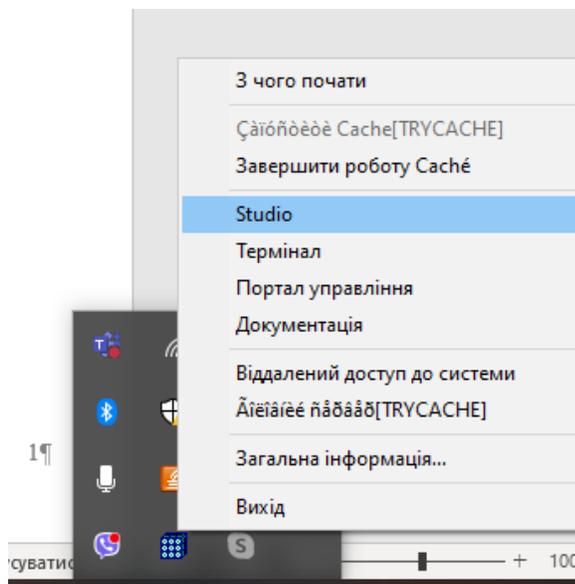


Рисунок 6.1 –Запуск програми Cache Studio з Cache-куба з панелі задач

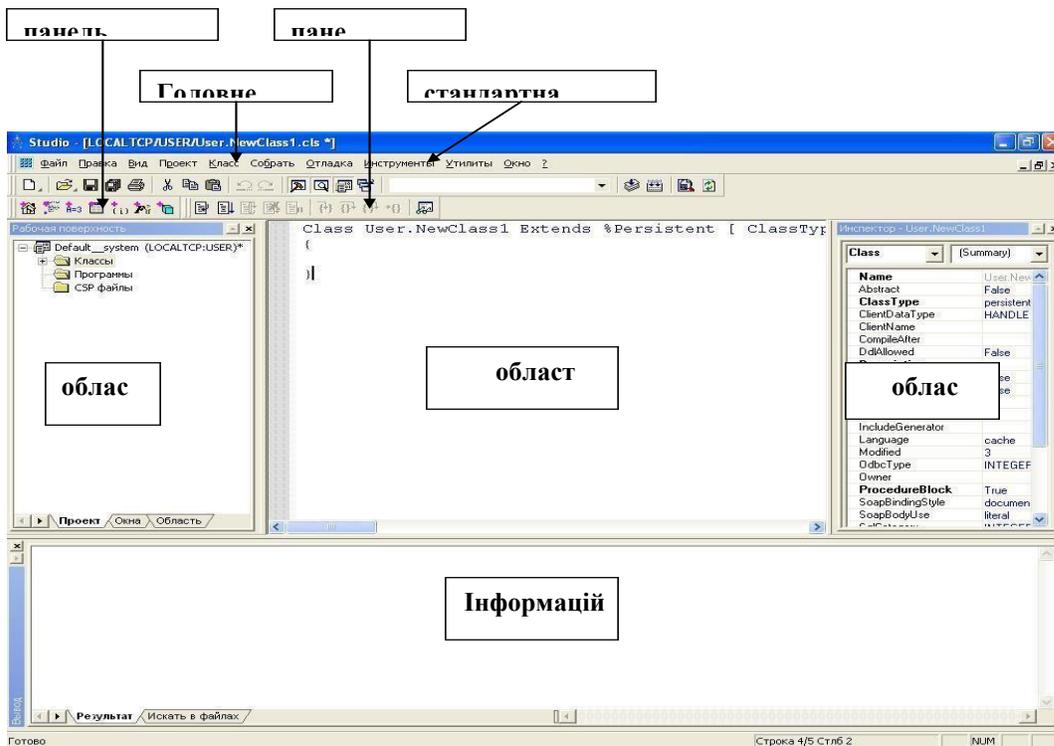


Рисунок 6.2 – Вид головного вікна програми Cache Studio

2. Основні теоретичні відомості про циклічні алгоритми

Розглянемо командні конструкції *For*, *While*, *Do / While*, які забезпечують багаторазове виконання заданого сегменту коду. Це називається **циклом**.

Загальна форма команди **For**:

$$\text{For } \langle \text{for parameter} \rangle \{ \text{code} \}$$

Тут блок {code} виконується стільки разів, скільки задано в параметрі команди <for parameter>. Команди, які прямують у тому ж рядку програми після закриття дужки, вже не належать до команди «For» і виконуються тільки один раз.

Цікавою особливістю команди «For» в Cache є різноманітність варіантів параметра команди. Тут можливі:

- Список різних виразів

For level = <value> [, <value> ...] {code}

- Завдання числового діапазону

For level = <num value>: <num value>: <num value> {code}

Тут перший вираз розглядається як початкове значення, друге - як величина інкремента і третє - як кінцеве значення змінної циклу.

- Числовий діапазон, як показано вище, але без кінцевого значення.

For level = <num value>: <num value> {code}

В цьому випадку в тілі циклу нарощується деяка змінна, а для виходу з циклу служить функція Quit.

- Безаргументна форма. У цій формі немає ні змінної циклу, ні початкового, ні кінцевого значень.

For {code} // як мінімум два пробіли після «For»!

Основні правила циклічних виразів:

- Початкове значення, інкремент і кінцеве значення (якщо воно задане) обчислюють з точністю до першого проходу тіла циклу і зберігаються всередині циклу. Зміна цих значень всередині циклу не впливає на число ітерацій.

- Після закінчення циклу змінна циклу має те значення, яке вона мала при виконанні останньої ітерації.

- У конфліктних випадках (наприклад, якщо кінцеве значення при позитивному інкременті менше початкового) тіло циклу не виконується жодного разу.

Необхідно відзначити, що різні форми конструкції «For» можуть вільно комбінуватися.

Загальна форма команд While і Do / While

Дві керуючі конструкції:

While <value> [, <value> ...] {code}

iDo {code} While <value> [, <value> ...]

також є циклами. Вони відрізняються методом перевірки умови закінчення циклу. У першій конструкції умова перевіряється до виконання блоку програми, а в другій - після. Крім того, для того, щоб блок програми виконався, всі вирази в списку повинні бути логічно істинними.

Важлива відмінність між цими конструкціями полягає в тому, що в разі операторів **Do / While** блок коду виконується хоча б один раз, чого не можна сказати про оператор **While**.

3. Основні теоретичні відомості про глобальні змінні

Cache ObjectScript використовує повністю інтегровані в мову збережені структури даних, так звані глобальні змінні. Ці змінні так само як і тимчасові

«локальні», можуть існувати у вигляді простих або індексованих структур. Глобальні змінні, або коротко глобали, будучи збереженими даними, створюють не тільки основу для тривалого зберігання об'єктів, а й основу для можливого в Cache ObjectScript надзвичайно ефективного прямого доступу. Багатовимірність даних в Cache реалізується через індекси, тому як синонім тут говорять про індексовані змінні.

Синтаксис, що лежить в основі індексованих змінних, досить простий. Індокси розміщують у дужки і відокремлюють один від одного комами, наприклад:

x (1990, "Август", 23) = "День народження"

Змінна *x* має поміщені в дужки індекси, в даному прикладі їх три. Те, що тут бачиться таким простим, набуває в Cache абсолютно особливе значення завдяки таким зауваженням:

- Немає необхідності в попередньому оголошенні розмірності масиву. Створюються лише задані командою Set записи.
- Це стає можливим завдяки концепції виряджені масивів, коли лише існуючі записи займають місце в пам'яті або на диску.

Масив виряджений; це означає, що з огляду на наявність вузла масиву у (2,7) не можна автоматично припускати наявність будь-якого іншого вузла, наприклад у (2,4).

За допомогою одного єдиного знаку циркумфлекс, або знаку Глобал \wedge - локальна змінна перетворюється в збережену (глобальну) змінну. При цьому база даних на основі глобальних сприймається не як окрема сутність, а як логічне розширення концепції локальних змінних. наприклад:

$$\wedge OTD.Tovar (123000,50, "Зелений ківі") = 4$$

Порядок виконання роботи

1. Вивчити теоретичний матеріал, викладений вище, а також в главах 3.2 та 3.3 методичних вказівок в форматі PDF.
2. З меню Cache-куба відкрити Cache Studio, ознайомитися з його інтерфейсом і створити нову програму. Див. рис.6.3.
3. На прикладах, викладених нижче, вивчити основні прийоми роботи з глобалами з використанням циклічних структур. Занести наступні дані про пацієнтів в глобал \wedge **Medizine**:

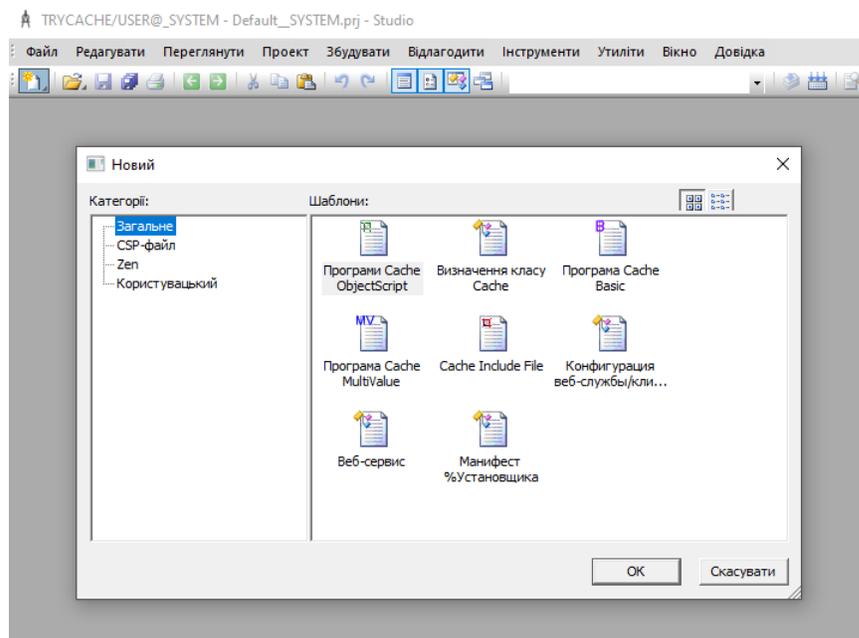


Рисунок 6.3 - Створення нової програми на мові Cache Object Script.

Таблиця 6.1 - Данні про пацієнтів

прізвище пацієнта	вік	стать	кров'яний тиск	температура тіла, С
Іванов	65	чоловіча	125/60	36,6
Сидорова	50	жіноча	Немає даних	Немає даних
Тарасов	27	чоловіча	110/80	38,6
Петров	45	чоловіча	Немає даних	36,6
Іваненко	34	жіноча	115/85	Немає даних
Горбунков	60	чоловіча	100/80	39,3
Іващенко	71	чоловіча	140/60	36,9
Турчинов	65	чоловіча	Немає даних	Немає даних

Програма має вигляд, наведений на рисунку нижче:

```

TRYCACHE/USER@_SYSTEM - Default_SYSTEM.prj - Studio - [Без назви4.MAC *]
Файл Редагувати Переглянути Проект Збудувати Відлагодити Інструменти Утиліти Вікно Довідка
PROG1.mac PROG2.mac PROG3.mac Без назви4.MAC *
Programm1
s ^Medizine(65, "чоловічий", "125/60", 36.6)="Іванов"
s ^Medizine(50, "жіночий")="Сидорова"
s ^Medizine(27, "чоловічий", "110/80", 38.6)="Тарасов"
s ^Medizine(45, "чоловічий", 36.6)="Петров"
s ^Medizine(34, "жіночий", "115/85")="Іваненко"
s ^Medizine(60, "чоловічий", "100/80", 39.3)="Горбунков"
s ^Medizine(71, "чоловічий", "140/60", 36.9)="Іващенко"
s ^Medizine(65)="Турчинов"
    
```

Рисунок 6.4 - Робота над власною програмою в Cache Studio

- Після набору коду в центральному вікні (рис.6.4) програму слід

відкомпілювати, шляхом натискання клавіш Ctrl + F7. Перекомпілюйте програму кожен раз, коли щось в ній міняєте або виправляєте помилки!

Після першої компіляції програма попросить збереження, див. рис.6.5.

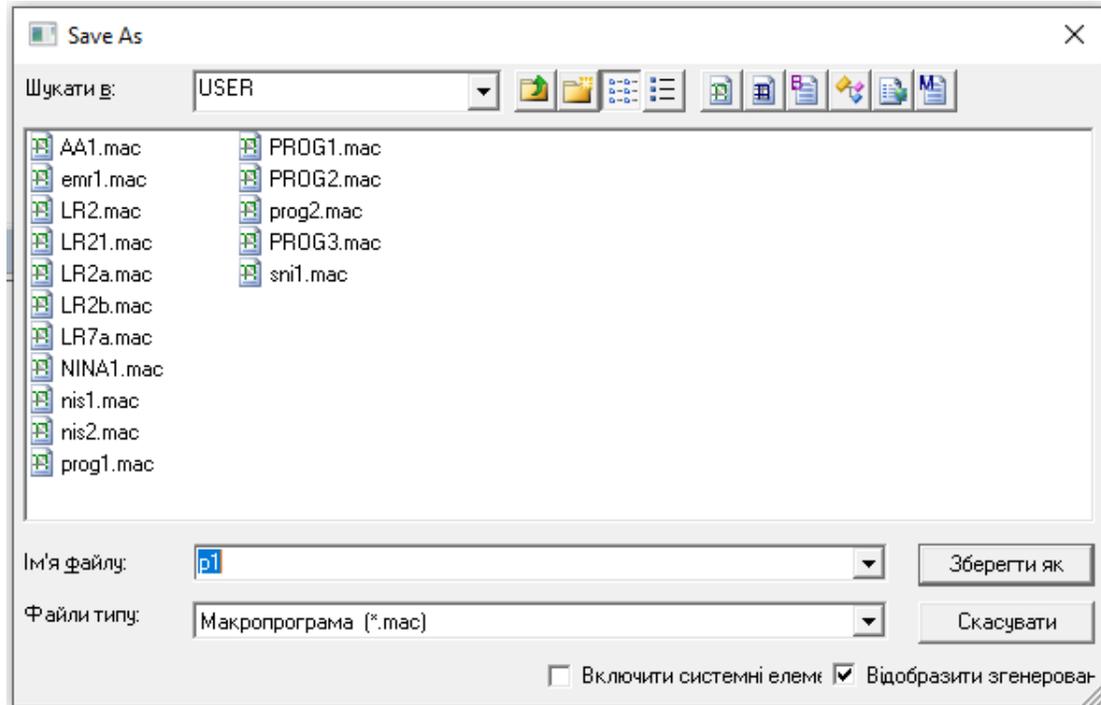


Рисунок 6.5 - Збереження програми на диску

5. Для виконання програми використовуйте програму «Термінал», яка також викликається з меню Cache-куба. Див. рис.6.6.

Виклик програми здійснюється за допомогою оператора «do», далі пробіл, далі знак циркумфлекс «^» і потім повне ім'я програми, що тестується (в нашому випадку це ^p1). В кінці кожного рядка натискається «Enter».

```

Cache TRM:13956 (TRYCACHE)
Файл  Редагувати  Довідка

Node: DESKTOP-6GDQTVK, Instance: TRYCACHE

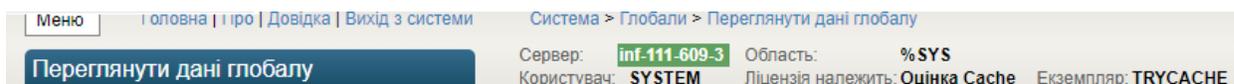
Username: _SYSTEM
Password: *****
USER>do ^pl

```

Рисунок 6.6 - Виконання створеної програми в «Терміналі»

Переконайтеся в тому, що всі дані збережені на диску в глобалі ^Medizine.

Для цього відкрийте портал управління системою:



Переглянути глобал в області %SYS:

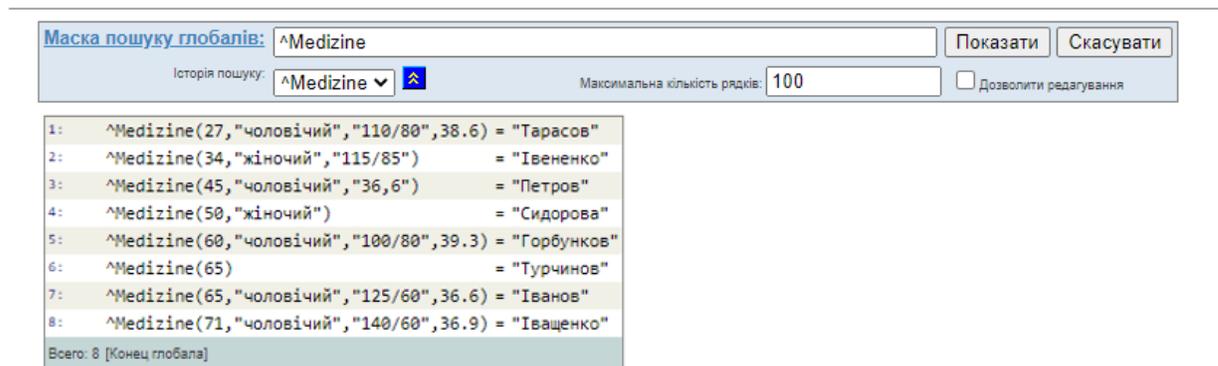


Рисунок 6.7 - Перегляд програми з ім'ям PROG1 в області User

• **Робота з функцією \$ Data.** Функція \$ Data (коротко \$ D) - це функція з ім'ям змінної в якості аргументу. Ця функція може працювати з локальною або з глобальною, скалярною або індексованою змінною. Значення функції \$ Data визначає, чи існує задана у вигляді аргументу змінна і яка в неї структура.

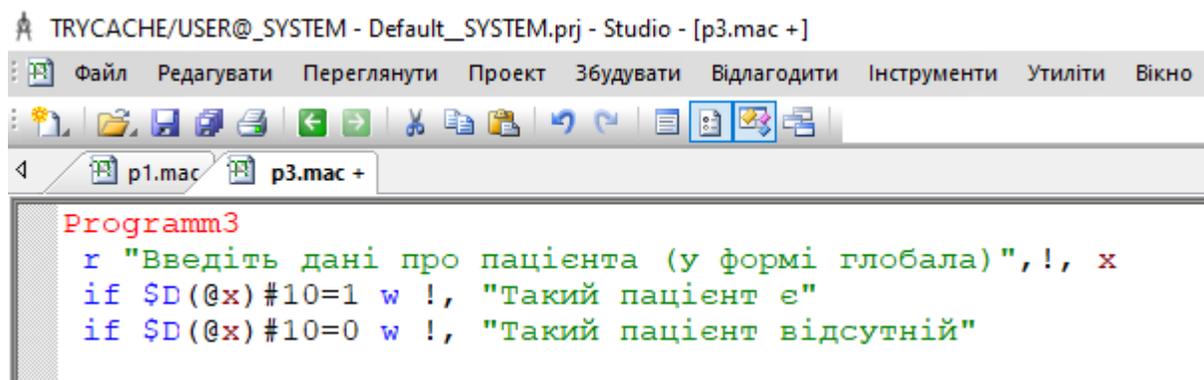
Функція повертає чотири різних результати:

1. Якщо її значення дорівнює 0 (10), то змінна не існує;
2. Якщо її результат 1 (11), то змінна існує.

Приклад програми, яка дозволяє визначити є даний пацієнт на лікуванні чи ні, наведена на малюнку нижче:

Програма має вигляд:

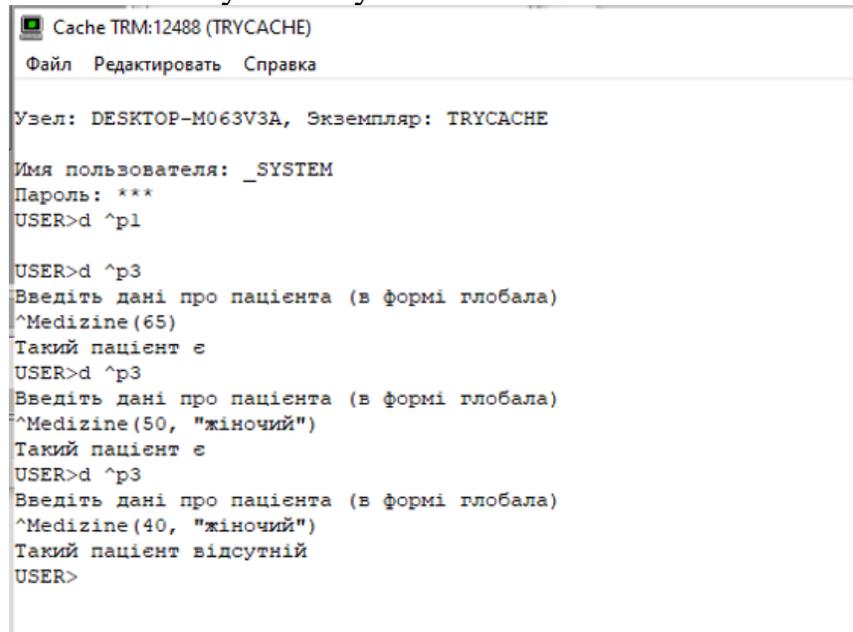
Тут застосований оператор «побічно» - @, який перетворює в даному випадку ім'я змінної в її значення. У загальному випадку він призначений для перетворення рядка символів в програмний код.



```
TRYCACHE/USER@_SYSTEM - Default_SYSTEM.prj - Studio - [p3.mac +]
Файл Редагувати Переглянути Проект Збудувати Відлагодити Інструменти Утиліти Вікно
p1.mac p3.mac +
Programm3
r "Введіть дані про пацієнта (у формі глобала)",!, x
if $D(@x)#10=1 w !, "Такий пацієнт є"
if $D(@x)#10=0 w !, "Такий пацієнт відсутній"
```

Рисунок 6.8 – Текст програми з функцією \$Data

Результат її роботи може бути наступним:



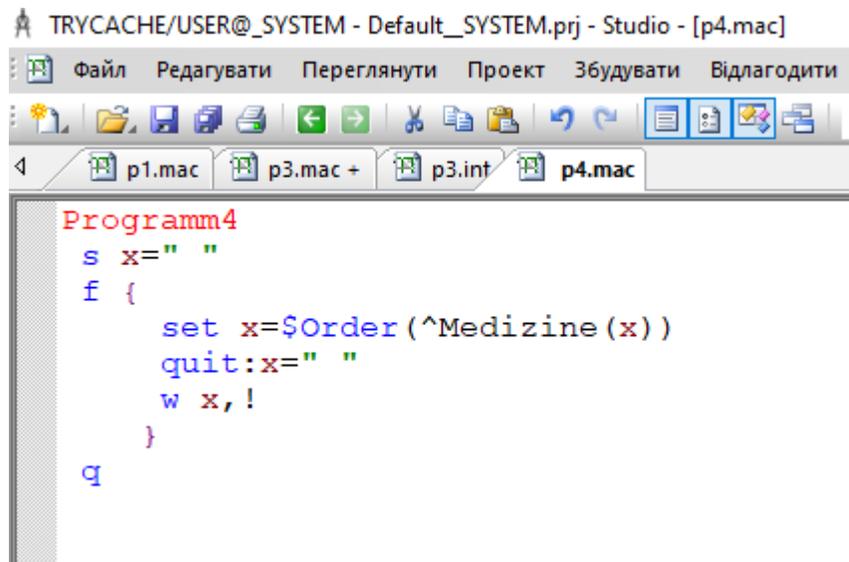
```
Cache TRM:12488 (TRYCACHE)
Файл Редактировать Справка
Узел: DESKTOP-M063V3A, Экземпляр: TRYCACHE
Имя пользователя: _SYSTEM
Пароль: ***
USER>d ^p1
USER>d ^p3
Введіть дані про пацієнта (в формі глобала)
^Medizine(65)
Такий пацієнт є
USER>d ^p3
Введіть дані про пацієнта (в формі глобала)
^Medizine(50, "жіночий")
Такий пацієнт є
USER>d ^p3
Введіть дані про пацієнта (в формі глобала)
^Medizine(40, "жіночий")
Такий пацієнт відсутній
USER>
```

Рисунок 6. 8 - Результат роботи програми з функцією \$Data

• **Робота з функцією \$ Order.** Вона є основною функцією для отримання чергового в послідовності сортування індексу локальної або глобальної індексованої змінної. Застосовуючи цикл, відповідно до послідовності сортування можуть бути отримані всі існуючі індекси даного рівня. Для того, щоб задати найперший індекс в межах заданого рівня, використовуємо порожній рядок. Приклад застосування виконаємо в Терміналі:

USER> w \$ O (^ Medizine (65)) 71

Отримаємо відповідь на наступне питання: Пацієнти яких віків присутні на лікуванні? Відповіддю буде програма:



```
TRYCACHE/USER@_SYSTEM - Default_SYSTEM.prj - Studio - [p4.mac]
Файл  Редагувати  Переглянути  Проект  Збудувати  Відлагодити
p1.mac  p3.mac +  p3.int  p4.mac
Programm4
s x=" "
f {
    set x=$Order(^Medizine(x))
    quit:x=" "
    w x,!
}
q
```

Результат налагодження програми в «Терміналі» має вигляд, представлений на рис.6.9.



```

USER>d ^p4
27
34
45
50
60
65
71
USER>

```

Рисунок 6. 9 - Отримання списку віку пацієнтів

Це один з класичних циклів Cache ObjectScript, використання якого або в цій формі, або в різних модифікаціях при прямому доступі до глобалів залишається актуальним.

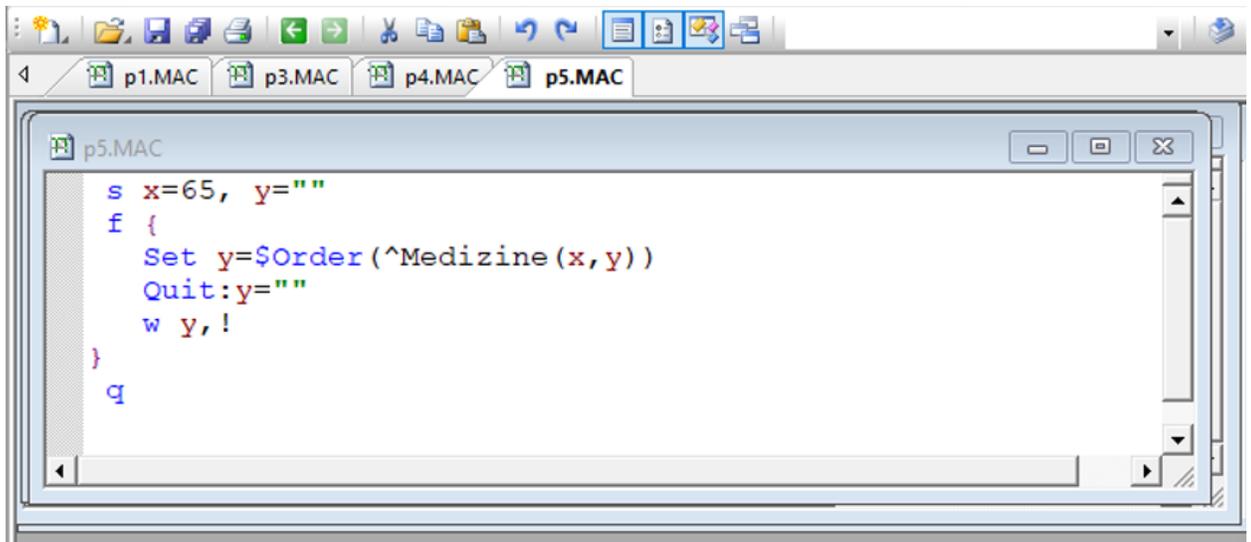
Яким чином отримують до вже заданому індексу перший індекс на другому рівні? Це здійснюється точно таким же способом, як і раніше, шляхом використання порожнього рядка на другому рівні індексації:

```

USER> w $ O (^Medizine (27, ""))
        чоловічий

```

Нехай потрібно відповісти на питання, чи є серед пацієнтів віком 65 років жінки? Відповіддю буде програма :



Результат виконання програми дав відповідь «чоловічий», інтерпретація

якого може означати, що жінок даного віку на лікування не надходило.

```
USER> d ^ p5
```

чоловічий

Функція \$ Query. Виходячи з особливого характеру функції \$ Order, вона, перш за все, призначена для отримання всіх по порядку існуючих індексів в межах даного рівня. Однак іноді більш природним є інший спосіб перегляду. Якщо перегляд послідовності індексів спочатку йде в глибину і показує деталі на більш високих рівнях до тих пір, поки ці рівні не закінчаться, а потім проглядається наступний запис на вищому рівні індексації, то це так званий пошук «depth-first» (спочатку вглиб) , здійснюваний за допомогою функції \$ Query. За допомогою цієї функції ми отримуємо ім'я вузла на наступному рівні, але на відміну від функції \$ O, виходить повне посилання у вигляді рядка символів:

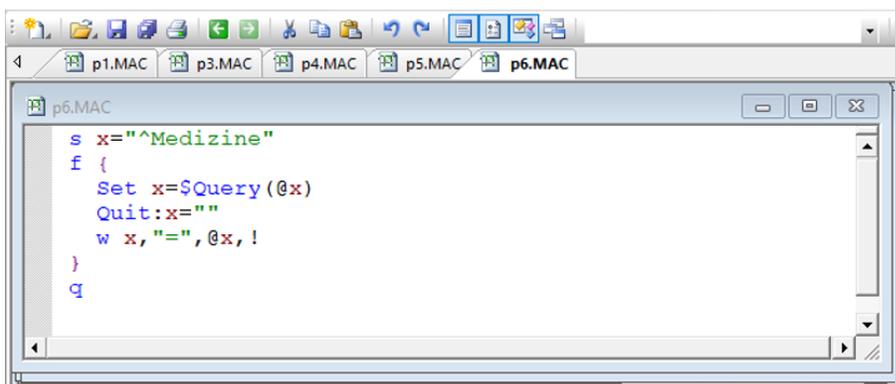
```
USER> w $ Q (^ Medizine (27))
```

```
^ Medizine (27, "чоловічий", "110/80", 38.6)
```

```
USER> w $ Q (^ Medizine (27, "чоловічий", "110/80", 38.6))
```

```
^ Medizine (34, "жіночий", "115/85")
```

Щоб отримати в нашій моделі даних всі певні записи, необхідно застосувати цикл у програмі :



```
s x="^Medizine"
f {
  Set x=$Query(@x)
  Quit:x=""
  w x,"=",@x,!
}
q
```

Результатом правильної роботи програми буде наступна відповідь у терміналі:



```
Cache TRM:3268 (TRYCACHE)
Файл Редагувати Довідка
```

```

USER>d ^p6
^Medizine (27,"чоловічий","110/80",38.6)=Тарасов
^Medizine (34,"жіночий","115/85")=Івененко
^Medizine (45,"чоловічий","36,6")=Петров
^Medizine (50,"жіночий")=Сидорова
^Medizine (60,"чоловічий","100/80",39.3)=Горбунков
^Medizine (65)=Турчинов
^Medizine (65,"чоловічий","125/60",36.6)=Іванов
^Medizine (71,"чоловічий","140/60",36.9)=Івашенко

USER>

```

аліз індексованих змінних за допомогою функцій \$ QLength і \$ QSubscript.

У деяких варіантах постановки задачі незалежно від того, чи маємо ми справу з глобальною або локальною індексованою змінною, виникає необхідність дослідження її структури. На передньому плані тоді стоять питання:

1. Скільки рівнів (індексів) має індексована змінна?
2. Яке значення і-го індексу?

Демонстрація використання даної функції в терміналі має наступний вигляд:

```

USER> w $ QL ( "^ Medizine (65)") 1
USER> w $ QL ( "^ Medizine (65," "чоловічий" ")") 2

```

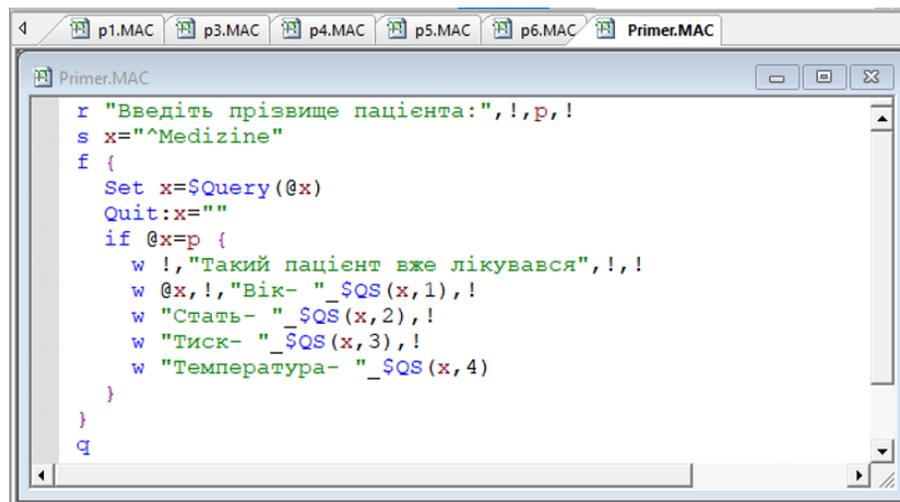
В останньому прикладі використані подвійні лапки, щоб інтерпретатор мови чи не переплутав індекс і аргумент функції \$ QL. Аргумент завжди береться в подвійні лапки.

Тепер звернемося до функції \$ QSubscript (скорочено \$ QS), яка має два аргументи. Першим аргументом є досліджуване значення імені індексованої змінної, другий задає, який саме індекс слід витягти. Якщо в загальному випадку значення імені має форму Name (s1, s2, s3sn), то результат функції \$ QS (Name (s1, s2, s3sn), m) дорівнює значенню m-го індексу в разі, якщо m не перевищує n:

USER> w \$ QS ("^ Medizine (27," "чоловічий" ", "110/80" ")", 3) 110/80

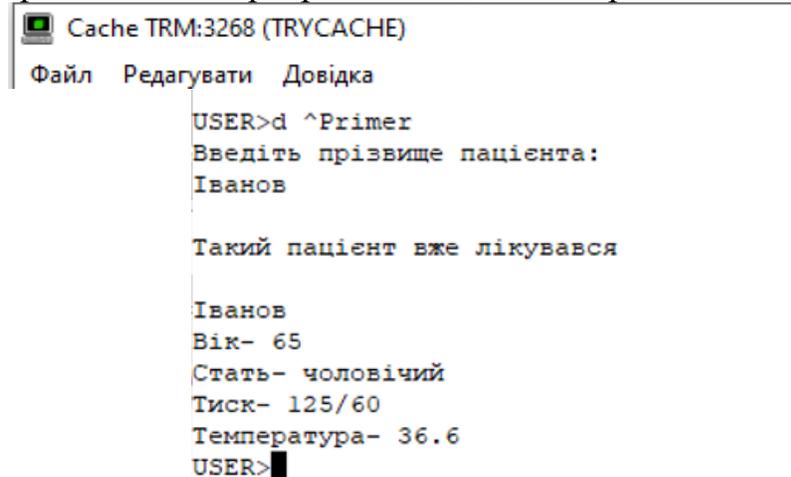
- Комбіноване використання розглянутих функцій:

Нехай потрібно дати відповідь на питання, чи надходив на лікування пацієнт з прізвищем «Іванов», і якщо надходив, то вивести інформацію про нього. Відповіддю є програма, яка запитує прізвище пацієнта і видає параметри, що характеризують його стан. Прізвище поміщається в змінну p, в циклі перегортаються усі наявні записи про пацієнтів і далі якщо знаходимо, що черговий запис дорівнює p (прізвище людини), то виводимо на екран всі дані про нього.



```
r "Введіть прізвище пацієнта:",!,p,!
s x="^Medizine"
f {
  Set x=$Query(@x)
  Quit:x=""
  if @x=p {
    w !,"Такий пацієнт вже лікувався",!,!
    w @x,!, "Вік- "_$QS(x,1),!
    w "Стать- "_$QS(x,2),!
    w "Тиск- "_$QS(x,3),!
    w "Температура- "_$QS(x,4)
  }
}
```

Результат роботи такої програми наведений на рис.6. 11.



```
Cache TRM:3268 (TRYCACHE)
Файл Редагувати Довідка
USER>d ^Primer
Введіть прізвище пацієнта:
Іванов

Такий пацієнт вже лікувався

Іванов
Вік- 65
Стать- чоловічий
Тиск- 125/60
Температура- 36.6
USER>
```

Рисунок 6. 11 - Результат роботи програми з функцією \$Query

Питання для самопідготовки

1. Чим глобальна змінна відрізняється від локальної?
2. Чим індексована змінна відрізняється від скалярної?
3. Що таке циклічний алгоритм? Які оператори реалізують його в мові Cache ObjectScript?
4. Для чого використовується програма Cache Studio?
5. За допомогою яких функцій в Cache ObjectScript здійснюється навігація в багатовимірних масивах?
6. Яка функція визначає існування даних на n-му рівні?
7. Результатом роботи, якої функції є ім'я n-го індексу масиву?
8. Яку роль виконує функція \$ Query? Що є її аргументом?

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ № 7

Тема: основи об'єктно-орієнтованого підходу к аналізу задач фармацевтичного профілю. UML – моделювання об'єктів предметної галузі

Мета роботи: освоїти основні принципи об'єктно-орієнтованого моделювання об'єктів професійної предметної області.

Теоретичні відомості

Одне з основних завдань, яке виникає при моделюванні інформаційних систем, полягає в тому, щоб відображати реальний світ, не спотворюючи внутрішніх закономірностей предметної області. Для цього була створена об'єктно-орієнтована (ОО), або об'єктна, технологія. Основним поняттям об'єктної технології є сам об'єкт, який визначається як інкапсульована сутність, що володіє властивостями і методами.

Інкапсуляція означає, що внутрішня реалізація об'єкта прихована від користувача. Але при цьому об'єкти мають відкритий інтерфейс, описуваний у вигляді набору властивостей і методів. У загальному випадку термін «*інтерфейс*» означає сукупність засобів, методів і правил взаємодії між елементами системи. Методи визначають поведінку об'єкта, а властивості – його «знання», іншими словами, дані.

В об'єктних системах об'єкти одного типу прийнято об'єднувати в класи об'єктів. Окремі суті цих класів - індивідуальні об'єкти - називають екземплярами класу. Наприклад, з точки зору об'єктного технології, кожен людський індивідум - це екземпляр класу «Людина». Клас об'єктів визначає властивості і методи, тобто той інтерфейс, який є єдиним для всіх його примірників. Це визначення встановлює свого роду рамки, які кожен екземпляр заповнює своїми індивідуальними значеннями. Стало бути, екземпляри мають тип відповідного класу. Крім цього, екземпляри мають об'єктну індивідуальність незалежно від

значень їх властивостей. Так, два примірники одного класу навіть при ідентичних значеннях всіх властивостей чітко помітні. Наприклад, однойцеві близнюки є двома різними індивідуумами або екземплярами, навіть якщо вони виявляють повністю однакові властивості. Примірники одного класу можуть відрізнятися лише значеннями своїх властивостей, але не своїми методами.

Методи встановлюються для всіх екземплярів при визначенні класу. Звідси випливає, що всі члени одного класу виявляють не тільки ідентичний інтерфейс, але і ідентичне «поведінку». При цьому розрізняють методи примірників і методи класу, розуміючи під цим, що відповідний метод або відноситься до певних екземплярів класу, або діє незалежно від них та, що всі члени одного класу виявляють не тільки ідентичний інтерфейс, але й ідентичну «поведінку». При цьому розрізняють методи примірників і методи класу, розуміючи під цим, що відповідний метод або відноситься до певних екземплярів класу, або діє незалежно від них.

Як приклад уявімо собі клас «Трикутник», що володіє такими методами, як «Намалювати», «Видалити», «Перемістити», а також властивостями «Позиція», «Розміри», «Колір». Окремі екземпляри мають значеннями властивостей: певною позицією, розмірами і кольором (рис. 7.2.).

ОО технологія має наступні переваги для проектування інформаційних систем:

- На відміну від процедурного підходу, при якому здійснюється структурна розробка інформаційної системи, об'єктна технологія сприяє осмисленому структуруванню інформації. Завдяки цьому автоматично здійснюється розбиття великої системи на доступні для огляду одиниці, які можуть розроблятися і супроводжуватися незалежно один від одного.

- Об'єкти мають добре визначені інтерфейси, за допомогою яких їх вдається застосовувати в інших об'єктах, можливо, не знаючи їх внутрішньої структури.

- Завдяки тому, що об'єкти відображають суті реального світу, їх з легкістю вдається повторно використовувати в різноманітних додатках, оперують тими ж самими сутностями.

- Відповідно до ОО - підходом, моделюється не процес, а інформаційні структури, які змінюються значно повільніше, тому існує тенденція до значно більш тривалого життєвого циклу об'єктно-орієнтованого програмного забезпечення.

Класи можна організувати у вигляді ієрархічноґ структури, яка за зовнішнім виглядом нагадує схему класифікації в понятійної логіці. Ієрархія понять будується наступним чином. В якості найбільш загального поняття або категорії береться поняття, що має найбільший обсяг і, відповідно, найменший зміст. Це найвищий рівень абстракції для даної ієрархії. Потім дане загальне поняття конкретизується, тобто зменшується його обсяг і збільшується зміст. З'являється менш загальне поняття, яке на схемі ієрархії буде розташовано на рівень нижче початкового. Цей процес конкретизаціїґ понять може бути продовжений до тих пір, поки на самому нижньому рівні не буде отримане поняття, подальша конкретизація якого в даному контексті або неможлива, або недоцільна.

Принцип, відповідно до якого знання про найбільш загальної категорії дозволяється застосовувати для більш приватної категорії, називається *спадкуванням*. Спадкування тісно пов'язане з ієрархією класів, яка визначає, які класи слід вважати найбільш абстрактними і загальними по відношенню до інших класів. При цьому якщо загальний або батьківський клас (предок) володіє фіксованим набором властивостей і поведінкою, то похідний від нього клас (нащадок) повинен містити цей же набір властивостей і подібну поведінку, а також додаткові, які будуть характеризувати унікальність отриманого класу. У цьому випадку говорять, що похідний клас успадковує властивості і поведінку батьківського класу.

Для ілюстрації принципу успадкування можна навести такий приклад. Існує

загальний клас "Комп'ютер". Він визначається як абстракція властивостей і поведінки всіх, реально існуючих електронних обчислювальних машин. При цьому загальними властивостями класу "Комп'ютер" можуть бути такі, як наявність процесора, оперативної пам'яті, пристроїв введення та виведення інформації. Якщо в якості похідного розглянути клас "Персональний комп'ютер", то всі виділені вище властивості будуть містити і цей клас. Можна сказати, що клас "Персональний комп'ютер" успадковує властивості батьківського класу "Комп'ютер".



Рисунок 7.1 -Ієрархія вкладеності класів для прикладу загального класу "Комп'ютер"

У свою чергу клас "Персональний комп'ютер" може бути класом-предком для інших класів, зокрема "Робоча станція", "Сервер" і "Ноутбук". З цієї точки зору всі зазначені класи успадковують властивості батьківського класу "Персональний комп'ютер", а можливо і скасовують деякі з них. У понятійній логіці для зображення понять використовуються кола або прямокутники. За

допомогою цієї графічної нотації ієрархію класів для розглянутого прикладу можна представити у вигляді вкладених прямокутників або кіл, кожен з яких відповідає окремому класу (рис. 7.1.).

Поліморфізм також один з основних принципів ООП. Під поліморфізмом (грец. Poly - багато, morfos - форма) розуміється властивість об'єктів приймати різні зовнішні форми в залежності від обставин. Стосовно до ООП поліморфізм означає, що дії, які виконуються однойменними методами, можуть відрізнятися в залежності від того, до якого з класів належить той чи інший метод. Наприклад, три об'єкти відповідних класів: двигун автомобіля, електричне світло в кімнаті і персональний комп'ютер. Для кожного з них можна визначити операцію «вимкнути» (). Однак результат виконання цієї операції буде відрізнятися для кожного з розглянутих об'єктів. Так для двигуна автомобіля виконання операції «вимкнути» () означає припинення подачі палива і його зупинку. Рятувальна операція «вимкнути» () для електричного світла в кімнаті означає простий клацання вимикача, після чого кімната занурюється в темряву. В останньому випадку для персонального комп'ютера виконання операції «вимкнути» () може бути причиною втрати даних, якщо це проводиться нерегламентованим чином.

Елементи класів Cache Intersystem

При визначенні класу встановлюється його тип і поведінка. Фактично це визначення складається з властивостей і методів класу (від класів типів даних не можуть утворюватися екземпляри, тому вони мають лише методи, але не властивості). До повного визначення класу відносяться і інші функціональні елементи, такі, як запити і індекси, а також параметри і ключові слова. Повний список елементів визначення класу охоплює:

- Однозначне ім'я класу;
- *Ключові слова* - кілька ключових слів, що дозволяють

модифікувати визначення класу;

- *Властивості* (Відомі такі як стан) - елементи даних для зберігання в примірниках класу. Властивості можуть бути константами, вбудованими об'єктами і посиланнями на збережені об'єкти. Класи типів даних не містять властивостей;

- *методи* - код, який реалізує ті чи інші функціональні можливості;
- *параметри класу* - значення, які здійснюють настройку функціональних можливостей класу під час його компіляції (зазвичай з використанням генераторів методів);

- *запити* - операції з безліччю екземплярів класу;

- *індекси* - структури в довготривалій пам'яті, оптимізують доступ до об'єктів.

Кожен елемент класу наділяється унікальним ім'ям. Зауважимо, що в цих іменах лише для полегшення сприйняття допустимі обидва регістра букв. Усередині Cache вони перетворюються в великі букви, що слід враховувати при призначенні імен. Імена, що починаються зі знака «%» (відсоток), зарезервовані для елементів системних класів.



Рисунок 7.2 - Елементи класів

Класи в Cache зазвичай визначаються за допомогою Cache Studio,

інтегрованим середовищем розробки класів. Повне визначення класу формується на мові CDL (Class Definition Language). наприклад:

```
Class User.Person Extends (% Persistent)
```

Тут ключове слово Class позначає початок опису класу, User.Person - ім'я пакета і класу, розділених крапкою, % Persistent - класи об'єктів.

Властивості представляють стан об'єктів. Якщо це здійснюється за допомогою зберігання деяких значень (як це має місце в Cache), але в цьому випадку говорять про **атрибути** об'єктів. Властивість має однозначне ім'я, тип даних, опціональний список ключових слів і опціональний список параметрів, визначених для відповідного типу даних. До основних типів даних Cache відносяться:

- % Binary (двійкове значення),
- % Boolean (логічне значення),
- % Currency (валюта),
- % Date (дата),
- % Float (число з плаваючою точкою),
- % Integer (ціле число),
- % List (список),
- % Name (ім'я в форматі «ім'я», «Прізвище»),
- % String (рядок символів),
- % Time (час),
- % TimeStamp (дата і час).

Властивості можуть представляти собою константи (прості значення даних деякого типу), а також посилання на об'єкти, вбудовані об'єкти, потоки даних (BLOBs), різні види колекцій, багатовимірні змінні і двонаправлені зв'язку між збереженими об'єктами. У простій формі властивість може бути типу константи (% String,% Integer та ін.) Наприклад, цілочисельну властивість Number з

найбільшим значенням 100 можна визначити наступним чином, використовуючи мову визначення класів CDL:

Property Count As% Integer (MAXVAL = 100);

Тут ключове слово *Property* позначає початок опису властивості класу, Count -

ім'я властивості.

В Cache кожен клас можна розглядати і як тип даних. Якщо при визначенні якості замість класу типу даних вказати зберігається клас, тим самим буде утворено посилання на об'єкт, що зберігається. У цьому випадку значеннями властивості будуть посилання на екземпляри відповідного класу. Наприклад, так можна визначити властивість *Manufacturer*, що є посиланням на збережений клас об'єктів *User.Manufacturer*:

Property Manufacturer As User.Manufacturer;

Вбудовані об'єкти працюють аналогічно посиланням на об'єкти. Істотна відмінність полягає в тому, що в якості типу даних тепер задається вбудований клас об'єктів. Отже, значення цієї властивості - не посилання на якийсь самостійний екземпляр об'єкта, а, навпаки, сам цей об'єкт вбудовується в об'єкт, який на нього посилається. З точки зору синтаксису визначення ідентичне посиланням на клас об'єктів:

Property Address As User.Address;

Потоком даних (Binary Large Object - BLOB) називають велику неструктуровану безліч даних, яку необхідно зберігати в базі даних. Наприклад, потоки даних можуть застосовуватися для зберігання документів, технічних креслень або малюнків.

Властивості, що володіють безліччю значень, можуть бути представлені в Cache у вигляді колекцій. Надалі ми будемо розрізняти колекції -масиви (Array Collections), сортовані відповідно до ключа, і колекції - впорядковані списки (колекції -списки, або List Collections). Колекції можуть містити константи,

вбудовані об'єкти і посилання на об'єкти. У колекції -масиві кожен елемент упорядковується в позицію, яка визначається ключем. Колекція-список являє собою упорядкований список елементів, в якому кожен елемент визначається номером своєї позиції. Наведемо кілька прикладів:

Property DateOfBirth As% Date [Collection = array]; Property Name As% String [Collection = list];

Елементи колекції-списку можуть являти собою вбудовані об'єкти або посилання на збережені об'єкти. Нижче - приклад списку посилань на збережені об'єкти:

Property Person As User.Person [Collection = list];

Відносини являють собою двонаправлені зв'язки між збереженими об'єктами. У багатьох аспектах відношення подібно взаємним посиланням, встановленим між двома збереженими об'єктами. Однак ставлення гарантує цілісність посилань, завдяки якій після видалення одного з об'єктів в іншому з них ніколи не з'явиться посилання на неіснуючий об'єкт. Відносини можуть бути двох типів: один-до-багатьох (незалежні об'єкти) і батько - нащадок (нащадок залежить від батьків). На кожній стороні відносини знаходиться клас збережених об'єктів, який може бути одним і тим самим класом в обох випадках.

Методами називають операції, які може виконувати об'єкт або клас об'єктів. Кожен метод має однозначне ім'я, формальну специфікацію аргументів і значення, що повертається і, звичайно ж, код методу. До складу кожного методу входить список формальних аргументів, який при виклику методу наповнюється актуальними значеннями часу виконання. Кожен можливий формальний аргумент в списку наводиться з його очікуваним типом даних і опціональним значенням за замовчуванням. Якщо тип даних не заданий, за замовчуванням приймається % String. Список формальних аргументів може бути порожнім, в цьому випадку метод не очікує ніяких аргументів. З іншого боку, при виклику не обов'язково ставити всі аргументи, тоді вони залишаються невизначеними або

наділяються значеннями за замовчуванням, якщо такі є. Приклад методу:

*// Підвищення зарплати співробітнику з ідентифікатором MyID на Percent
кількість відсотків.*

```
ClassMethod IncreaseSalary (MyID As% Integer, Percent As% Float) As%  
    Status  
    {  
    // Тіло методу  
    }
```

Тут `ClassMethod` - означає початок опис методу класу,
`IncreaseSalary` - ім'я методу,
`MyID As% Integer, Percent As% Float` – вхідні параметри,
`% Status` - тип результату, що повертається.

При роботі з об'єктами реального світу складність відображуваних типів даних зростає. Наприклад, в комерційному додатку ми маємо справу з клієнтами, яких хочемо відобразити в класі «Клієнт». Він володіє не тільки простими властивостями строкового або цілочисельного типу, але і властивістю «Ім'я».

Припустимо, воно складається з прізвища, імені, по батькові, а також, можливо, з звань і наукових ступенів. Такі складові (або похідні) типи даних в прогресивних системах визначаються як розширені типи даних і можуть використовуватися так само, як і базові типи даних, що поставляються разом з системою. При цьому визначення типу даних є особливим випадком визначення класу. З іншого боку, кожен клас об'єктів одночасно може розглядатися і як певний тип даних. Припустимо, що в розглянутому прикладі комерційного додатку необхідно управляти мережею агентів по збуту, інформація про яких зберігається в класі «Агент». Тоді об'єкт класу «Клієнт» може мати властивість типу «Агент» для відображення факту наявності посилання на агента, який обслуговує даного клієнта. На відміну від агента зі збуту, адреса не повинна ідентифікуватися незалежно від клієнта, тому його природно уявити як

вбудований об'єкт. В цьому випадку його властивості «Вулиця» і «Місто» будуть існувати лише в контексті об'єкта «Клієнт». Однак адреса матиме власне визначення класу, яке, можливо, вдасться використовувати повторно (наприклад, при визначенні класу «Постачальник»).

Вбудований об'єкт, в свою чергу, може містити інший вбудований об'єкт; глибина вбудовування логічно не обмежена. Великі неструктуровані безлічі даних, як правило, зберігаються у вигляді потоків даних. У нашому прикладі цим механізмом можна було б скористатися для зберігання графічного зображення маршруту проїзду до клієнта. Властивість може повторюватися багато разів (незалежно від його типу даних). У цьому випадку говорять про *колекцію*. Так, в об'єкті «Клієнт» може міститися необмежена кількість рахунків у вигляді вбудованих об'єктів; також колекція може містити і прості значення, такі, як рядки символи, розширені типи даних або посилання на об'єкти.

Порядок виконання роботи

1. З меню Cache-куба відкрити Studio і створити новий клас Patient0 (див. рис.7.3).

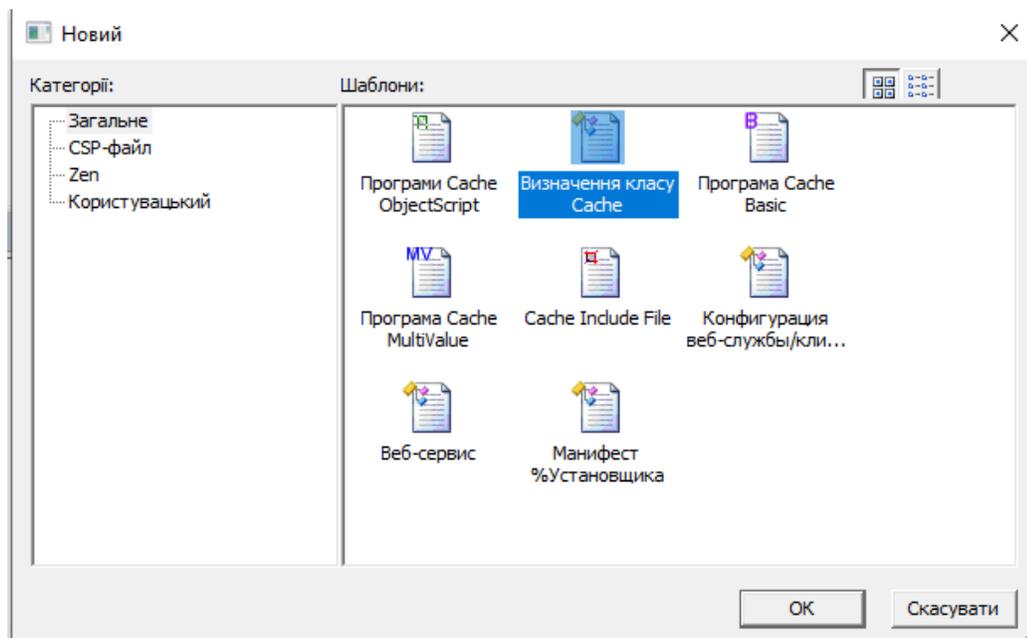


Рисунок 7.3 - Створення нового класу.

2. Дотримуючись майстру створення класу задати характеристики класу, див. рис.7.4- 7.7.

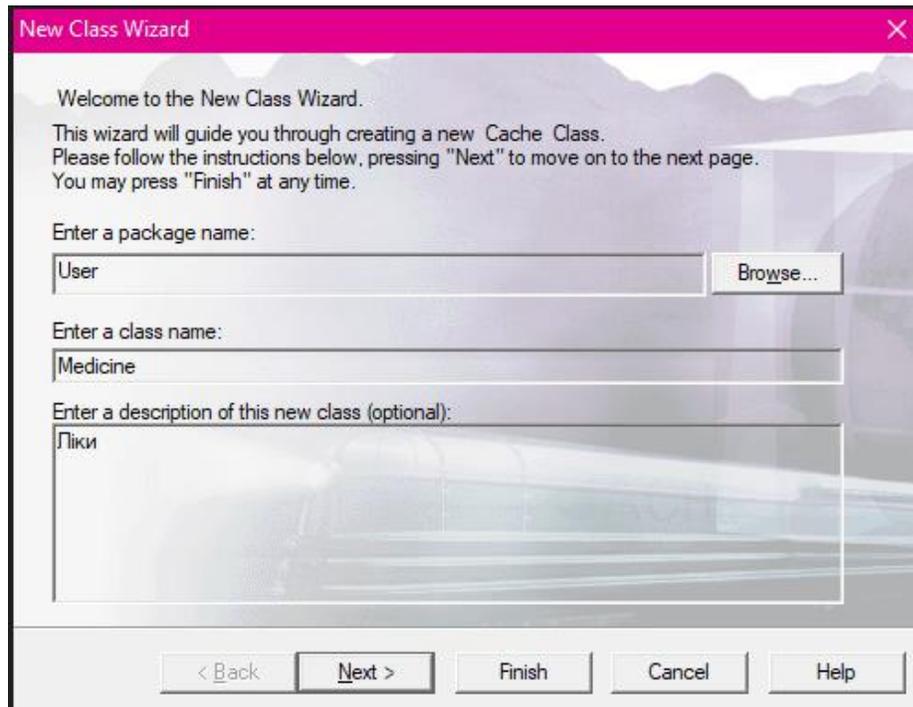


Рисунок 7. 4 - Завдання імені та опис класу.

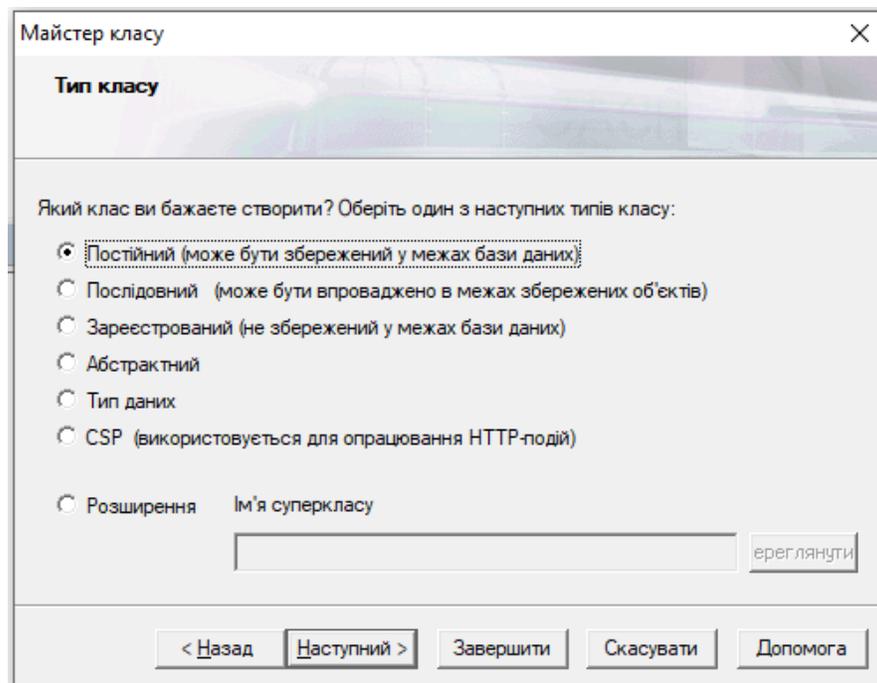


Рисунок 7. 5 - Завдання типу класу.

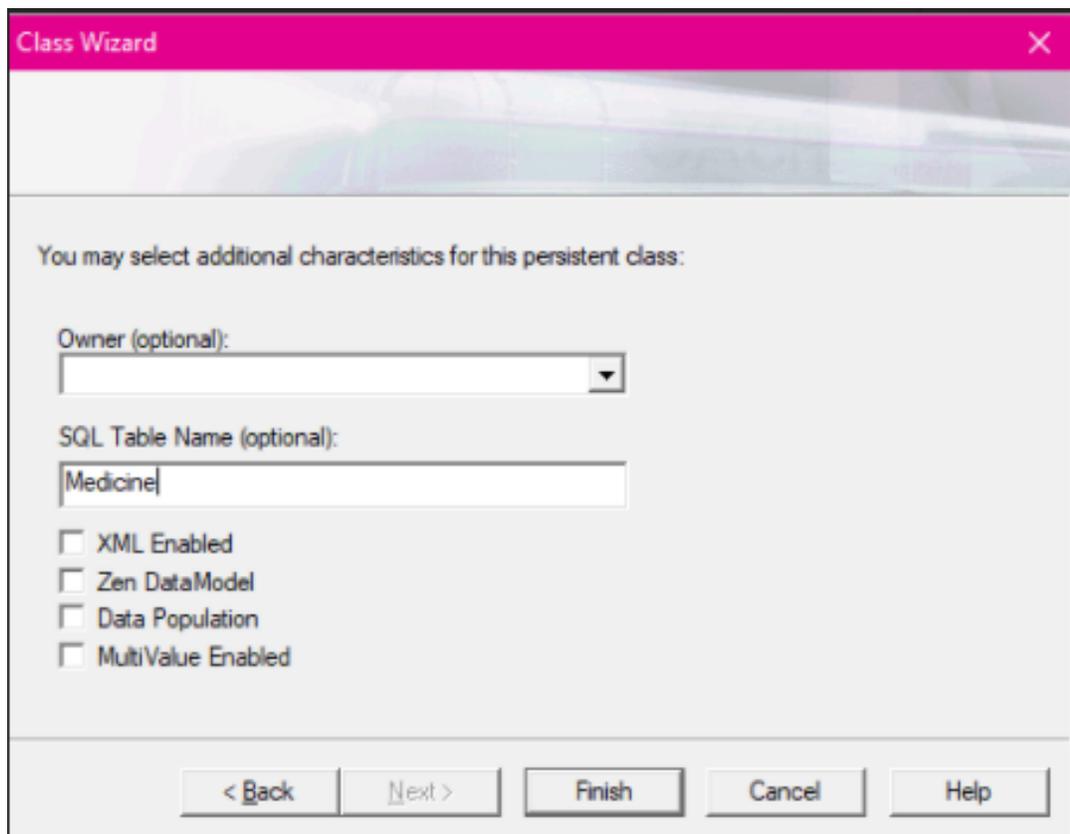


Рисунок 7. 6 - Завдання імені таблиці SQL (також Patient0).

Після цього у вікні редактора коду з'явиться опис класу на мові COS. Див. Рис. 7. 7.

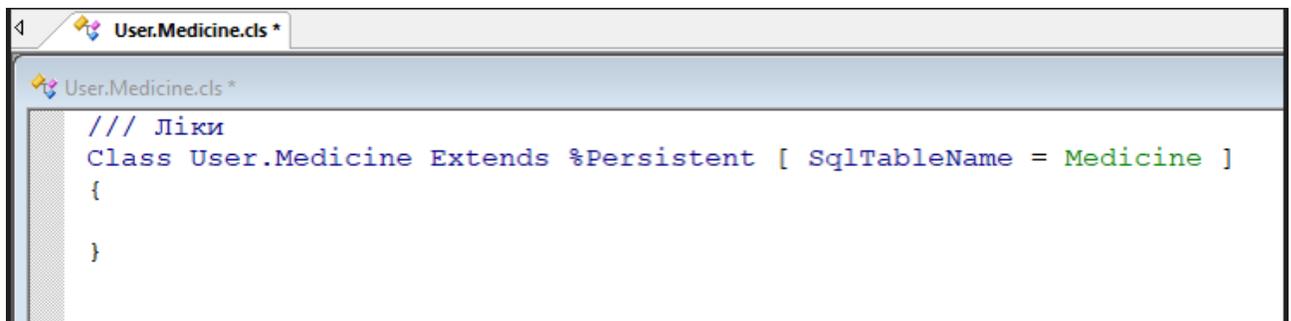


Рисунок 7. 7 - Опис класу на мові COS.

3. Створити властивості класу Medicine

TITLE – назва препарату;

CATEGORY - категорія;

PRICE - ціна у грн.;

DOSAGE – дозування у мг;

AMOUNT – кількість в упаковці;

APPLICATION – особливості застосування;

MANUFACTURER – підприємство- виробник та країна.

Для створення чергової властивості класу викликати майстер створення властивості, див. рис. 7.8. При цьому відкривається майстер властивостей, див рис. 7.9- 7.14.

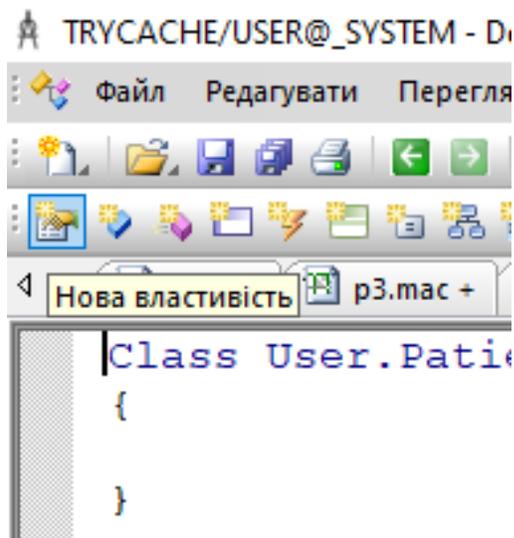


Рисунок 7. 8 - Створення нової властивості

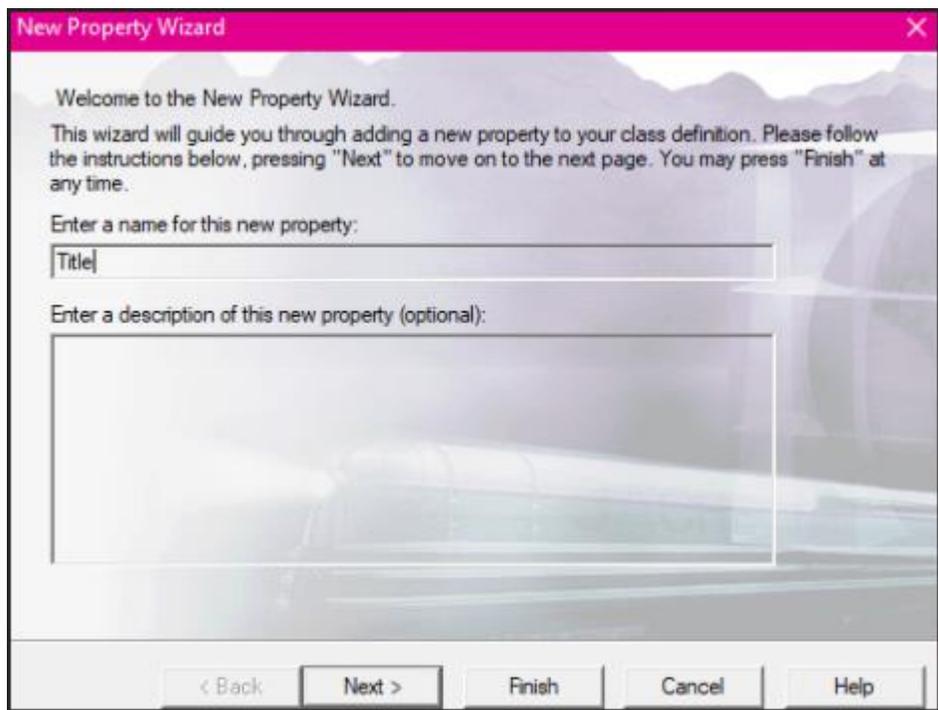


Рисунок 7. 9 - Завдання імені та опис властивості TITLE

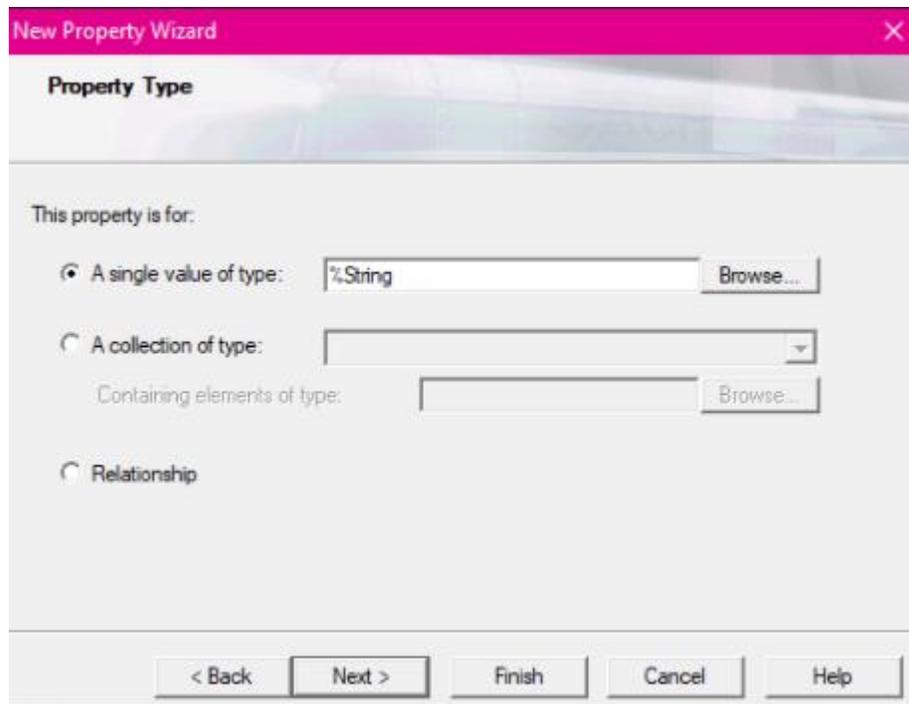


Рисунок 7. 10- Завдання типу властивості (% String - текстове поле).

УВАГА! При визначенні типів інших властивостей використовувати відповідно:

% Float - для ціни та дозування,

% Integer - для кількості в упаковці.

Для інших типів полів встановіть параметр %String.

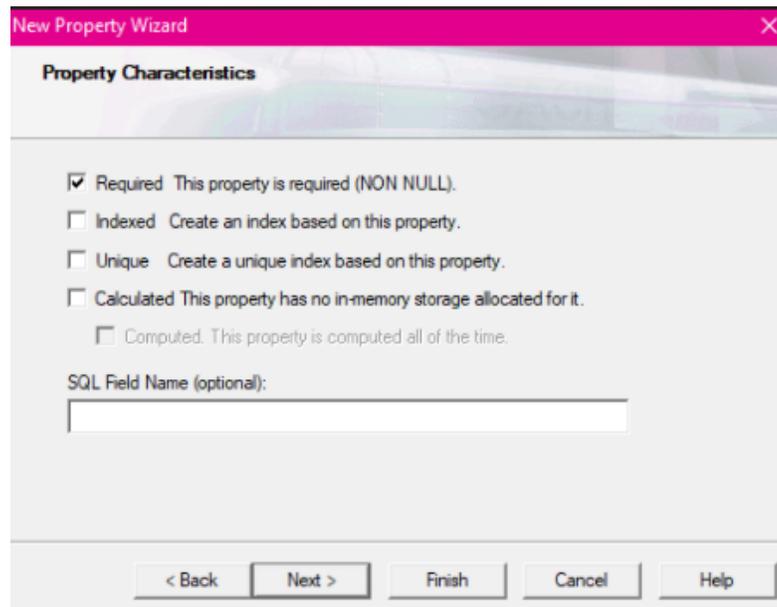


Рисунок 7. 11 - Показники властивості (Обов'язкове - означає, що порожнім це поле бути не може).

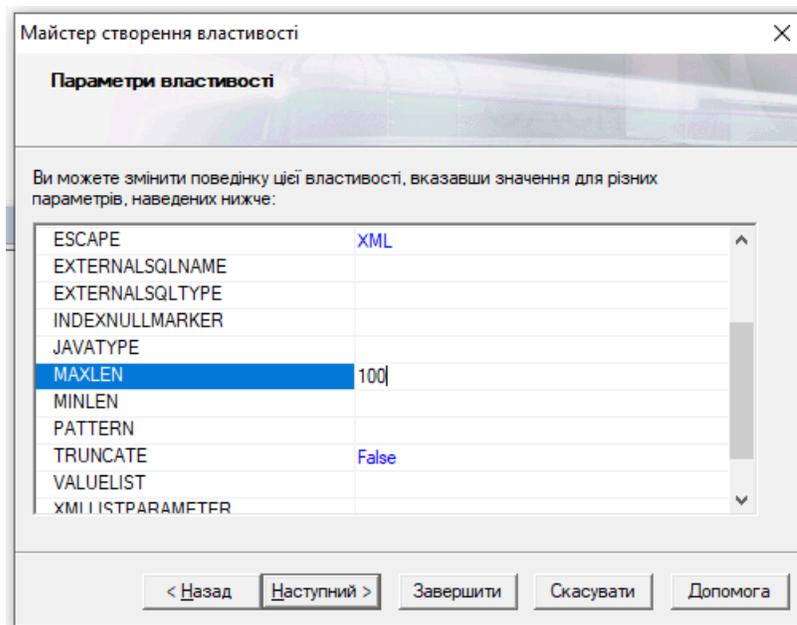


Рисунок 7. 12 – Встановлення для властивості APPLICATION типу %String з

параметром MaxLen – максимально допустима довжина поля - 100 символів

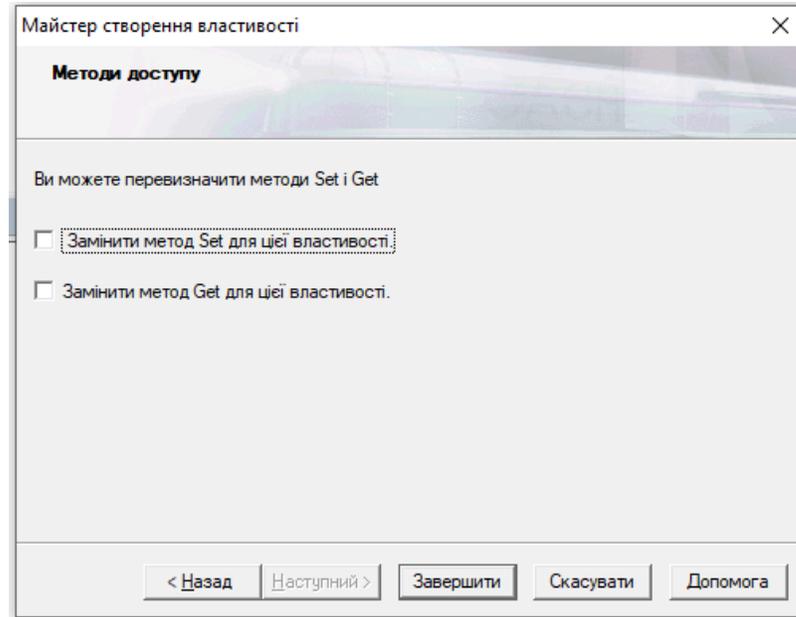


Рисунок 7. 13 - Завершення роботи майстра.

4. Після визначення всіх властивостей у вікні редактора коду повинна з'явитися наступна інформація:

```
User.Medicine.cls
/// Ліки
Class User.Medicine Extends %Persistent [ SqlTableName = Medicine ]
{
    Property Title As %String [ Required ];

    /// Категорія
    Property Category As %String [ Required ];

    /// Ціна у грн.
    Property Price As %Float [ Required ];

    /// Дозування
    Property Dosage As %Float [ Required ];

    /// Кількість в упаковці
    Property Amount As %Integer [ Required ];

    /// Особливості застосування
    Property Application As %String(MAXLEN = 100) [ Required ];

    /// Підприємство-виробник та країна
    Property Manufacturer As %String [ Required ];
}
```

Рисунок 7.14 – Файл із створеним класом Medicine та його властивостями

5. Відкомпілювати клас (Ctrl + F7). Виконуйте цю операцію завжди, коли Ви що-небудь змінюєте в описі класу, його властивостей, методів або запитів.
6. Створити інтерфейс для введення або пошуку потрібної інформації про пацієнтів. Для цього потрібно створити нову CSP - сторінку.

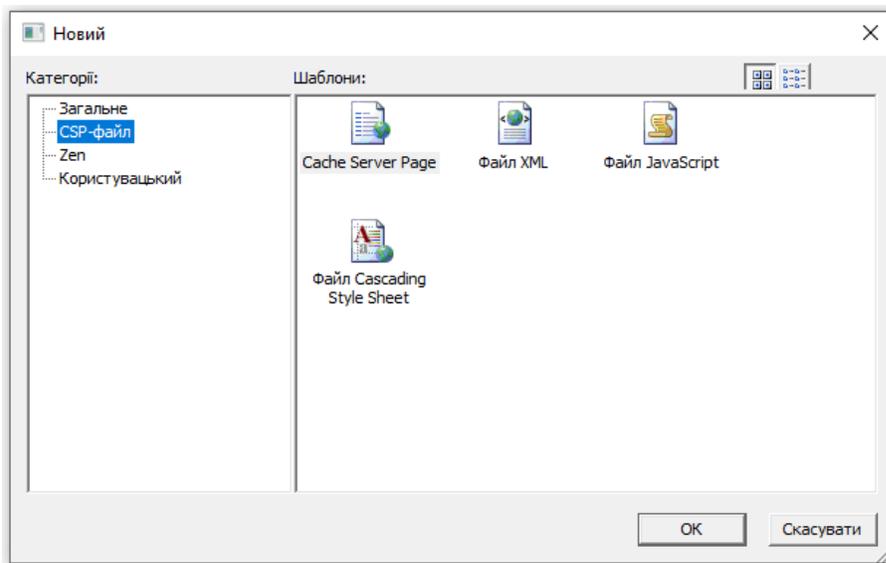
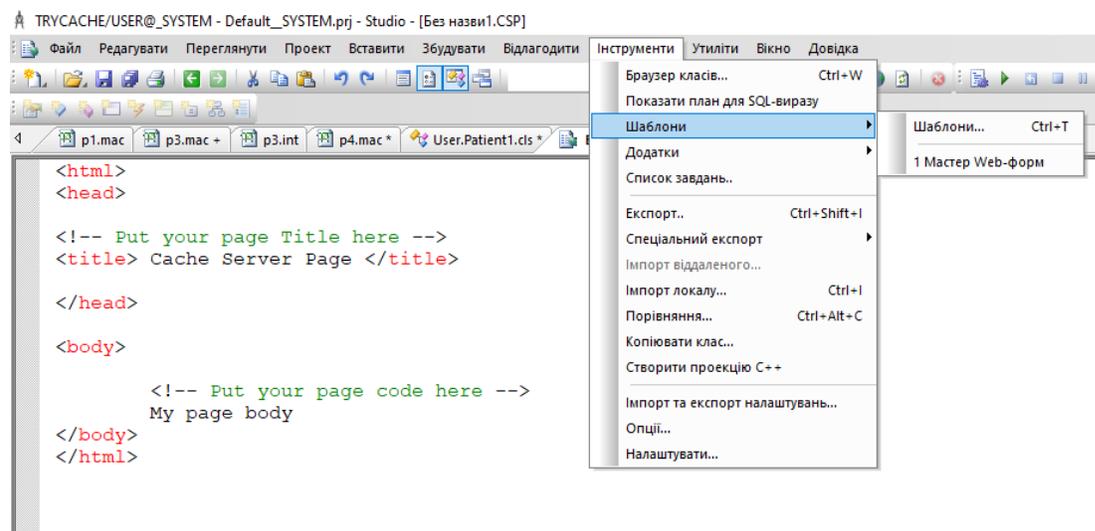


Рисунок 7. 15 - Створення нового CSP - файлу.

7. З пункту меню «Інструменти» вибрати «Шаблони», потім «Майстер Web- форм». Виконуючи вказівки майстра побудувати форму для введення значень.



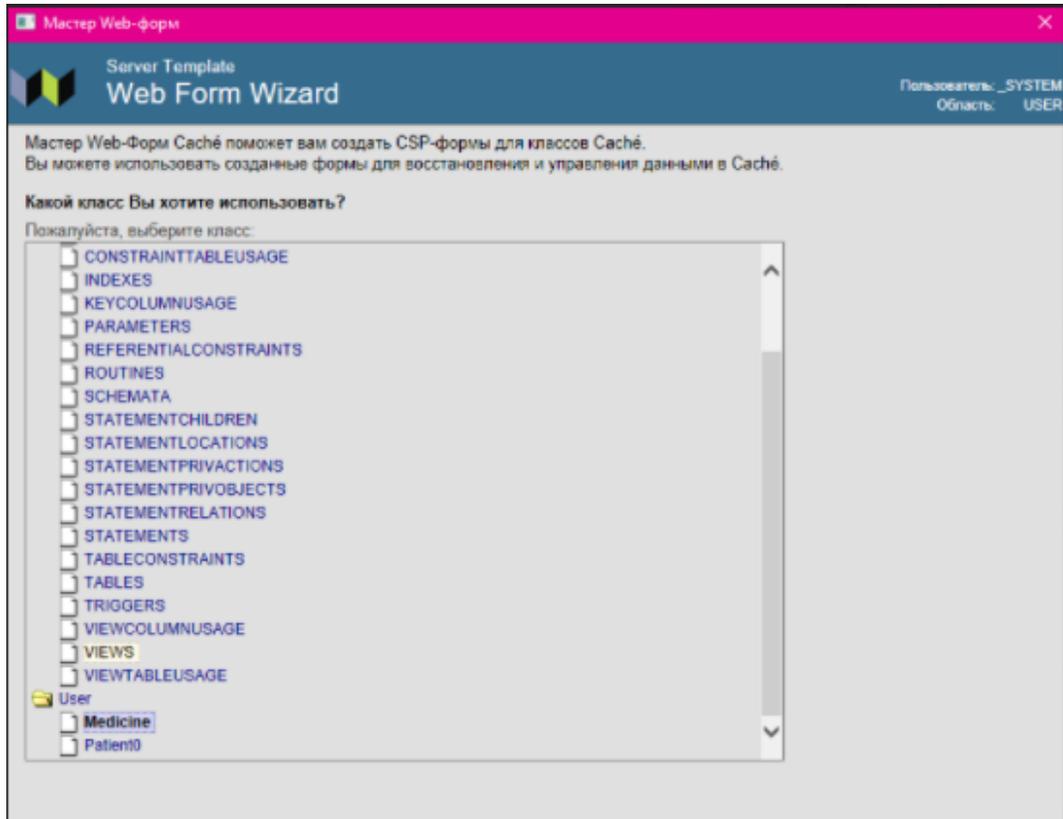


Рисунок 7. 16 - Мастер форм Cache

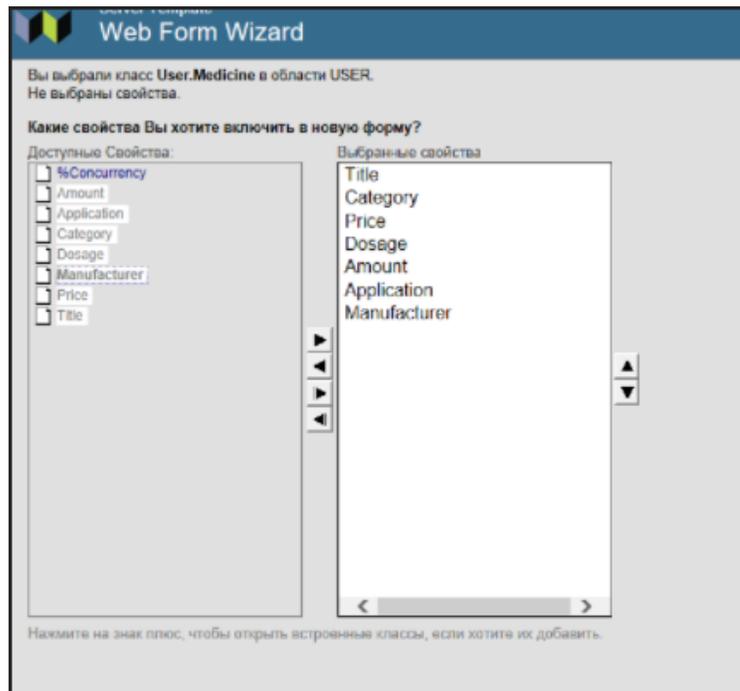


Рисунок 7. 17 - Выбор полей, доступных для введения значений властивостей

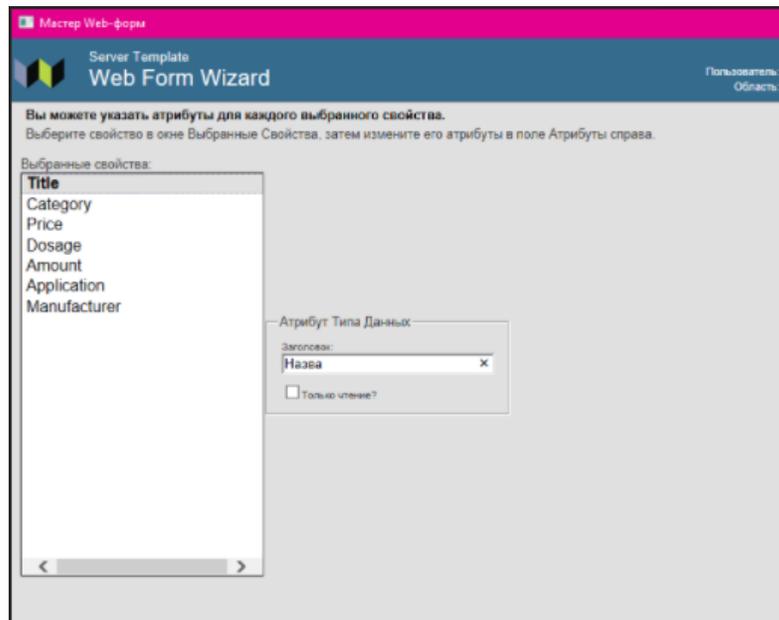
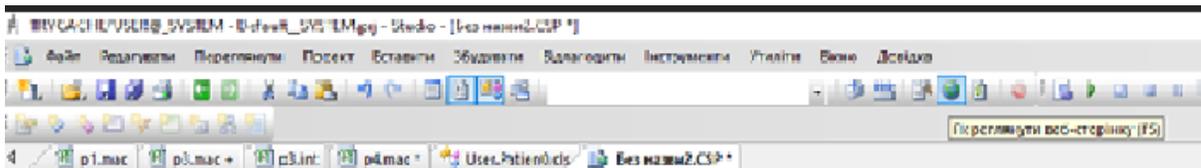


Рисунок 7. 18 - Створення підписів (заголовків) до полів форми

8. Після цього заповнити форму даними. Для цього відобразити csp - сторінку у вигляді веб – сторінки, натиснувши на панелі інструментів піктограму із земною кулею «Перегляд веб-сторінки» і внести не менше ніж 5 препаратів (інформацію можна взяти з сайту tabletki.ua), див. рис.7.19-7.22.



```

csp/user/Medicine.csp
<head>
<title>Cache Server Page - User.Medicine (USER)</title>
</head>
<h1 align='center'>User.Medicine</h1>
<!-- This function is needed by the search button on the form -->
<script language='javascript'>
<!--
function update(id)
{
    #server(..formLoad(id));
    return true;
}
// -->
</script>

<!-- set the domain so error messages could be properly localized -->
<cspp: class DOMAIN='%CSPBind'>

<!-- use CSP:OBJECT tag to create a reference to an instance of the class -->
<cspp:object name='objForm' classname='User.Medicine' OBJID='#(%request.Get("OBJID"))#'>

<!-- use csp:search tag to create a javascript function to invoke a search page -->
<cspp:search name='form_search' classname='User.Medicine' where='%Id()' options='popup,nopredicates' onselect='update'>

<form name='form' cspbind='objForm' cspjs='All' onsubmit='return form_validate();'>
<center>
<table cellpadding='3'>

```

Рисунок 7.19 – Як переглянути щойно створену веб -сторінку

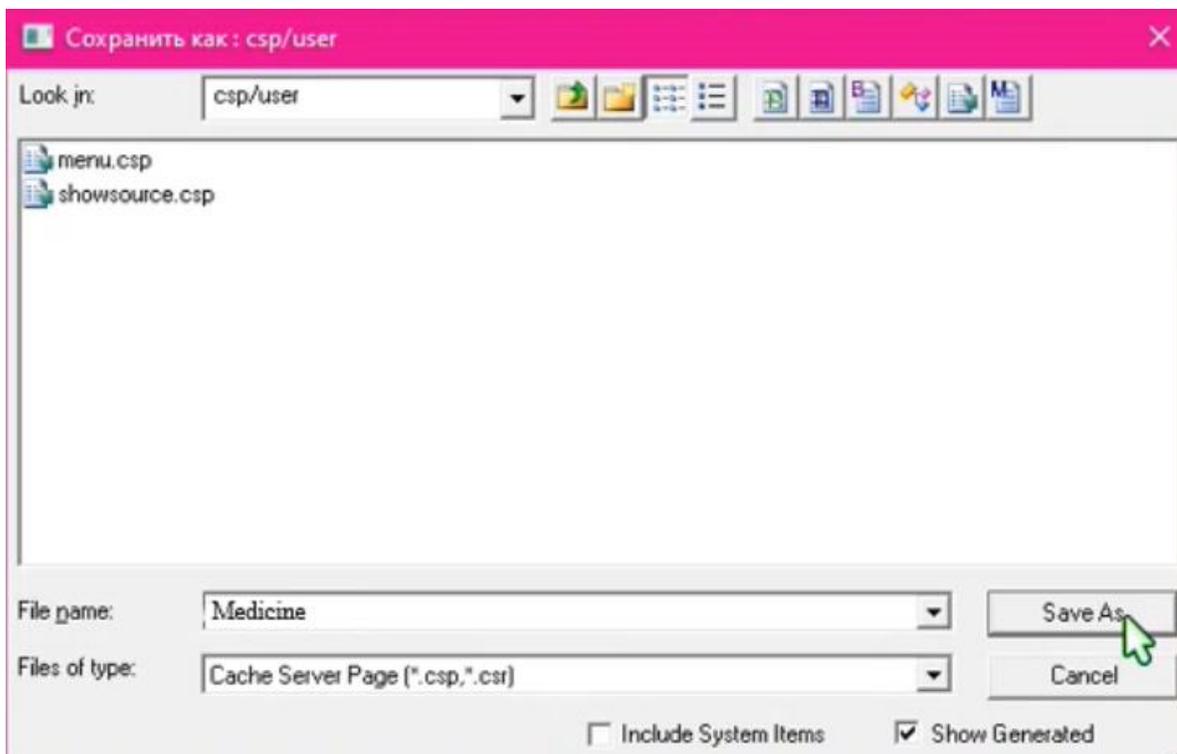


Рисунок 7. 20 - Збереження csp – сторінки, файлу надати ім'я «Medicine», розмістити цей файл треба у папці «csp/user»



Рисунок 7.21- Вигляд форми як веб-сторінки

User.Medicine

*Назва:

*Категорія:

*Ціна у грн.:

*Дозування:

*Кількість в упаковці:

*Особливості застосування:

*Підприємство-виробник та країна:

(* обязательные поля)

Рисунок 7. 22 - Заповнення форми у вигляді веб-сторінки

При заповненні форми даними про чергового пацієнта натиснути кнопку «Зберегти» і «Очищення» - для наступного пацієнта.

Питання для самопідготовки

1. Поясніть суть об'єктної технології проектування інформаційних систем.
2. Що таке об'єкт? Чим характеризується об'єкт з точки зору ГО-технології?
3. Поясніть суть понять «абстракція» і «інкапсуляція».
4. Що таке клас? Екземпляр класу? З точки зору ГО-технології?
5. Які переваги ГО-технології для проектування інформаційних систем?
6. Поясніть принцип «наслідування» і «поліморфізм».
7. Які елементи опису класу об'єктів можна використовувати при описі об'єктів за допомогою Cache?
8. Як на мові CDL (Class Definition Language) задається опис класу?
9. Як на мові CDL задається опис властивості? Навести приклад.

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПРАКТИЧНЕ ЗАНЯТТЯ №8

Тема: принципи організації web - сайту. Мова розмітки HTML.

Мета роботи: оволодіти навичками форматування електронних документів тегами мови HTML і створення web-сторінки. Вивчення основних концепцій HTML, необхідних для розуміння принципів розробки документів для мережі Інтернет.

Теоретичні відомості

1. Основні поняття

Глобальна обчислювальна мережа, ГОС (англ. Wide Area Network, WAN) являє собою комп'ютерну мережу, що охоплює великі території і включає в себе десятки і сотні тисяч комп'ютерів. ГОС служать для об'єднання розрізаних мереж так, щоб користувачі і комп'ютери, де б вони не знаходилися, могли взаємодіяти з усіма іншими учасниками глобальної мережі. Кращим прикладом ГОС є Інтернет, але існують і інші мережі, наприклад FidoNet.

HTML (HyperText Markup Language, мова гіпертекстової розмітки) - спеціальні інструкції браузеру, за допомогою яких створюються Веб-сторінки. Тобто Web-сторінки - це документи в форматі HTML, що містять текст і спеціальні **теги** (дескриптори) HTML. Теги HTML необхідні для форматування тексту (тобто надання йому потрібного вигляду), який "розуміє" браузер. Документи HTML зберігаються у вигляді файлів з розширенням .htm або .html.

Теги HTML повідомляють браузеру інформацію про структуру й особливості форматування web-сторінки. Кожен тег містить певну інструкцію і полягає в кутових дужках «< >». Більшість тегів складаються з частин, що відкривають і закривають тег і впливають на текст, укладений всередині.

Для більш швидкого знайомства з HTML-кодом можна переглядати вподобану вам сторінку в режимі "У вигляді HTML". Для цього в браузері треба

вибрати пункт меню "Вид" - " У вигляді HTML ". Після цього відкриється нове вікно з вихідним текстом HTML-коду. Переглядаючи готовий код HTML, ви дізнаєтеся, як теги мови застосовуються більш досвідченими програмістами. Крім того, це один із способів подолання різних проблем, що виникають час від часу.

Таким чином web-сторінка - це окремий документ Веб, який здатний утримувати інформацію різного виду - текст, малюнки, фото, відео, аудіозаписи. Такі веб-сторінки зберігаються на спеціальних комп'ютерах, які називаються серверами. Збережена на веб-сервері сторінка стає доступною для використання іншими користувачами. Колекція веб-сторінок, об'єднана за будь-якою ознакою, називається веб-сайтом або просто сайтом.

URL (Uniform Resource Locator) - це унікальна адреса, якою володіє кожна веб-сторінка в мережі. Якщо користувачеві відомий URL сторінки, то він може її відобразити у себе в браузері, набравши її адресу в рядку браузера.

У загальному випадку URL складається з найменування використовуваного протоколу, назви сервера і позначення шляху доступу до сторінки.

Протокол визначає правила звернення до веб-сторінки.

Назва сервера (доменне ім'я) - позначає комп'ютер, що містить дану веб-сторінку.

Шлях доступу - вказує місце розташування сторінки на диску сервера. Наприклад в адресі даної сторінки <http://learn.com/html/index.html>

http - найменування протоколу

learn.com - найменування сервера або доменне ім'я

/html/index.html - шлях доступу до сторінки на самому сервері.

Крім http (Hyper Text Transfer Protocol) протоколу, існують інші види протоколу, що застосовуються для доступу до даних в Internet, - наприклад, FTP (File Transfer Protocol). Сайти FTP, на які слід посилатися за допомогою аббревіатури ftp, дозволяють користувачеві вивантажувати на сервер дані, що

зберігаються у вигляді окремих файлів.

Web-браузер - це програма, встановлена на комп'ютері користувача, яка служить для пошуку і відображення інформації в мережі. Браузер зчитує веб-сторінки та інші файли з диска сервера та відображає їх вміст на моніторі комп'ютера користувача. Серед найбільш популярних браузерів слід назвати Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Safari.

Гіперпосилання дозволяють швидко переходити до інших веб-сторінок. Гіперпосилання виключають необхідність введення URL необхідної сторінки, до якої звертається користувач. Гіперпосилання, як правило, виділяються кольором і підкресленням. Як гіперпосилання може виступати графічне зображення, в такому випадку воно зазвичай виділяється рамкою.

2. Структура HTML-документу

Існує кілька основних тегів HTML, які повинні бути присутніми в тексті будь-якої Веб-сторінки. Кожна Веб-сторінка повинна містити тег <HTML>, що розташовується в самому початку. Цей тег описує ваш документ як Веб-сторінку, виконану в форматі HTML. Безпосередньо за дескриптором <HTML> зазвичай знаходиться тег <HEAD>, який вказує на наявність тексту, що містить найменування сторінки і додаткові відомості про неї.

В розділ HEAD зазвичай вкладений тег <TITLE>, що використовується для позначення найменування сторінки. Найменування сторінок зазвичай відображаються в рядку заголовка вікна браузера. Назва Веб-сторінки може бути створена лише за допомогою символів букв, цифр і пробілу.

Потім розташовується тег <BODY>, який вказує на початок власне "тіла" Веб-сторінки. В цьому розділі розміщуються весь інший текст, графіка, таблиці та інші елементи вмісту сторінки, які побачить відвідувач, який звернувся до вашого сайту.



Рисунок 8.1 - Структура веб-сторінки

Всі вищевказані теги відносяться до так званих парних тегів. Парні теги, або контейнери, складаються з двох частин – відкриваючого та закриваючого тегів. Закриваючий тег відрізняється від відкриваючого наявністю символу слеш перед назвою тегу (див. рис.8.1). Таким чином парні теги виділяють частину документу, на яку поширюється «вплив» цих тегів. Існують також непарні теги, які не потребують закриваючої частини. До таких відносяться, наприклад, теги `
` - перехід на нову строку, `` - додавання малюнку та ін.

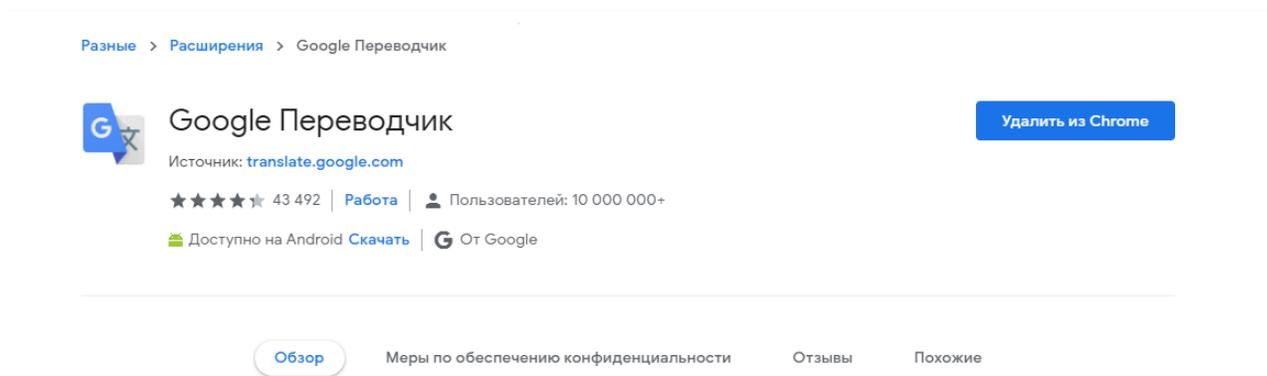
Гарний HTML-редактор вже має набір вищевказаних тегів при створенні нової сторінки. Варто зазначити, що до вибору найменувань Веб-сторінок слід поставитися з усією серйозністю, тому що вони використовуються пошуковими системами для формування каталогів Веб-сторінок. Пошукова система - це сайт, який допомагає користувачам швидко знаходити цікаву для них інформацію. Ваша сторінка з більшою ймовірністю опиниться знайденою пошуковою

системою, якій поставлено певний набір ключових слів пошуку, якщо ці слова присутні в назві сторінки.

Порядок виконання роботи

1. Вивчити теоретичний матеріал з мови HTML.
2. Практично вивчити прийоми роботи і отримати навички розмітки документа за допомогою мови HTML. Нижче наведені фрагменти коду, які слід набрати в HTML-редакторі (<https://www.w3schools.com/html/>). Далі необхідно спостерігати, як саме відображається документ.

Для більшої зручності можна встановити розширення для браузеру Google Chrome **Google Перекладач**:



Та виконувати практичну роботу на українській мові.

Обрати сторінку, яку треба перекласти, натиснути праву кнопку миші та обрати – Перевести на українську.

Приклад HTML:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Спробуйте самі »

3. **Найпростіше форматування тексту.** Для створення абзацу слугує парний тег **<P>**. Для додавання фрагменту тексту смислового значення зручно використовувати засоби зміни накреслення шрифту. Для виділення тексту напівжирним шрифтом використовується тег **** або ****, Для виділення курсивом - **<I>** або ****, Для підкреслення тексту - **<U>**, Для закреслення тексту - **<STRIKE>**. Фрагменти напівжирного або курсивного тексту використовують для введення нових термінів, вираження важливих думок. З підкресленням тексту треба бути дуже обережним, тому що в Інтернеті таким чином прийнято виділяти гіперпосилання. Тег **<CENTER>** дозволяє центрувати текст на сторінці по горизонталі (рис.8.2).

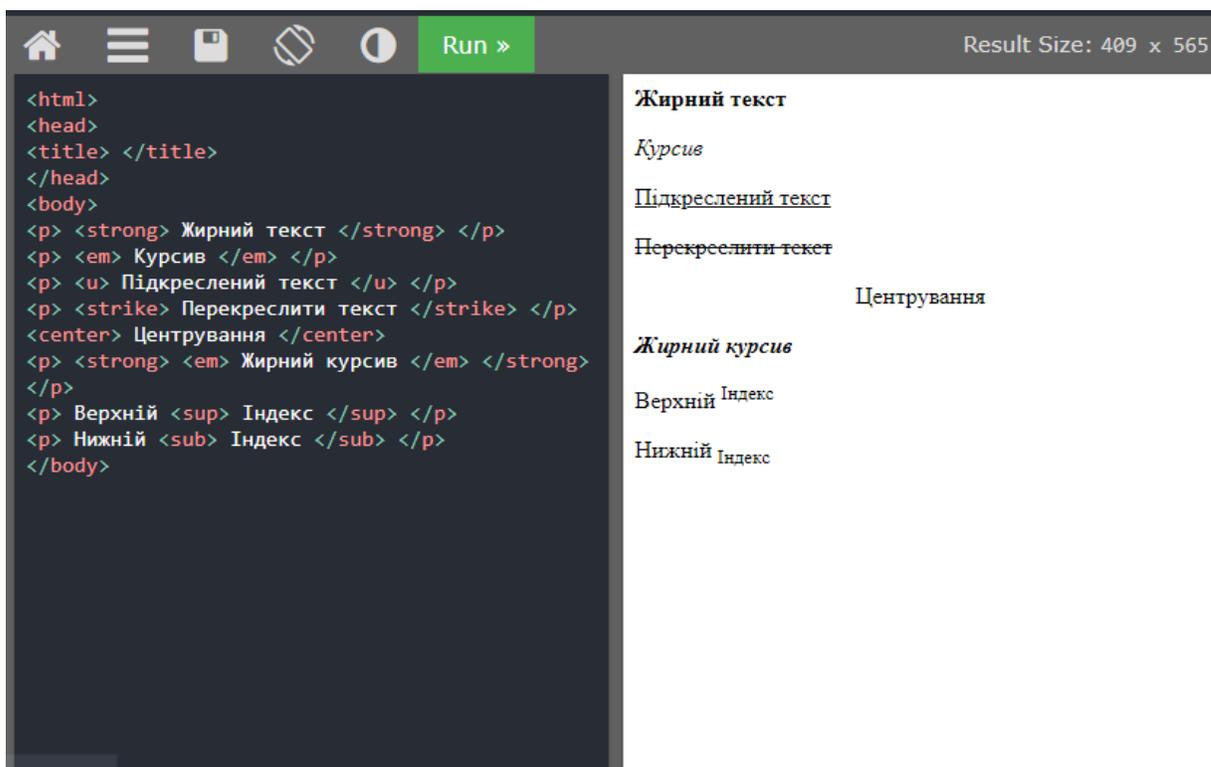


Рисунок 84.2 – HTML-код простого форматування тексту та його вигляд у онлайн редакторі **w3schools.com** .

Це зручно для виділення найменувань або заголовків. HTML дозволяє одночасно застосовувати декілька дескрипторів, призначених для зміни накреслення шрифту. Однак в цьому випадку необхідно стежити, щоб завершальні дескриптори, що містять символ косою риси, вводилися в зворотному порядку. Для перетворення тексту в нарядковий індекс застосовують тег **<SUP>**, для підрядкового індексу - **<SUB>**. Ці теги знадобляться при введенні математичних текстів, хімічних формул, наукових статей, побудові виносок і т.і. Зазвичай індекси відображаються шрифтом меншого розміру в порівнянні з основним. Цю обставину слід враховувати при виборі розміру основного шрифту, тому що при малому розмірі індекси можуть бути просто не видно.

4. Для зміни **розміру шрифту** використовують атрибут SIZE. Атрибути

знаходяться у відкриваючому тегу після його назви (див. рис.8.3). Один тег може містити кілька атрибутів. Чим більший шрифт, тим легшим для сприйняття стає текст. Однак, дрібний шрифт дозволяє вмістити в межах екрану більший обсяг інформації.

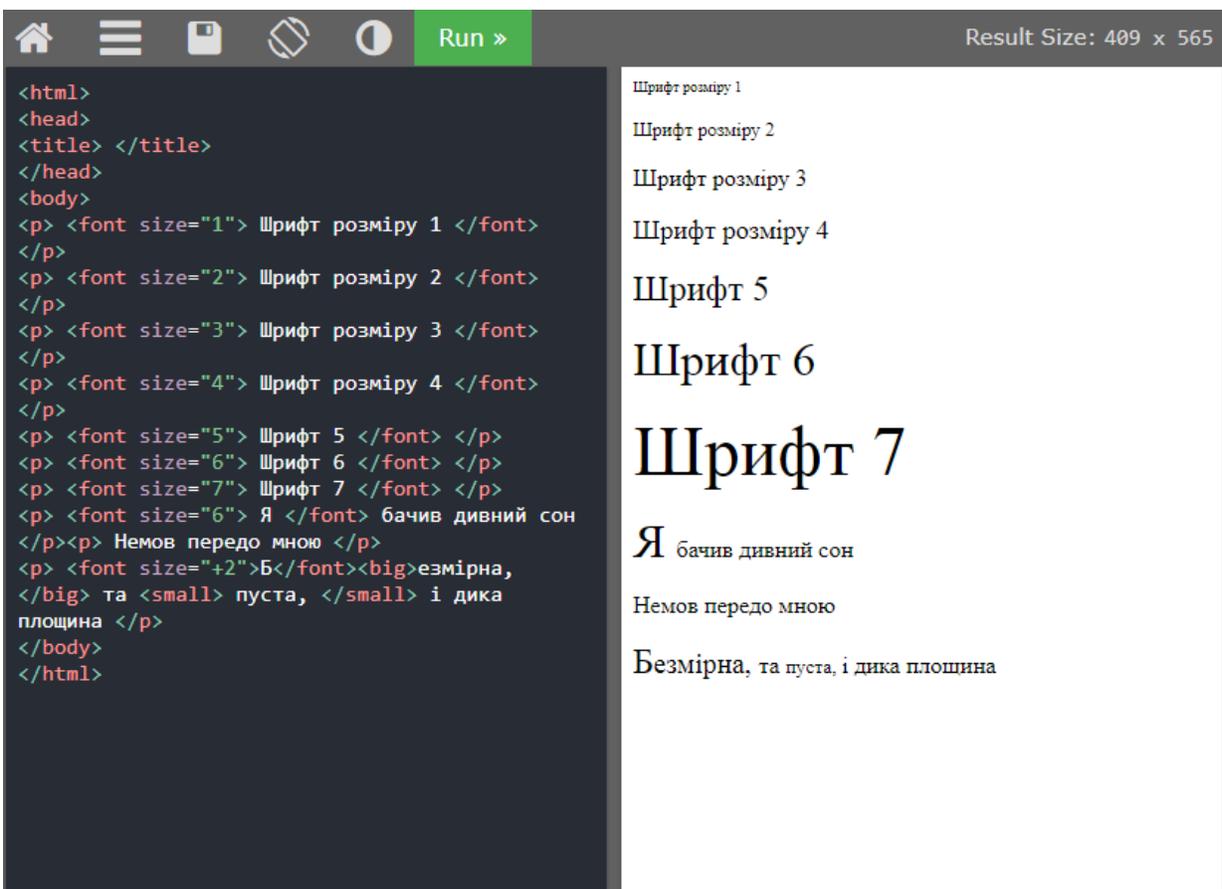


Рисунок 8.3 – Форматування розміру і гарнітури шрифту.

Використовуючи даний атрибут з тегом `<BASEFONT>`, Ви можете змінити розмір тексту веб-сторінки цілком. Застосовуючи атрибут `SIZE` спільно з тегом ``, можна впливати на зовнішній вигляд окремого фрагменту тексту. Доступно 7 розмірів шрифтів. Найменший позначається цифрою 1, а найбільший - 7. Змінювати шрифт можна за допомогою тегів `<BIG>` і `<SMALL>`. `<BIG>` збільшує шрифт виділеного фрагменту по відношенню до попереднього тексту, а `<SMALL>`, відповідно, зменшує. Змінюючи розмір окремих букв, можна

домогтися цікавих ефектів. Наприклад, таким чином можна отримати збільшену прописну букву початку абзацу.

5. Форматування списків. Для створення нумерованих списків використовуються теги `` ``. тегом `` відзначається початок / закінчення всього списку. тегом `` позначають початок / кінець окремого елемента списку (рис.8.4). За замовчуванням елементи списку нумеруються за порядком 1, 2, 3 ...

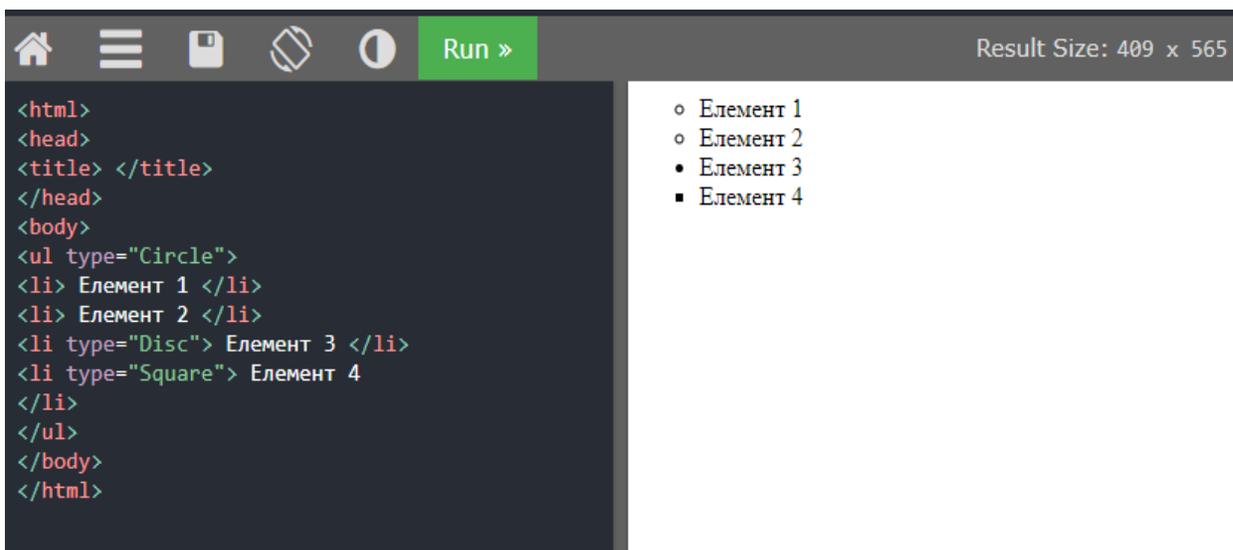


Рисунок 8.4 – Створення списків.

За допомогою атрибута `TYPE` можна змінити стиль нумерації. Для створення маркованих списків застосовують теги ``, `` Тегом `` зазначається початок / закінчення всього списку. Тегом `` позначають початок / кінець окремого елемента списку. За замовчуванням елементи списку маркуються чорним кружечком. За допомогою атрибута `TYPE` можна змінити стиль маркування. В межах одного списку можна використовувати різне маркування елементів списку.

6. Гіперпосилання. Гіперпосилання може пов'язувати сторінки як в межах одного сайту, так і вказувати на будь-яку сторінку в Інтернеті (див. рис.8.5

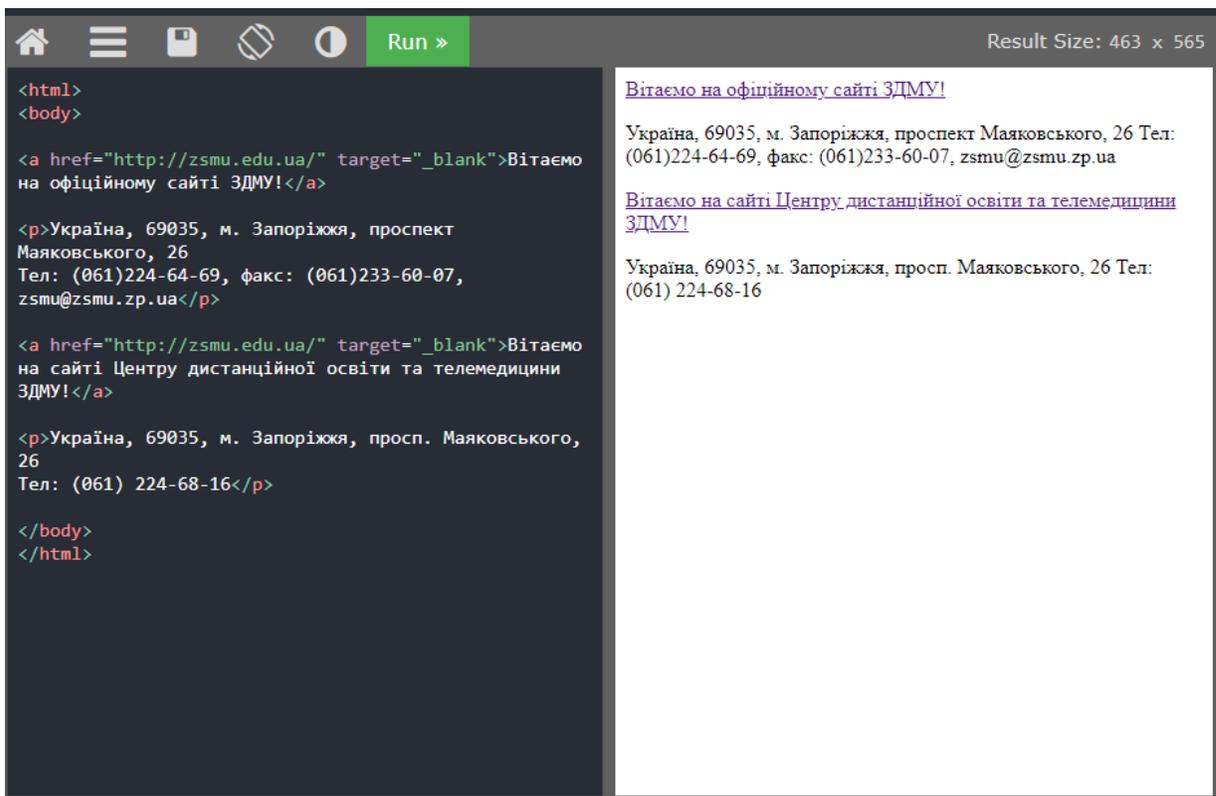


Рисунок 8.5 – Вставка гіперпосилань

). При побудові посилання на чужі сторінки завжди треба користуватися абсолютною адресою сторінки (<http://www.site.com/page.html>). Якщо створюється посилання на сторінку в межах сайту, то краще використовувати відносний URL ([page.html](#), [catalog/page.html](#)). Роблячи графічне гіперпосилання, пам'ятайте, що деяким користувачам графіка недоступна, тому обов'язково включайте відповідні текстові елементи. Для створення гіперпосилання служить дескриптор `<A>`. За допомогою атрибута TARGET можна завантажити сторінку в новому вікні браузера. Цей атрибут призначений для завдання найменування вікна. Ім'я вікна використовується в службових цілях. Щоб відкрити сторінку в новому вікні треба використовувати константу `_blank`.

7. Таблиці. Таблиці у форматі HTML створюються за допомогою парного тегу `<table>`. Строки таблиці задаються парним тегом `<tr>`, а окремі комірки – тегом `<td>` (див.рис.8.6-8.7).

```

<html>
<head>
<title> Список та таблиця </title>
</head>
<body>
<font color="# Ff0000">
<h1> Електронна медична картка. </h1>
</font>
<p align="Justify"> Електронний медичний запис, Електронна медична картка (електронна історія хвороби; електронний паспорт пацієнта; англ. electronic medical record(EMR), electronic patient record, electronic health record (EHR)) – медична картка пацієнта медзакладу в електронній (безпаперовій) формі. Складається і зберігається в автоматизованій інформаційній базі даних медичної установи. Електронні картки мають поступово замінити традиційні картки медичного страхування. Картка містить електронний носій, котрий зберігатиме історію хвороб пацієнта, інформацію про зроблені щеплення та його бажання стати донором.
</p>
<h4> Закони: </h4>
<ol>
<li> Закон України «Про стандартизацію» за № 3164-IV від 01.12.2005 року </li>
<li> Закон України «Про електронні документи та електронний документообіг» за № 851-IV від 22.05.2003 року </li>
<li> Закон України «Про електронний цифровий підпис» від 15.01.2009 за № 879-17 </li>
</ol>
<center>
<table border="1">
<tr align="Center">
<td> № п/п </td>
<td> Прізвище пацієнта </td>
<td> Дата народження </td>
</tr>
<tr align="Center">
<td> 1 </td>
<td> Іванов П.Ю. </td>
<td> 23.11.2000 </td>
</tr>
</table>
</center>
</body>
</html>

```

Рисунок 8.6- Створення таблиць (HTML-код).

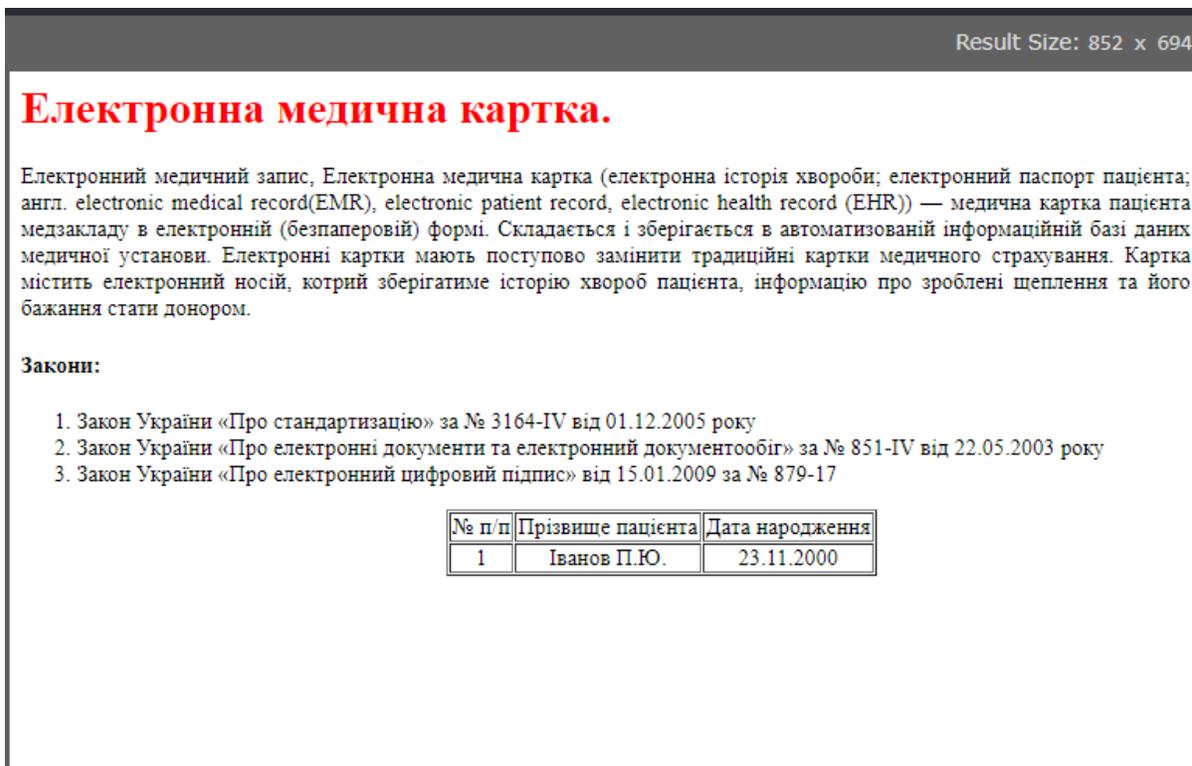


Рисунок 8.7 – Створення таблиць (вигляд у онлайн редакторі).

8. Створити на мові HTML таблицю з переліком 5-6 препаратів з обраної кожним студентом самостійно категорії ліків на сайті tabletki.ua. Зразок виконання такої вправи показаний на рис.8.8.

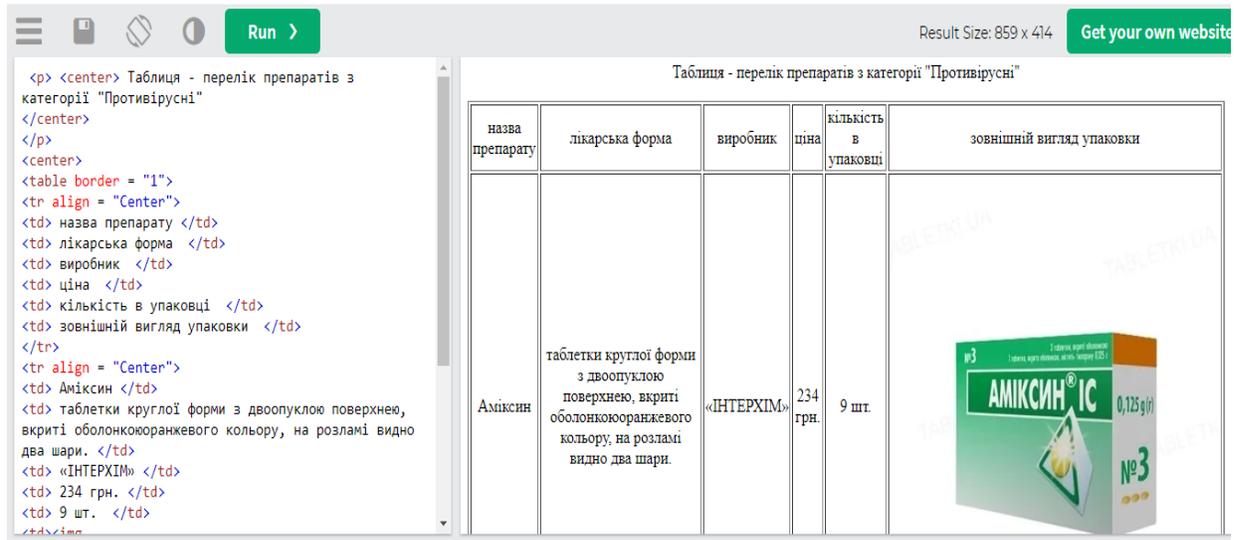


Рисунок 8.8 – Фрагмент таблиці ліків, створеної мовою HTML

Фрагмент лістингу програми на мові HTML для створення таблиці ліків (преші два рядки) за зразком (рис.4.8), наведений нижче:

```
<p> <center> Таблиця - перелік препаратів з категорії "Противірусні"
</center>
</p>
<center>
<table border = "1">
<tr align = "Center">
<td> назва препарату </td>
<td> лікарська форма </td>
<td> виробник </td>
<td> ціна </td>
```

```

<td> кількість в упаковці </td>
<td> зовнішній вигляд упаковки </td>
</tr>
<tr align = "Center">
<td> Аміксин </td>
<td> таблетки круглої форми з двоопуклою поверхнею, вкриті оболонкою
оранжевого кольору, на розламі видно два шари. </td>
<td> «ІНТЕРХІМ» </td>
<td> 234 грн. </td>
<td> 9 мм. </td>
<td><img src=https://cdn.tabletki.ua/img/goods/61e128ab-b98e-4c44-89d1-
2653bbffa0c4/img_0.jpg?v=AAAAAAmPFUs </td>
</tr>
</table>
</center>
</body>

```

Для вставлення картинки зовнішнього вигляду упаковки ліків з сайту tabletki.ua можна скористатися можливостями редактору <https://www.w3schools.com/html>, який рекомендує отримати доступ до зображень з будь-якої веб-адреси Світової павутини, прописавши повний шлях до зображення. Наприклад,

```



```

При створенні HTML-документу можна користуватися програмою – помічником, що знаходиться за посиланням

https://www.w3schools.com/html/html_editors.asp.

Питання для самопідготовки:

1. Дати визначення глобальної комп'ютерної мережі.
2. Для чого використовується мова HTML?
3. Що таке браузер? Які браузери Ви знаєте?
4. Що таке теги? Перерахувати основні теги мови HTML для форматування тексту.
5. Що таке гіперпосилання? Для чого вони потрібні?
6. Як створити таблицю в форматі HTML?
7. Сфери застосування мови HTML.

Підсумковий контроль

Після перевірки викладачем виконаного завдання пройдіть підсумкове тестування за темою на комп'ютері у комп'ютерному класі або дистанційно.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ:

Основна

1. Форкун Ю. В. Інформатика : навч. посіб. / Ю. В. Форкун, Н. А. Длугунович. - – Львів : Видавництво «Новий світ – 2000», 2018. – 464 с.
2. Ришковець Ю.В., В.А., Висоцька В.А. Алгоритмізація та програмування. Ч. 1 : навч. посіб. – Львів : Видавництво «Новий світ – 2000», 2018. – 337 с.
3. Ришковець Ю.В., В.А., Висоцька В.А. Алгоритмізація та програмування. Ч. 2 : навч. посіб. – Львів : Видавництво «Новий світ – 2000», 2018. – 316 с.
4. Інформаційні технології у фармації: навчальний посібник / Рижов О.А., Іванькова Н.А., Нессонова М.Н., Строїтелева Н.І. – Львів: видавець Марченко Т.В., 2020. – 212 с.
5. Формалізація, алгоритмізація та програмування фармацевтичних задач засобами Turbo-Pascal : навчальний посібник для студентів фармацевтичного факультету спеціальності 226 «Фармація, промислова фармація» / Рижов О.А., Строїтелева Н.І. – Запоріжжя: ЗДМУ, 2019. – 141 с.
6. Павлиш В.А. Основи інформаційних технологій і систем: підручник / В. А. Павлиш, Л. К. Гліненко, Н. Б. Шаховська. — Львів: Львівська політехніка, 2018. — 620 с.
7. Основи медичної інформатики. Практикум (навчальний посібник) — Т.І. Бондаренко, Всеукраїнське спеціалізоване видавництво «Медицина», 2018, 272с.
8. Бойко А.І. Розвиток теорії фармацевтичної інформатики в Україні: Монографія. Львів: кварт.2020. – 198с.
9. Бородкіна І.Л. Теорія алгоритмів. Посібник для студентів вищих навчальних закладів. Вид-во: Центр навчальної літератури: 2019, 184 с.
10. Буйницька О.П. Інформаційні технології та технічні засоби навчання. Вид-во: Центр навчальної літератури: 2019, 240 с.

Допоміжна

1. Комп'ютери та комп'ютерні технології: навч. посіб. / Ю. Б. Бродський, К. В. Молодецька, О. Б. Борисюк, І. Ю. Гринчук. – Житомир : Вид-во «Житомирський національний агроекологічний університет», 2017. – 186 с.
2. Рижов О.А., Іанькова Н.А., Андросов О.І. Організація інформаційного середовища студента на базі хмарних технології MS Office 365- методична розробка по проведенню практичних занять з курсу- Запоріжжя. :ЗДМУ, 2017- 32 с.
3. Рижов О.А., Іанькова Н.А., Андросов О.І. Основи проектування навчальних елементів для онлайн курсів edX. - методична розробка по проведенню практичних занять з курсу.- Запоріжжя. :ЗДМУ, 2017. – 22с.
4. Актуальні питання сучасної медицини і фармації - 2021: зб. тез доповідей наук.-практ. конф. з міжнар. участю молодих вчених та студентів 15-16 квітня 2021 р. / МОЗ України, ЗДМУ, наук. т-во студентів, аспірантів, докторантів і молодих вчених ; голова оргком.: Ю. М. Колесник. - Запоріжжя : ЗДМУ, 2021. - 201 с.
5. Подлевський Б.М. Теорія інформації в задачах Автори: Б.М. Подлевський, Р.Є. Рикалюк Вид-во: Центр навчальної літератури: 2017, 271 с.
6. Карпенко М.Ю. Технології створення програмних продуктів та інформаційних систем. навч. посібник. Автори: М. Ю. Карпенко, Н.О. Манакова, І.О. Гавриленко; Харків : ХНУМГ ім. О.М. Бекетова, 2017. - 93 с .
7. Булах, І. Є. Комп'ютерне моделювання у фармації : навч. посіб. / І. Є. Булах, Л. П. Войтенко, І. П. Кривенко. - 2-ге вид., випр. - Київ : Медицина, 2017. - 208 с.

Інформаційні ресурси:

1. Національна бібліотека України ім. Вернадського. <http://www.nbu.gov.ua/>
2. Офіційний сайт ЗДМУ. <http://zsmu.edu.ua/>
3. Наукова бібліотека ЗДМУ: <http://library.zsmu.edu.ua/>
4. Національна Медична Бібліотека США (National Library of Medicine) – вільний та безкоштовний доступ до MEDLINE
<http://www.ncbi.nlm.nih.gov/PubMed>
5. Кафедра медичної інформатики та комп'ютерних технологій навчання Національного медичного університету імені О. О. Богомольця
<http://nmuofficial.com/zagalni-vidomosti/kafedri/department-medical-biological-physics/>
6. Віртуальна хімічна лабораторія <http://chemcollective.org/vlab>
7. Віртуальна хімічна лабораторія <http://www.chem.ox.ac.uk/vrchemistry/>
8. Офіційний сайт розробників програми комп'ютерного моделювання Scilab: безплатне дистрибутиви, навчальні матеріали, підтримка <https://www.scilab.org/>
9. Компендіум, лікарські препарати <http://www.compendium.com.ua>
10. Офіційний сайт OpenOffice.org <http://www.openoffice.org/>