

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

Зміст

Лабораторна робота №1	3
Назва роботи: Створення найпростіших Windows-додатків в середовищі Visual Studio .NET.	3
Мета роботи: Навчитися створювати найпростіші Windows-додатки в середовищі Visual Studio .NET	3
Теоретичні відомості	3
Хід виконання	12
Варіанти завдань для самостійної роботи	12
Зміст звіту:	13
Контрольні запитання	13
Лабораторна робота №2	15
Назва роботи: Клас та об'єкт	15
Мета роботи: Ознайомитися з особливостями ООП, навчитися створювати класи та об'єкти	15
Теоретичні відомості	15
Приклад створення об'єкту певного класу	16
Хід виконання	17
Варіанти завдань для самостійної роботи	17
Зміст звіту:	19
Контрольні запитання	19
Лабораторна робота №3	20
Назва роботи: Масиви об'єктів	20
Мета роботи: Ознайомитися з особливостями ООП, навчитися створювати масиви об'єктів	20
Теоретичні відомості	20
Приклад створення масиву об'єктів	20
Хід виконання	21
Варіанти завдань для самостійної роботи	21
Зміст звіту:	23
Контрольні запитання	23
Лабораторна робота №4	24
Назва роботи: Наслідування	24
Мета роботи: Ознайомитися з особливостями ООП та його основними властивостями, навчися використовувати ідеологію наслідування.	24
Теоретичні відомості	24
Хід виконання	25
Хід виконання	33
Зміст звіту:	33
Контрольні запитання	33
Лабораторна робота №5	35
Назва роботи: Використання механізму віртуальних функцій	35
Мета роботи: Ознайомитися з особливостями використання віртуальних функцій.	35
Теоретичні відомості	35
Завдання	35
Хід виконання	37
Зміст звіту:	37
Контрольні запитання	38
Лабораторна робота №6	39
Назва роботи: Абстрактні класи	39

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об’єктно-орієнтоване програмування»**

Жуковський В.В.

Мета роботи: Навчитись використовувати абстрактні класи та інтерфейси.	39
Теоретичні відомості	39
Завдання	41
Хід виконання	41
Зміст звіту:	41
Контрольні запитання	41
Лабораторна робота №7	42
Назва роботи: Простір імен System.Drawing	42
Мета роботи: Навчитись використовувати класи з простору імен System.Drawing для рисування на формі.	42
Теоретичні відомості	42
Завдання	42
Варіанти завдань для самостійної роботи	42
Хід виконання	43
Зміст звіту:	43
Контрольні запитання	43
Лабораторна робота №8	44
Назва роботи: Регулярні вирази	44
Мета роботи: Навчитись використовувати класи з простору імен RegularExpression для операцій з текстом.	44
Теоретичні відомості	44
Завдання	44
Варіанти завдань для самостійної роботи	44
Хід виконання	44
Зміст звіту:	45
Контрольні запитання	45
Лабораторна робота №9	46
Назва роботи: Породжуючі паттерни	46
Мета роботи: Навчитись використовувати породжуючі паттерни абстрактна фабрика, будівельник та фабричний метод.	46
Теоретичні відомості	46
Завдання	46
Хід виконання	46
Зміст звіту:	46
Контрольні запитання	47
Лабораторна робота №10	48
Назва роботи: Структурні паттерни	48
Мета роботи: Навчитись використовувати структурні паттерни згідно поставленої задачі	48
Теоретичні відомості	48
Завдання	48
Хід виконання	48
Зміст звіту:	49
Контрольні запитання	49

Лабораторна робота №1

Назва роботи: Створення найпростіших Windows-додатків в середовищі Visual Studio .NET.

Мета роботи: Навчитися створювати найпростіші Windows-додатки в середовищі Visual Studio .NET

Теоретичні відомості

IDE (Integrated Development Environment) це інтерактивне середовище розробки програмних продуктів. Це середовище (програма або комплекс програм) в якій програміст створює програми, компілює їх, підлагоджує, супроводжує ітд. Тобто необхідний набір інструментальних засобів для циклу розробки ПЗ. Середовища з якими ви могли стикатися до цього це Turbo Pascal, Turbo Basic, Turbo/Borland c, DevCpp, Delphi та інші, вони містили в собі зручний редактор, довідкову інформацію, компілятор ітд.

Microsoft Visual Studio — лінійка продуктів компанії Майкрософт, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів.

Найбільш значущі версії пакету:

Visual Studio — найкраще комплексне середовище IDE для розробників .NET і C++ у Windows. Повноцінний набір інструментів та функцій для покращення та вдосконалення кожного етапу розробки програмного забезпечення.

Visual Studio Code — засіб для створення, редагування та налагодження (debugging) сучасних вебзастосунків і програм для хмарних систем. Це є найкращий вибір для JavaScript та веб-розробників, з розширеннями, що підтримують будь-які мови програмування. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X. Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015.

Visual Studio для Mac — Комплексна IDE для розробників .NET, вбудована у MacOS. Включає першокласну підтримку розробки хмарних, мобільних та веб-застосунків, а також ігор.

Встановлення Visual Studio

Встановлення Visual Studio відбувається шляхом запуску онлайн інсталятора, який знаходиться на веб-ресурсі <https://visualstudio.microsoft.com/>. Для безкоштовного використання в рамках навчального процесу найкраще підходить версія Community Edition.

Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» Жуковський В.В.

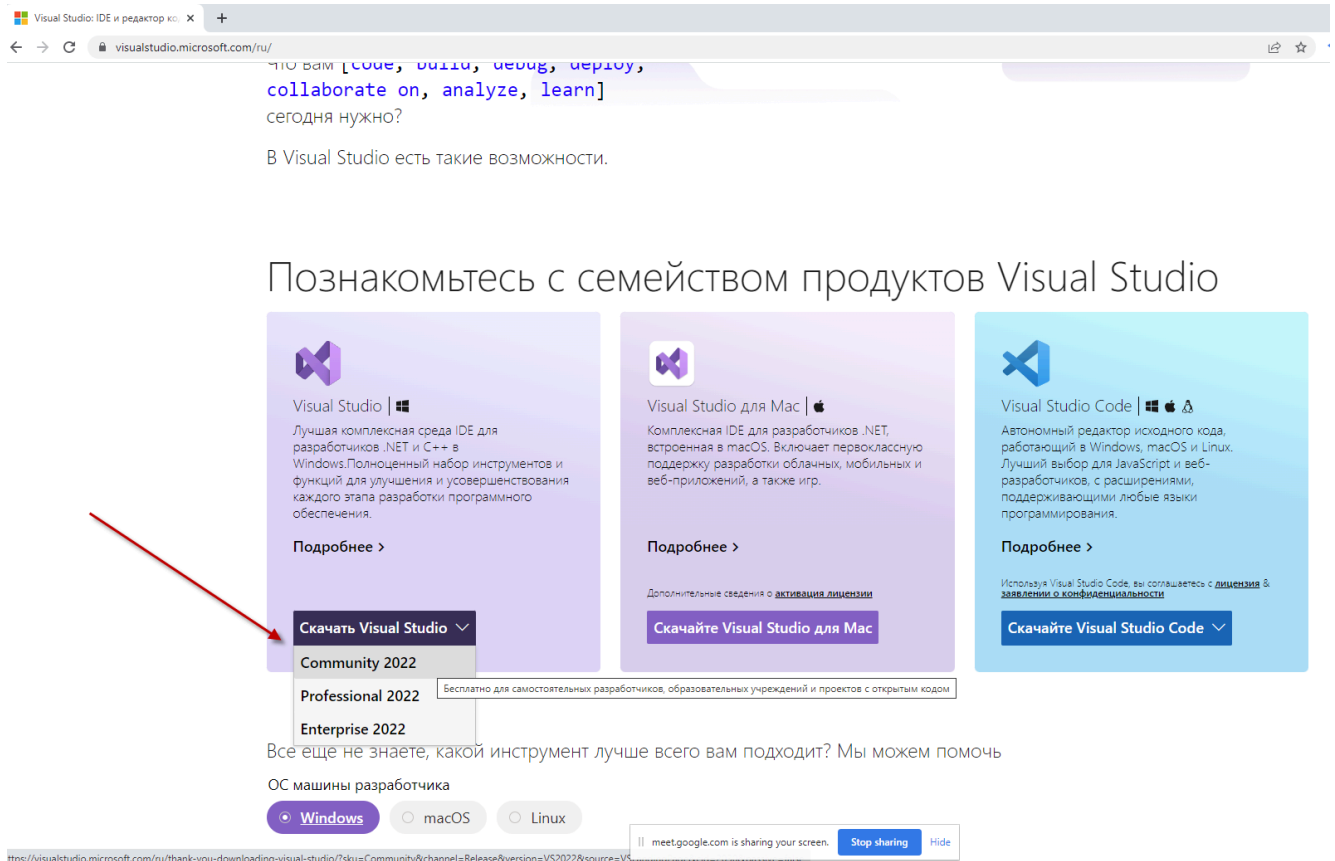
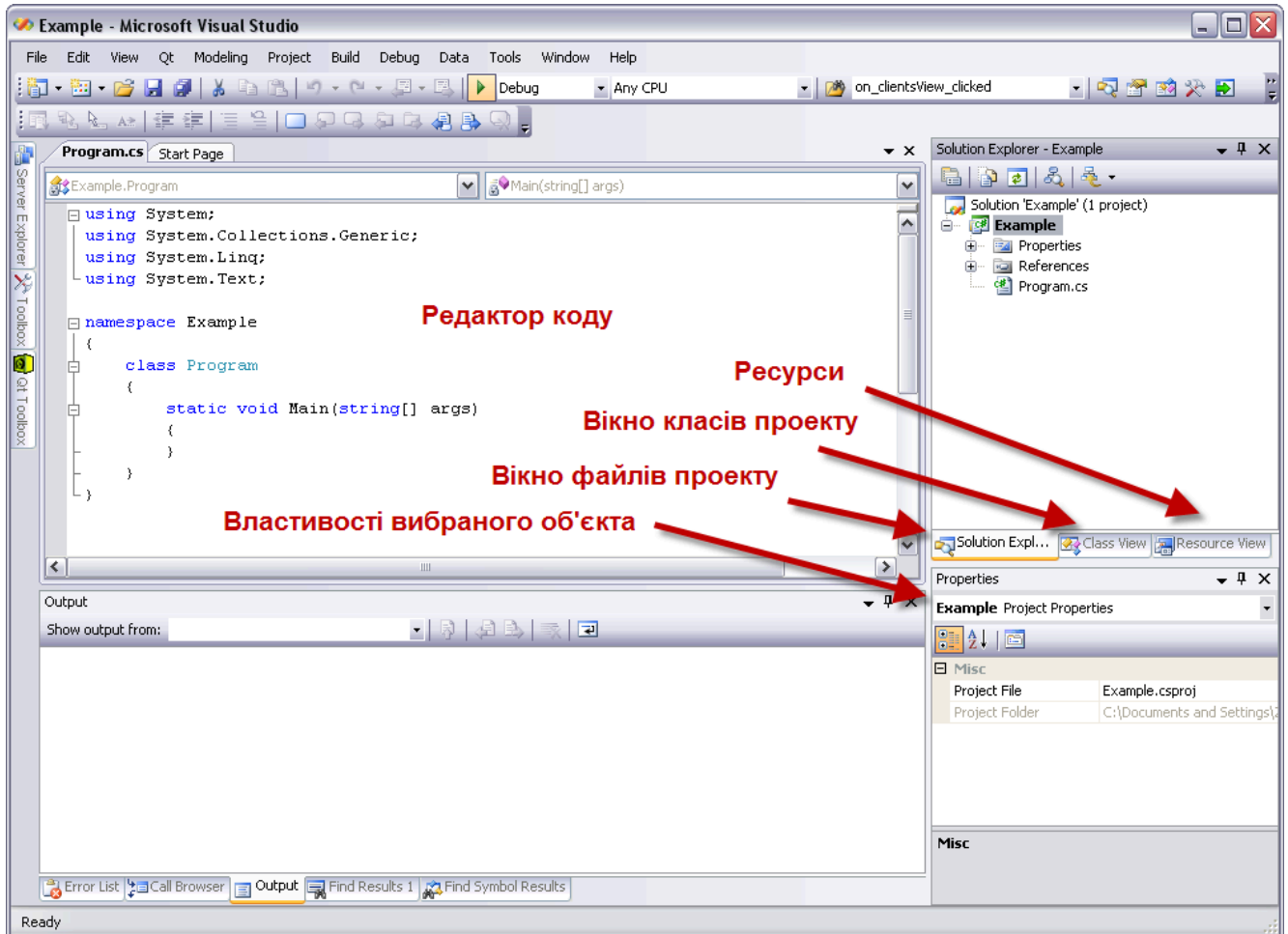


Рис. 1. Процесс выбора онлайн-инсталлятора Visual Studio Community Edition

Створення проекту

Для того щоб створити новий проект виберіть File -> New Project -> Visual C# -> Console Application. Дайте назву вашому проекту, вікно IDE після цього набуде приблизно такого вигляду:

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.



Головне меню:

File Edit View Qt Modeling Project Build Debug Data Tools Window Help

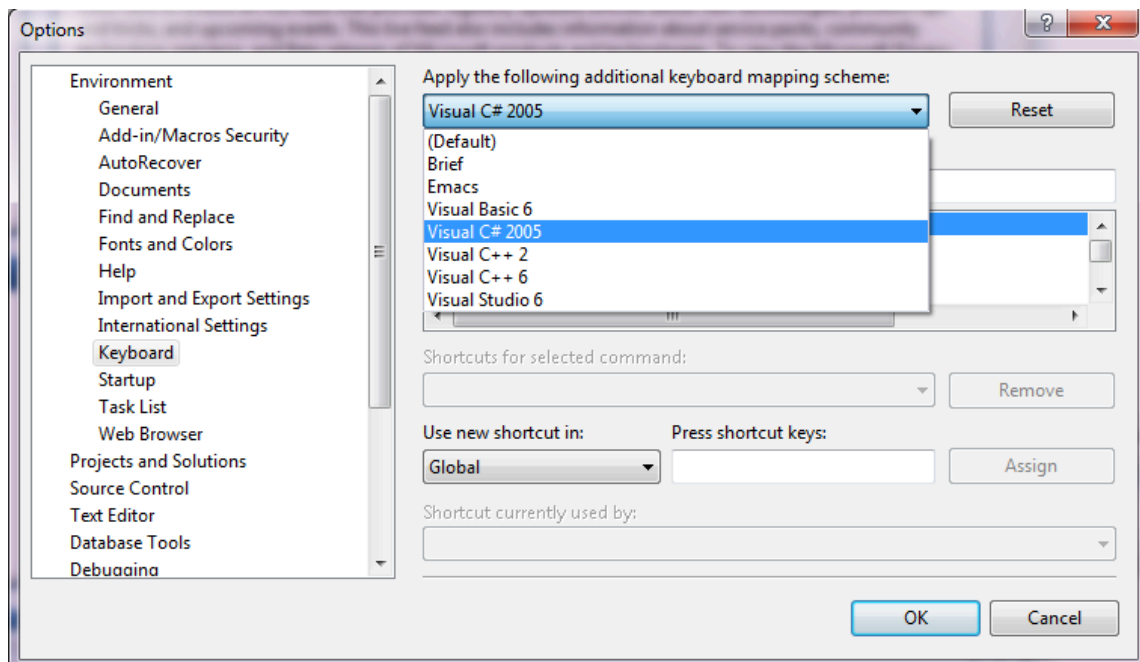
- File - відкриття, створення, додавання, закривання, друк та ін.
- Edit - стандартні команди редагування: копіювання, вставка, вирізання та інші.
- View - команди для приховування та відображення всіх вікон і панелей інструментів.
- Qt – спеціальне меню після встановлення фреймворку Qt
- Modelling - спеціальне меню після встановлення SDE
- Project - команди для роботи з проектом: додавання елементів, форм, посилань та ін.
- Build - команди компіляції програми.
- Debug - команди для відлагодження програми.
- Data - команди для роботи з даними.
- Tools - команди додаткових інструментів і налаштування Visual Studio.NET.
- Window - управління розташуванням вікон.
- Help - довідка.

Visual Studio hotkeys

Можливо дехто вже користується різноманітними плагінами, розширеннями для Visual Studio (наприклад Visual Assist, R#). Варто зауважити, що багато з них досить зручні. Разом з code snippets значно пришвидшують роботу. Тим, хто не знайомий з такими речами варто спробувати і почати використовувати.

А зараз спробуємо розглянути основні гарячі клавіші середовища Visual Studio.

В Visual Studio є декілька схем для розробників, переключати які можна через меню, як показано нижче:



Пошук

Хтось користується пошуком для рефакторингу, хтось для ознайомлення зі своїм же кодом, тому життя можна покращити досить легко:

Ctrl + F - відкривається стандартне вікно пошуку, якщо в буфер нещодавно було скопійовано текст, то він автоматично підставитися.

Ctrl + F3 - пошук «активного» тексту вниз (який виділений або слово на якому стоїть курсор), без копіювання в буфер обміну.

Ctrl + Shift + F3 - пошук «активного» тексту вгору.

Ctrl + I - і починаєте друкувати текст, такий собі runtime пошук.

Intellisense

Вам лінь писати текст повністю?! Швидкість випадуючого Intellisense значно повільніше швидкості вашої думки?! (**IntelliSense** це технологія авто доповнення Microsoft, найбільш відома в Microsoft Visual Studio. Допишує назви функцій при введенні початкових букв. Крім прямого призначення IntelliSense використовується для доступу до документації і уникання двозначностей в іменах змінних, функцій, методів ітд.)

Тоді саме час застосувати наступні комбінації:

Ctrl + Space - виклик Intellisense з показом «слів»

Ctrl + J - якщо ви забули назву методу класу або його властивості, то ця комбінація вам допоможе. Можна застосовувати і попередню, але ця більш «розумна».

Ctrl+Shift+Space – якщо забули список параметрів у метода

Ctrl + K, X - показ всіх сніпетів, які у вас є, у вигляді списку, що випадає, використання інтуїтивно зрозуміле:)

коротке_ім'я_сніпета + Tab - використовувати сніпет в коді. Список базових сніпетів можна подивитися [тут](http://msdn.microsoft.com/en-us/library/z41h7fat(VS.80).aspx) ([http://msdn.microsoft.com/en-us/library/z41h7fat\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/z41h7fat(VS.80).aspx)).

Навігація за кодом

Тут варто сказати окремо і досить багато, навігація дуже зручна, особливо коли звикаєш:)

Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування»

Жуковський В.В.

F12 - перейти до визначення / оголошенню класу / методу / властивості. Дуже практично ви шукайте в коді, «а де ж оголошується ця змінна».

Ctrl + G - перейти на певний номер рядка в коді. (Відображення номерів рядків можна включити: Text Editor -> Мова -> Display -> Line numbers - поставити галку). Іноді досить зручно, особливо якщо отримуєте опис необробленого винятку, а там вказано, як правило, номер рядка.

Ctrl +] - коли курсор встановлено на дужку, багаторядковий коментар (/ * * /), регіон (# region), рядок, обрамлену лапками, ця комбінація дозволить вам побачити початок і закінчення цих «блоків», переміщаючи курсор між ними.

Ctrl + M, M – що не довго тикати мишкою по \pm , повертаючи і розгортаючи «блоки»

Ctrl + M, O - комбінація аналогічна попередньої, тільки згортає всі блоки

Ctrl + - - перехід до попереднього активного тексту (того який ви редагували)

Ctrl + Shift + - - перехід до наступного активного тексту, якщо він звичайно є;

Зверніть увагу що студія, за своєю природою MDI, значить повинна бути можливість переміщатися між вікнами:

Ctrl + Tab - аналог Alt + Tab в Windows, тільки можна застосувати до студії.

Дуже зручним інструментом у студії є **закладки**, так-так, за аналогією з книжковими. Можна відзначити код щоб потім перейти до нього:

Ctrl + B, T - поставити / зняти закладку.

Ctrl + B, Ctrl + C - видалити всі закладки.

Ctrl + B, N - перейти до наступної закладки.

Ctrl + B, P - перейти до попередньої закладки.

Ctrl + \, E - перейти до вікна з повідомленнями про помилки.

Форматування та редагування

Багато знають і використовують **Ctrl + C** і **Ctrl + V**, але студія володіє ще рядом зручних комбінацій:

Ctrl + Shift + V - не відпускаючи **Ctrl + Shift** і, натискаючи на **V**, ви зможете вставити один з 10 останніх шматків тексту, скопійованих в буфер обміну.

Tab - збільшити відступ.

Shift + Tab - зменшити відступ.

Мені не вистачає вертикального виділення тексту. Виявляється і така можливість є! Варто натиснути **Alt**, і не відпускаючи, виділяти мишкою текст.

Якщо код дістався вам «брудний», не тільки за структурою, але і за зовнішнім оформленням, використовуйте:

Ctrl + E + D - форматування всього документа, дозволяє привести документ в більш-менш прийнятний вигляд.

Ctrl + E + F - форматування вибраної частини тексту

Іноді з метою рефакторінгу або ще якихось благих починань, є сенс перевести всі літери у верхній або нижній регістр:

Ctrl + Shift + U - перевести виділений блок у верхній регістр

Ctrl + U - перевести виділений блок в нижній регістр

Приклад простої програми на мові C#

```
using System;

class Example
{
    public static void Main()
    {
        Console.WriteLine("Hello World!");
    }
}
```

Windows-форми

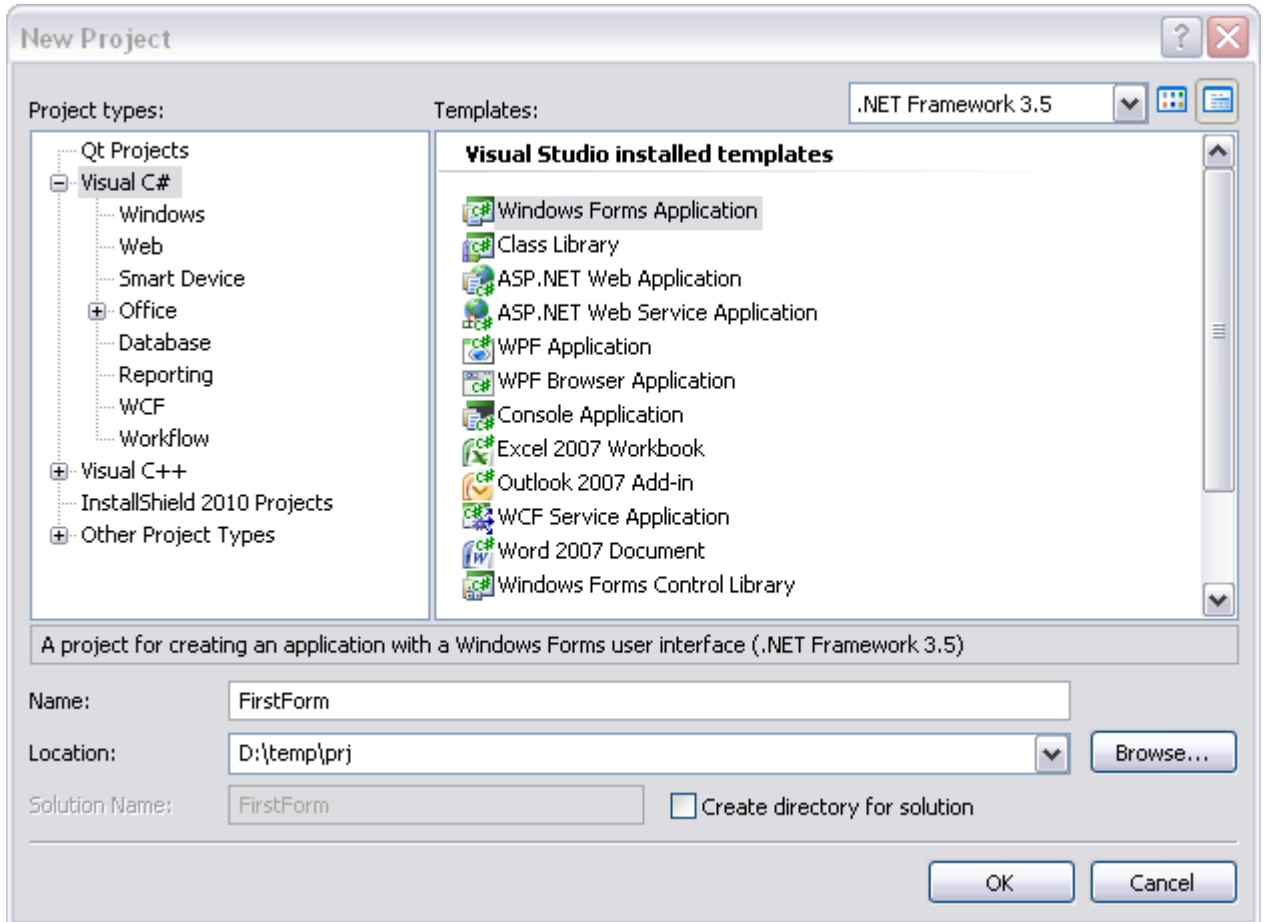
Що таке форма?

Всякий раз, коли ви працюєте з будь-якою програмою, ви бачите на екрані прямокутне вікно. У цьому вікні і розташовується вся інформація, яка вам потрібна. Форми можуть мати стандартний вигляд, такий, наприклад, як у програми Microsoft Word, або нестандартний - як у Winamp. Отже, форма - це екранний об'єкт, що забезпечує функціональність програми.

Ядром Windows-програм, написаних на C #, є форма. Форма інкапсулює основні функції, необхідні для створення вікна, його відображення на екрані та отримання повідомлень. Форма може представляти собою вікно якого типу, включаючи основне вікно програми, дочірнє або навіть діалогове вікно.

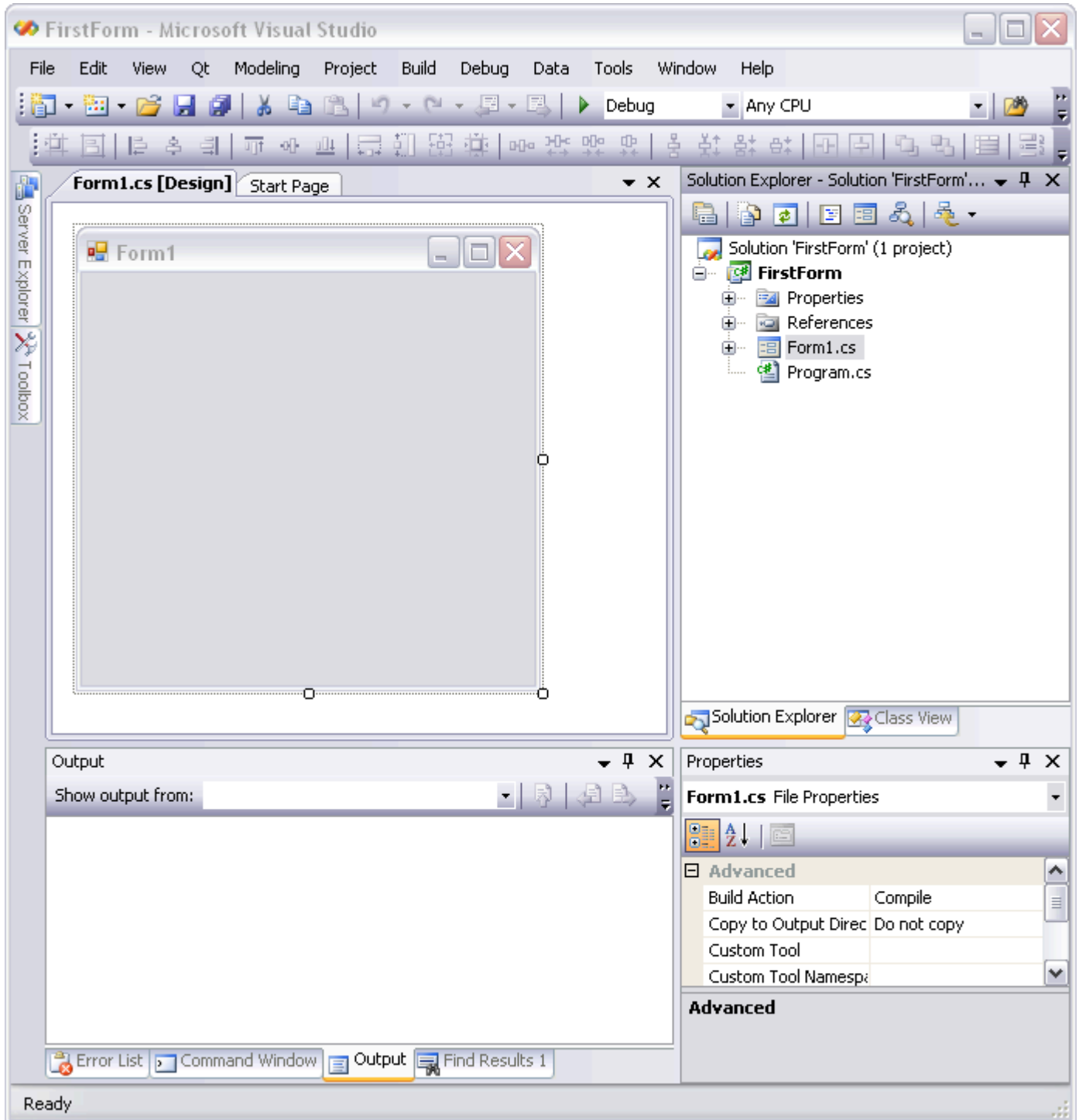
Як створити проект з формою

Запускаємо Visual Studio .NET, вибираємо File / New / Project - з'являється діалогове вікно (Ctrl + Shift + N призводить до того ж результату), в якому вибираємо проект Visual C # та Windows Application:



У полі Ім'я задаємо ім'я проекту - FirstForm і зберігаємо його в папку, яка визначається полем Location. Отриману папку ви зможете згодом перемістити на інший комп'ютер і продовжити роботу - в ній будуть знаходитися всі створені нами файли цього проекту. На екрані з'явилася порожня Windows-форма.

Після того як ми створили новий проект, побачимо основні частини середовища розробки:



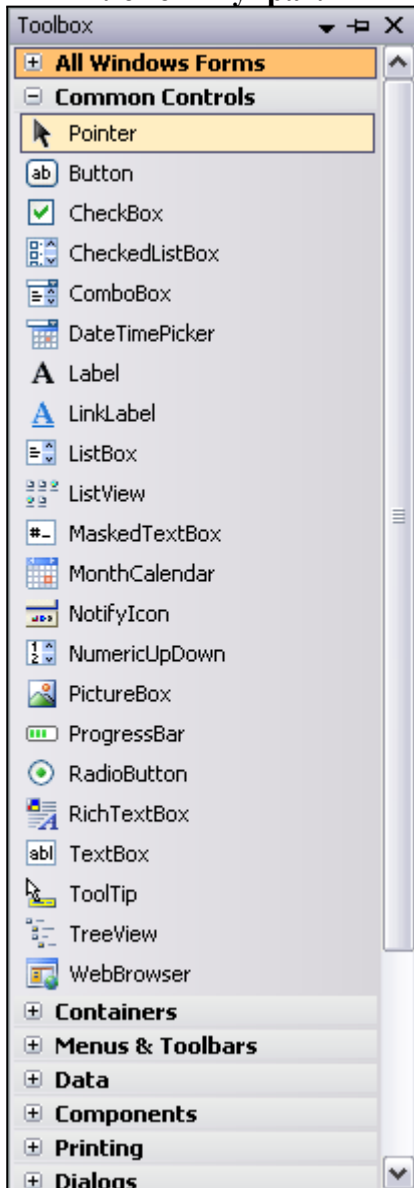
Спочатку вікно створюється порожнім. Потім в нього додаються меню і елементи управління, наприклад кнопки, списки та прапорці. Таким чином, форму можна представити у вигляді контейнера для інших Windows-об'єктів.

Як виконується Windows-проект

За замовчуванням щойностворений проект містить клас Form1 - нащадок класу Form. Цей клас містить точку входу в проект - функцію Main (з цієї функції починається виконання програми). Яка викликає статичний метод Run, який створює об'єкт класу Form1 і відкриває форму - видимий образ об'єкта - для інтерактивної роботи користувача. Все це поки що

здається складним і заплутаним, але потім після того як будуть більш детальніше розглянуті класи і їхня ієрархія все стане на свої місця.

Елементи управління

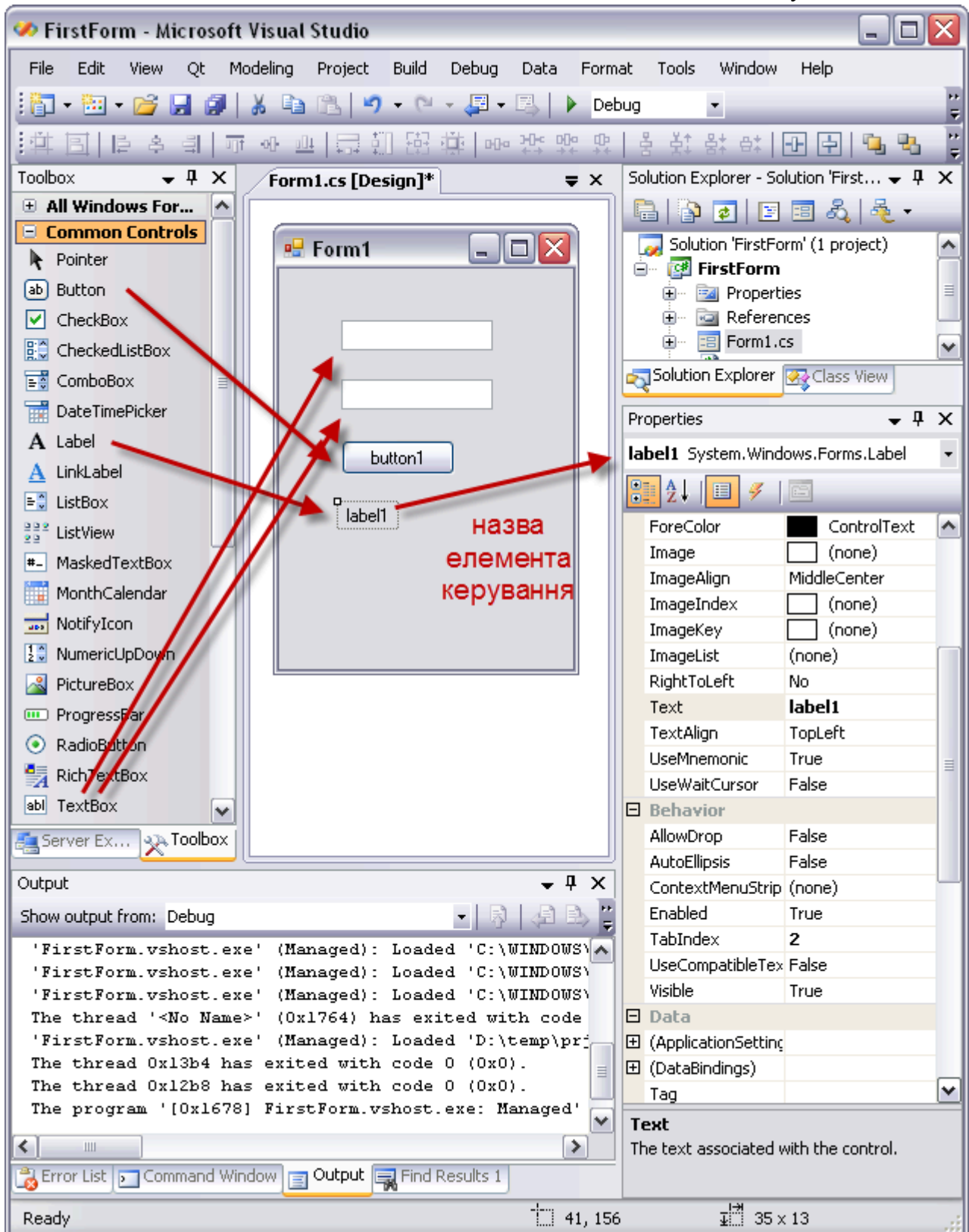


Як додати елементи управління(керування) на форму? Найчастіше, це робиться вручну в режимі проектування. Доступні елементи управління, які відображаються на спеціальній панелі (Toolbox), перетягуються на форму. Цей процес контролюється дизайнером форм (Designer Form). Як тільки на етапі проектування ви перетягуєте на форму елемент керування, негайно в тексті класу з'являються відповідні рядки коду. Звичайно, все можна робити і програмно - поява відповідних рядків коду призводить до появи елементів управління на формі. Потрібно розуміти, що форма - це видимий образ класу форми, а елементи управління, розміщені на формі - це видимі образи клієнтських об'єктів відповідних класів, спадкоємців класу Control. Так що форма з її елементами управління є пряме відображення програмного коду.

Кожен вид елементів управління описується власним класом. Бібліотека FCL (Framework Class Library) містить велику кількість класів, які задають різні елементи управління. Одним з типів проектів, доступних на C #, є проект, який створює елемент управління, так що ніщо не заважає створювати власні елементи управління і розміщувати їх на формах поряд з вбудованими елементами. Багато фірм спеціалізуються на створенні елементів управління - це один з видів повторно використовуваних компонентів.

Створення простого проекту

Розглянемо як створити простий проект, який буде сумувати 2 числа. Для цього перемістимо на форму елементи керування, що будуть відповідати за введення цих чисел (TextBox), кнопку (Button) при нажатті на яку буде відбуватися операція сумування і мітку (Label), що буде виводити результат:



Після цього потрібно дописати обробник натискання на кнопку. Клікнувши по ній двічі переходимо до редактора коду, де напишемо наступний код:

```
private void button1_Click(object sender, EventArgs e)
{
    double d1 = 0;
    double d2 = 0;
```

```
double rez = 0;

d1 = double.Parse(textBox1.Text.ToString());
d2 = double.Parse(textBox2.Text.ToString());
rez = d1 + d2;
label1.Text = rez.ToString();
}
```

Зауважте, що `textBox1`, `textBox2`, `label1` це назви відповідних компонентів, що знаходяться на формі, їх назви можна побачити, а також змінити у вікні властивостей(Properties), яке зазвичай знаходиться справа.

Все, тепер після компіляції програми, ви отримаєте просту програму, що вміє сумувати 2 числа. Хорошою практикою є надання зрозумілих назв для елементів керування, тобто замість `label1` написати `result` ітд.

Хід виконання

1. Ознайомитись з середовищем програмування Microsoft Visual Studio.
2. Створити проект в Microsoft Visual Studio.Net IDE.
3. Написати просту програму.
4. Запустити на виконання.
5. Розробити алгоритм створення найпростішого Windows-додатку в середовищі Visual Studio .NET з використанням форм. Приклад такого створення можна переглянути у відео-презентації <https://youtu.be/kUvKtoU3lt4>
6. Передбачити перевірку введених користувачем даних на коректність (від'ємні значення, букви, недопустими символи, тощо).
7. Продемонструвати результати викладачу.
8. Оформити звіт та захистити лабораторну роботу.

Варіанти завдань для самостійної роботи

1. Написати програму знаходження площі рівностороннього трикутника, якщо відомий його периметр.
2. Написати програму, для обчислення об'єму кулі, якщо дано її радіус.
3. Написати програму для підрахунку швидкості, з якою спортсмен пробігає дистанцію, якщо задано дистанція та час витрачений на біг.
4. Написати програму для визначення оптимальної ваги людини, якщо задано її зріст.
5. Дано три числа a , b , c , які задають сторони рівнобедреного трикутника. Визначити площу даного трикутника.
6. Дано три числа a , b , c , які задають сторони рівнобедреного трикутника (c – основа). Визначити висоту трикутника, опущену на основу c .
7. Написати програму для обчислення площі паралелограма, якщо задано його сторони a , b та гострий кут α .
8. Написати програму для обчислення площі прямокутника, якщо задано його сторони a та b .
9. Відома довжина кола L . Знайти площу круга, обмеженого цим колом $S = \pi R^2$.
10. Написати програму для обчислення вартості поїздки, якщо задано відстань, вартість палива, споживання палива (літрів/100 км).
11. Написати програму, яка б розраховувала вартість купівлі із врахуванням знижки. Вхідними даними мають бути: кількість продукції, вартість за одиницю продукції, знижка.
12. Знаючи площу прямокутного трикутника і один катет a , знайти периметр даного трикутника.

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

13. Дано три числа a, b, c , які задають сторони трикутника. Визначити периметр трикутника.
14. Написати програму для знаходження площі сфери за заданим радіусом R ($S = 4\pi R^2$).
15. Написати програму знаходження об'єму піраміди, якщо задані площа основи $S_{осн}$ та висота піраміди h ($V = \frac{1}{3} S_{осн} h$).
16. Скласти програму знаходження площі прямокутного трикутника, якщо відомі два його катети a та b ($S = \frac{ab}{2}$).
17. Написати програму для знаходження площі рівнобедреного прямокутного трикутника, якщо відома його гіпотенуза c ($S = \frac{c^2}{4}$).
18. Написати програму для знаходження площі ромба за заданими діагоналями l та f ($S = \frac{1}{2} lf$).
19. Написати програму для знаходження площі трапеції за заданими основами a, b та висотою h ($S = \frac{a+b}{2} h$).
20. Написати програму для знаходження площі круга діаметром d ($S = \pi r^2$, де r – радіус круга).
21. Капітал в сумі 1000 грн. внесений на депозит до банку під 4,5 процентів річних. Написати програму, яка б розраховувала накопичення капіталу за 7 років. ($K_n = K_n \left(1 + \frac{p}{100}\right)^n$).
22. Написати програму для знаходження катету b прямокутного трикутника, якщо відомий інший катет a та гіпотенуза c . Використати теорему Піфагора $c^2 = a^2 + b^2$.
23. Дано три числа a, b, c , що задають сторони трикутника. Написати програму для визначення площі трикутника використовуючи формулу Герона $S = \sqrt{p(p-a)(p-b)(p-c)}$, де p – півпериметр трикутника.
24. Написати програму, яка б дозволяла проводити операції над двома числами (міні калькулятор): додавання, віднімання, множення, ділення із використанням допоміжних змінних.

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Що таке IDE?
2. Що таке форма?
3. Які IDE ви знаєте?
4. Опишіть Microsoft Visual Studio IDE?
5. Що таке hotkeys?

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

6. Які hotkeys ви знаєте?
7. Як швидко запустити програму на виконання?
8. Як здійснити пошук в тексті програми?
9. Як подивитися де оголошена певна змінна?
10. Що таке сніпет (snippet)?
11. Як створити проект з формою?
12. Які елементи керування ви знаєте?
13. Як додати елементи керування на форму?
14. Де подивитися назву елемента керування?

Лабораторна робота №2

Назва роботи: Клас та об'єкт

Мета роботи: Ознайомитися з особливостями ООП, навчитися створювати класи та об'єкти

Теоретичні відомості

Об'єктно-орієнтоване програмування (ООП) - це новий підхід до програмування. По мірі розвитку обчислювальної техніки і ускладнення задач виникли різні моделі програмування. Перші компілятори підтримували процедурну модель програмування, в основі якої лежить використання функцій. Використовуючи процедурну модель програмування, програмісти могли писати програми до декількох тисяч рядків довжиною. Наступний етап розвитку зв'язаний з переходом до структурної моделі програмування. Зміст структурного програмування полягає в представленні програми у вигляді сукупності взаємопов'язаних процедур і тих даних, якими ці процедури оперують. При цьому широко використовуються програмні блоки і допускається тільки мінімальне використання оператора GOTO. Супровід таких програм простіший, чим процедурно-орієнтованих. Використовуючи структурне програмування, середній програміст може створювати і супроводити програми до декількох десятків тисяч рядків.

Для написання більш складних задач знадобиться новий підхід до програмування, який і був реалізований в моделі ООП. Модель ООП основана на декількох основоположних концепціях.

Абстракція даних - це можливість визначати новий тип даних, з яким можна працювати майже так, як і з основними типами даних. Такі типи часто називаються абстрактними типами даних, хоча термін типи даних, визначені користувачем, є більш точним.

Інкапсуляція - це механізм, який об'єднує дані і код, який маніпулює цими даними, а також захищає ті і інші від стороннього втручання.

Для реалізації цих двох основоположних концепцій в мові C# використовуються класи. Термін клас визначає тип об'єктів. При цьому кожний представник класу називається об'єктом. Кожний об'єкт завжди має свій, унікальний стан, визначений поточним значенням його даних-членів. Функціональне призначення класу визначається можливостями дії над об'єктами класу, які задаються його функціями-членами (методами). В кожному класі розподіляється пам'ять для зберігання даних, і встановлюються допустимі операції для кожного об'єкту даних даного типу. Створення об'єктів даного класу здійснюється спеціальною функцією-членом, яка називається конструктором, а знищення - іншою спеціальною функцією-членом, яка називається деструктором. Клас дозволяє робити недоступними внутрішні дані, представляючи їх як відкриті (public), закриті (private) і захищені (protected). Клас встановлює чітко визначений інтерфейс для взаємодії об'єктів цього типу з решта світом. Отриманими об'єктами можна управляти за допомогою повідомлень (запитів), які представляють собою просто виклики функцій-членів.

Поняття класу є найважливішим в мові C#. Синтаксис опису класу схожий на синтаксис опису структури:

```
public class Customer
{
    //Далі тут описуються поля, властивості, методи ітд...
}
```

Ключовому слову class передуює рівень доступу. Оскільки в даному випадку використовується public, то будь-хто може створювати об'єкти цього класу. Ім'я класу

вказується після ключового слова `class`. Частина визначення є тілом класу, в якому задаються дані і поведінка. Поля, властивості, методи і події в класі позначаються терміном члени класу.

Створення об'єктів

Клас і об'єкт - це різні речі, хоча в деяких випадках вони взаємозамінні. Клас визначає тип об'єкта, але не сам об'єкт. Об'єкт - це конкретна сутність, заснована на класі і таким може називатися екземпляром класу.

Об'єкти можна створювати за допомогою ключового слова `new`, за яким написано ім'я класу:

```
Customer object1 = new Customer();
```

При створенні екземпляра класу посилання на цей об'єкт передається програмісту. У попередньому прикладі `object1` є посиланням на об'єкт, класу `Customer`. Це посилання вказує на новий об'єкт, але не містить дані цього об'єкта. Фактично, можна створити посилання на об'єкт без створення самого об'єкту:

```
Customer object2;
```

Створення таких посилань, які не вказують на об'єкт, не рекомендується, так як спроба доступу до об'єкта по такому посиланні призведе до збою під час виконання. Проте таке посилання можна використати виконуючи присвоєння вже існуючого екземпляра класу:

```
Customer object3 = new Customer();  
Customer object4 = object3;
```

В даному коді створюються два посилання на об'єкти, які вказують на один об'єкт. Тому будь-які зміни об'єкта, виконані за допомогою `object3`, будуть видні при подальшому використанні `object4`.

Коли створюється функція для взяття значення деякого поля, то прийнято використовувати англійське слово `get`(взяти) разом з назвою поля, а для зміни слово `set` (дати). В якості формального параметра для функція зміни поля краще використовувати ту ж назву поля, але з малої букви.

Приклад створення об'єкту певного класу

```
// на мові C#  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
  
namespace ConsoleApplication1  
{  
    public class Person  
    {  
        // Поле  
        public string name;  
  
        // Конструктор  
        public Person()  
        {  
            name = "unknown";  
        }  
  
        // Метод  
        public void SetName(string newName)  
        {  
            name = newName;  
        }  
    }  
}
```

```
    }  
}  
class TestPerson  
{  
    static void Main()  
    {  
        Person person = new Person();  
        Console.WriteLine(person.name);  
  
        person.SetName("Петро Іванчук");  
        Console.WriteLine(person.name);  
  
        // Чекає на нажаття будь-якої клавіші.  
        Console.WriteLine("Press any key to exit.");  
        Console.ReadKey();  
    }  
}
```

Код, що наведений вище є працездатним, але водночас не є коректним з точки зору дотримання “хорошого” тону розробки. Таким чином варто його відкоригувати і реалізувати механізм інкапсуляції (приховання публічних методів).

Хід виконання

1. Розробити алгоритми використання класів згідно варіантів завдань для самостійної роботи.
2. Написати відповідну програму на мові C#. Програма повинна мати як мінімум 2 методи: введення даних та обчислення.
3. Перевірити код на дотримання Code Convention C#:
(<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions> ,
<https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/identifier-names>)
4. Перевірити на дотримання принципу інкапсуляції.
5. Завантажити код програми на GitHub у репозиторій заданий викладачем.
6. Захистити лабораторну роботу.

Варіанти завдань для самостійної роботи

1. Створити об’єкт класу „арифметична прогресія”. Клас характеризується першим членом a_0 , різницею d та кількістю членів n . Визначити за допомогою методу класу, суму прогресії.
2. Створити об’єкт класу „трикутник на площині”, який заданий координатами своїх вершин. Визначити за допомогою методу класу його, площу.
3. Створити об’єкт класу „опуклий чотирикутник на площині” (чотирикутник заданий координатами вершин). Визначити за допомогою методу класу його, периметр.
4. Створити об’єкт класу „круг на площині”, який заданий радіусом та координатами центру. Визначити за допомогою методу класу його, площу.
5. Створити об’єкт класу „тривимірний вектор”. Визначити за допомогою методу класу його, довжину.
6. Створити об’єкт класу „функція $\sin(ax+b)$ ”. Визначити за допомогою методу класу, чи приймає вона найбільше значення у введеній користувачем точці x .

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

7. Створити об'єкт класу „слово”. Визначити за допомогою методу класу, кількість цифр в ньому.
8. Створити об'єкт класу „кубічний многочлен $a_3x^3 + a_2x^2 + a_1x + a_0$ ”. Визначити за допомогою методу класу, найменше значення на введеному користувачем відрізку $[a, b]$ із заданою точністю ε (використати метод простого перебору).
9. Створити об'єкт класу „площина $Ax + By + Cz + D = 0$ ”. Визначити за допомогою методу класу, чи належить введена користувачем точка, цій площині.
10. Створити об'єкт класу „тетраedr”, який заданий координатами своїх вершин в просторі. Визначити за допомогою методу класу його об'єм.
11. Створити об'єкт класу „конус”, який заданий координатами центру основи, координатами вершини та радіусом основи. Визначити за допомогою методу класу, його твірну.
12. Створити об'єкт класу „людина”. Визначити за допомогою методу класу, чи народилася вона в щасливі дні (ділення суми цифр числа, місяця, року народження на 7 дають три однакові остачі).
13. Створити об'єкт класу „пряма $ax + by + c = 0$ ”. Визначити за допомогою методу класу, чи належить одна із двох введених користувачем точок.
14. Створити об'єкт класу „точка в просторі”. Визначити за допомогою методу класу, чи всі координати цілочислові і які попадають в перший октант.
15. Створити об'єкт класу „квадратне рівняння $ax^2 + bx + c = 0$ ”. Визначити за допомогою методу класу, чи воно має дійсні розв'язки.
16. Створити об'єкт класу „абонент”. Визначити за допомогою методу класу, суму цифр телефону.
17. Створити об'єкт класу „куля”. Куля задається радіусом та координатами центру. Визначити за допомогою методу класу, чи попадає в неї, введена користувачем точка.
18. Створити об'єкт класу „числова послідовність” (містить ≤ 5 членів). Визначити за допомогою методу класу, найменше середнє квадратичне відхилення між послідовними членами.
19. Створити об'єкт класу „перетворення площини $\begin{cases} x = a_1\xi + b_1\eta + c_1, \\ y = a_2\xi + b_2\eta + c_2; \end{cases}$ ”. Визначити за допомогою методу класу, чи переводять задані користувачем точки (x, y) та (ξ, η) одна в одну.
20. Створити об'єкт класу „геометрична прогресія”. Клас характеризується першим членом a_0 , знаменником q та кількістю членів n . Визначити за допомогою методу класу, чи є в ній найбільший останній член.
21. Створити об'єкт класу „сума $\sum_{i=1}^3 \frac{a_i}{i(i+1)}$ ”. Визначити за допомогою методу класу суму.
22. Створити об'єкт класу „матеріальна точка”. Точка характеризується координатами початкового положення та вектором рівномірної швидкості $\vec{v} = (v_1, v_2, v_3)$. Визначити за допомогою методу класу, чи за введений користувачем час t попадають в перший октант.

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.**

23. Створити об'єкт класу „квадратне рівняння $ax^2 + bx + c = 0$ ”. Визначити за допомогою методу класу, чи воно має дійсні розв'язки.
24. Створити об'єкт класу „арифметична прогресія”. Клас характеризується першим членом a_0 , різницею d та кількістю членів n . Визначити за допомогою методу класу, суму прогресії.
25. Створити об'єкт класу „функція $\sin(ax+b)$ ”. Визначити за допомогою методу класу, чи приймає вона найбільше значення у введеній користувачем точці x .

Зміст звіту:

Назва та мета роботи
Завдання.
Текст програми.
Висновки

Контрольні запитання

1. Чим відрізняється поняття „клас” від поняття „об'єкт”?
2. Які ключові слова використовуються для описання класів в C#?
3. Які специфікатори доступу використовуються в класах?
4. Чи є конструктор класу методом даного класу?
5. Які основні призначення конструкторів та деструкторів класу?

Лабораторна робота №3

Назва роботи: Масиви об'єктів

Мета роботи: Ознайомитися з особливостями ООП, навчитися створювати масиви об'єктів

Теоретичні відомості

Коли створюється функція для взяття значення деякого поля, то прийнято використовувати англійське слово `get`(взяти) разом з назвою поля, а для зміни слово `set` (дати). В якості формального параметра для функція зміни поля краще використовувати ту ж назву поля, але з малої букви.

Як проводити дослідження використання пам'яті? Одна з підказок - запитати про можливі варіанти дослідження в ChatGPT. Приклад комунікації:

Запитання (важливо чітко сформулювати завдання для chatGPT):

Існує завдання дослідити роботу деструктора при створенні великої кількості об'єктів. Постає питання як описати використання пам'яті і автоматичний виклик `garbage collector`-а після очищення. Є програма на C#, є інструменти `task manager` і `diagnostic tool visual studio`. Які аспекти можна використати в цій оцінці?

Відповідь:

Дослідження роботи деструктора і автоматичного сборщика сміття (`garbage collector`) при створенні великої кількості об'єктів в C# може бути корисним для оцінки продуктивності вашої програми та виявлення можливих утечек пам'яті. Щоб виконати таке дослідження, ось кілька кроків і аспектів, які ви можете використовувати:

1. Створення великої кількості об'єктів:

Створіть код, який створює значну кількість об'єктів в вашій програмі. Наприклад, можна використовувати цикли для створення багатьох об'єктів класу.

2. Використання деструкторів:

Додайте деструктори (`finalizers`) до класів об'єктів, які ви створюєте. Деструктори викликаються автоматично під час сборки сміття і можуть бути використані для вивчення, коли об'єкти очищаються.

3. Відстеження використання пам'яті:

Використовуйте інструменти для відстеження використання пам'яті, такі як `Task Manager` або профайлери від `Visual Studio`. Спостерігайте за зростанням використання пам'яті під час створення об'єктів.

4. Аналіз автоматичної сборки сміття:

Спостерігайте за роботою автоматичного сборщика сміття. Велика кількість створених об'єктів може призвести до частіших запусків сборки сміття. Використовуйте інструменти Visual Studio, такі як Diagnostic Tools, для аналізу роботи сборки сміття.

5. Вимір впливу на продуктивність:

Вимірюйте час, необхідний для створення та очищення великої кількості об'єктів. Порівнюйте цей час з величиною використаної пам'яті і роботою сборщика сміття.

6. Пошук утечек пам'яті:

Використовуйте інструменти для пошуку утечек пам'яті, такі як утиліта .NET Memory Profiler, для виявлення можливих проблем з утечками пам'яті після завершення роботи програми.

Ці аспекти допоможуть вам оцінити вплив створення великої кількості об'єктів на роботу вашої програми і виявити проблеми, пов'язані з використанням пам'яті та сборкою сміття в C#.

Приклад створення масиву об'єктів

```
// на мові C#
using System;
using System.Collections.Generic;
using System.Text;

namespace lab4
{
    class Vector
    {
        double x;
        double y;
        double z;
        public Vector(double x, double y, double z)
        {
            this.x = x;
            this.y = y;
            this.z = z;
        }
        public void InputCoordinates(double vx, double vy, double vz)
        {
            x = vx;
            y = vy;
            z = vz;
        }
        public double GetLength()
        {
            double l = Math.Sqrt(Math.Pow(x, 2) + Math.Pow(y, 2) + Math.Pow(z, 2));
            return l;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Vector[] v = new Vector[100];
            double x = 0, y = 0, z = 0;
        }
    }
}
```

```
double vx, vy, vz;
int count;
int k = 0;
Console.WriteLine("Please, input vector count:");
count = Convert.ToInt32(Console.ReadLine());
for (int i = 0; i < count; i++)
{
    v[i] = new Vector(x, y, z);
}
for (int i = 0; i < count; i++)
{
    Console.WriteLine("Input coordinates for vector#" + (i + 1));
    Console.WriteLine("input x:");
    vx = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("input y:");
    vy = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("input z:");
    vz = Convert.ToInt32(Console.ReadLine());
    v[i].InputCoordinates(vx, vy, vz);
}

double max = v[0].GetLength();
for (int i = 0; i < count; i++)
{
    if (max < v[i].GetLength())
    {
        max = v[i].GetLength();
        k = i;
    }
}
Console.WriteLine("Vector with max length - " + (k + 1));
}
```

Хід виконання

1. Розробити алгоритми використання масиву об'єктів згідно варіантів завдань для самостійної роботи.
2. Написати відповідну програму на мові C#, що містить конструктор та деструктор.
3. Продемонструвати роботу Garbage Collection.
4. Захистити лабораторну роботу.

Варіанти завдань для самостійної роботи

1. Створити масив об'єктів класу „квадратна матриця 3×3 ”. Серед n об'єктів визначити номер матриці з найбільшою сумою діагональних елементів (найбільшим слідом).
2. Створити масив об'єктів „трикутник на площині”, який заданий координатами своїх вершин. Серед n трикутників визначити той, в якого найбільша площа.
3. Створити масив об'єктів „опуклий чотирикутник на площині” (чотирикутник заданий координатами вершин). Визначити той чотирикутник, у якого найбільший периметр.

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

4. Створити масив об'єктів „круг на площині”, який заданий радіусом та координатами центру. Визначити той круг, у якого найбільша площа.
5. Створити масив об'єктів „тривимірний вектор”. Визначити той вектор, який має найбільшу довжину.
6. Створити масив об'єктів „масив з 5-ти елементів”. Визначити той масив, в якого найменший максимальний елемент.
7. Створити масив об'єктів „слово”. Визначити те слово, в якому найбільша кількість цифр.
8. Створити масив об'єктів „кубічний многочлен $a_3x^3 + a_2x^2 + a_1x + a_0$ ”. Визначити той многочлен, який приймає найменше мінімальне значення на введеному користувачем відрізку $[a, b]$ із заданою точністю ε (використати метод простого перебору).
9. Створити масив об'єктів „площина $Ax + By + Cz + D = 0$ ”. Визначити площини, яким належить введена користувачем точка.
10. Створити масив об'єктів „тетраедр”, який заданий координатами своїх вершин в просторі. Із заданого масиву об'єктів „тетраедр” визначити тетраедр з найбільшим об'ємом.
11. Створити клас „конус”, який заданий координатами центру основи, координатами вершини та радіусом основи. Із заданого масиву об'єктів вказаного класу визначити конус з найбільшою твірною.
12. Створити клас „людина”. Із заданого масиву людей визначити тих, які народилися в щасливі дні (ділення суми цифр числа, місяця, року народження на 7 дають три однакові остачі).
13. Створити масив об'єктів класу „пряма $ax + by + c = 0$ ”. Визначити номери тих прямих, яким належить одна із двох введених користувачем точок.
14. Створити масив об'єктів класу „точка в просторі”. Визначити кількість точок, в яких всі координати цілочислові і які попадають в перший октант.
15. Створити масив об'єктів класу „тривимірний масив”. Визначити номер об'єкта, в якого найбільший мінімальний елемент.
16. Створити масив об'єктів класу „абонент”. Визначити того абонента, в якого найбільша сума цифр телефону.
17. Створити масив об'єктів класу „куля”. Куля задається радіусом та координатами центру. Визначити номери тих куль, в які попадає введена користувачем точка.
18. Створити масив об'єктів класу „числова послідовність” (містить ≤ 5 членів). Визначити ту послідовність, в якій найменше середнє квадратичне відхилення між послідовними членами.
19. Створити масив об'єктів класу „перетворення площини $\begin{cases} x = a_1\xi + b_1\eta + c_1, \\ y = a_2\xi + b_2\eta + c_2; \end{cases}$ ”. Визначити номери тих перетворень, які переводять задані користувачем точки (x, y) та (ξ, η) одна в одну.
20. Створити масив об'єктів класу „геометрична прогресія”. Клас характеризується першим членом a_0 , знаменником q та кількістю членів n . Визначити номер тієї прогресії, в якій буде найбільший останній член.
21. Створити масив об'єктів класу „сума $\sum_{i=1}^3 \frac{a_i}{i(i+1)}$ ”. Визначити номер об'єкта з найбільшою сумою.

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

22. Створити масив об'єктів класу „матеріальна точка”. Кожна точка характеризується координатами початкового положення та вектором рівномірної швидкості $\vec{v} = (v_1, v_2, v_3)$. Визначити номери точок, які за введений користувачем час t попадають в перший октант.
23. Створити масив об'єктів класу „квадратне рівняння $ax^2 + bx + c = 0$ ”. Визначити номери тих рівнянь, які не мають дійсних розв'язків.
24. Створити масив об'єктів класу „арифметична прогресія”. Клас характеризується першим членом a_0 , різницею d та кількістю членів n . Визначити номер тієї прогресії, в якій буде найбільша сума.
25. Створити масив об'єктів класу „функція $\sin(ax+b)$ ”. Визначити номер функції, яка приймає найбільше значення у введеній користувачем точці x .

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Чим відрізняється поняття „клас” від поняття „об'єкт”?
2. Які ключові слова використовуються для описання класів в C#?
3. Які специфікатори доступу використовуються в класах?
4. Чи є конструктор класу методом даного класу?
5. Які основні призначення конструкторів та деструкторів класу?

Лабораторна робота №4

Назва роботи: Наслідування

Мета роботи: Ознайомитися з особливостями ООП. Навчися використовувати ідеологію наслідування.

Теоретичні відомості

Мова C# дозволяє наслідувати дані-члени (поля) і методи одного класу. При цьому новий клас називається похідним (дочірнім) класом. Клас, елементи якого наслідуються, називається базовим (батьківським) класом для свого похідного класу. Наслідування дає можливість деякі спільні риси поведінки класів абстрагувати в одному класі. Похідні класи, наслідуючи цю спільну поведінку, можуть її дещо змінити, перевизначаючи деякі методи базового класу, чи доповнити, вводячи нові дані-члени (властивості) і функції-члени (методи). Таким чином, визначення похідного класу значно скорочується, оскільки потрібно визначити тільки відмінні його риси від базового класу.

Приклад коду:

```
using System;

public class ParentClass
{
    public ParentClass()
    {
        Console.WriteLine("Parent Constructor.");
    }

    public void print()
    {
        Console.WriteLine("I'm a Parent Class.");
    }
}

public class ChildClass : ParentClass
{
    public ChildClass()
    {
        Console.WriteLine("Child Constructor.");
    }

    public static void Main()
    {
        ChildClass child = new ChildClass();

        child.print();
    }
}
```

Результат:

```
Parent Constructor.
Child Constructor.
I'm a Parent Class.
```

Методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування»

Жуковський В.В.

Специфікатори доступу - це `public`, `protected`, `private`; він не є обов'язковим і по-замовчуванню приймає значення `private`. Специфікатор доступу при наслідуванні визначає рівень доступу до елементів базового класу, який отримує елементи похідного класу.

При будь-якому способі наслідування в похідному класі доступні тільки відкриті (`public`) і захищені (`protected`) члени базового класу. Інакше кажучи, закриті члени базового класу залишаються закритими, незалежно від того, як цей клас унаслідуються.

Як ми раніше вже відмічали, не всі методи класу унаслідуються. Перерахуємо їх:

- конструктори;
- деструктори;
- оператори присвоєння, визначені програмістом;
- друзі класу.

Якщо представлений програмістом конструктор не має параметрів, тобто є конструктором по-замовчуванню, то при створенні екземпляру похідного класу автоматично викликається конструктор базового класу. Після того як об'єкт створений, конструктор базового класу стає недоступним.

Хоча конструктор базового класу і унаслідуються, викликається він тільки компілятором, коли конструюється об'єкт похідного класу. Конструктор, на відмінну від інших успадкованих функцій, викликати явно неможливо.

Щодо деструкторів похідних класів також існують деякі правила. Деструктор похідного класу повинен виконуватися раніше деструктора базового класу. Коли деструктор похідного класу виконає свою частину роботи по знищенню об'єкту, викликається деструктор базового класу. Причому вся робота по організації відповідного виклику накладається на компілятор, програміст не повинен турбуватися про це.

Хід виконання

- Розробити алгоритми використання механізму наслідування, описавши батьківський та похідний класи і перевантажуючи відповідні методи.
- Написати відповідну програму на мові C#.
- Захистити лабораторну роботу.

Варіанти завдань для самостійної роботи:

1. Створити клас „Трикутник”.

Створити відповідні методи:

- задання координат вершин;
- виведення координат вершин на екран;
- обчислення площі.

Описати похідний від нього клас „Опуклий чотирикутник” з відповідними перевантаженими методами:

- задання координат вершин;
- виведення координат вершин на екран;
- обчислення площі.

Створити об'єкти „трикутник”, „опуклий чотирикутник” та обчислити їх площі.

2. Створити клас „одновимірний вектор розмірності 4”.

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

Створити відповідні методи:

- задання елементів вектора;
- виведення вектора на екран;
- знаходження максимального елемента вектора.

Описати похідний від нього клас „матриця” розмірності 4×4 з відповідними перевантаженими методами:

- задання елементів матриці;
- виведення матриці на екран;
- знаходження максимального елемента матриці.

Створити об'єкти класів „одновимірний вектор” та „матриця”. Знайти максимальні елементи кожного об'єкта.

3. Описати клас „дробово-лінійна функція” виду $\frac{a_1x + a_0}{b_1x + b_0}$

Створити відповідні методи:

- задання коефіцієнтів чисельника та знаменника;
- виведення коефіцієнтів на екран;
- знаходження значення в заданій точці x_0 .

Створити похідний від нього клас „дробова функція” $\frac{a_2x^2 + a_1x + a_0}{b_2x^2 + b_1x + b_0}$ з відповідними перевантаженими методами:

- задання коефіцієнтів чисельника та знаменника;
- виведення коефіцієнтів на екран;
- знаходження значення в заданій точці x_0 .

Створити об'єкти класів „дробово-лінійна функція” та „дробова” та обчислити їх значення у відповідній точці.

4. Описати клас „пряма виду $a_1x + a_2y + a_0 = 0$ ” (тут a_0, a_1, a_2 – задані числа).

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить задана точка прямій.

Створити похідний клас „гіперплощина виду $a_4x_4 + a_3x_3 + a_2x_2 + a_1x_1 + a_0 = 0$ ” (тут a_0, a_1, a_2, a_3, a_4 – задані числа, $X = (x_1, x_2, x_3, x_4)$ – точка 4-вимірного простору).

Перевантажити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить задана точка гіперплощині.

Створити об'єкти класів „пряма” та „гіперплощина”. Визначити, чи належать введені користувачем точки створеним об'єктам.

5. Створити клас „куля $(x_1 - b_1)^2 + (x_2 - b_2)^2 + (x_3 - b_3)^2 = R^2$ ”.

Описати методи:

- задання відповідних коефіцієнтів;

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

- виведення відповідних коефіцієнтів на екран;
- визначення об'єму кулі.

Створити похідний клас „еліпсоїд” $\frac{(x_1 - b_1)^2}{a_1^2} + \frac{(x_2 - b_2)^2}{a_2^2} + \frac{(x_3 - b_3)^2}{a_3^2} = 1$ ”, де $a_i, b_i, i = \overline{1,3}$ – задані числа. Перевантажити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення об'єму еліпсоїда.

Створити об'єкти класів „куля” та „еліпсоїд”. Визначити об'єми кулі та еліпсоїда.

6. Створити клас „двовимірна матриця” $A[3][3]$ ”.

Створити методи:

- задання елементів матриці, використовуючи введення з клавіатури;
- задання елементів матриці використовуючи випадкові числа;
- знаходження мінімального елемента матриці.

Створити похідний клас „тривимірна матриця” $A[3][3][3]$ ”. Перевантажити відповідні методи:

- задання елементів матриці, використовуючи введення з клавіатури;
- задання елементів матриці використовуючи випадкові числа;
- знаходження мінімального елемента матриці.

Створити об'єкти класів „двовимірна матриця” та „тривимірна матриця”. Визначити мінімальні елементи всіх матриць.

7. Створити клас „півплощина” $a_1x_1 + a_2x_2 \leq b$ ”.

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить введена користувачем точка $X = (x_1, x_2)$ даній півплощині.

Створити похідний клас „півпростір” $a_1x_1 + a_2x_2 + a_3x_3 \leq b$ ”, де a_1, a_2, a_3, b – задані числа, $X = (x_1, x_2, x_3)$ – точка простору. Перевантажити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення, чи належить введена користувачем точка $X = (x_1, x_2, x_3)$ даному півпростору.

Створити об'єкти класів „півпростір” та „півплощина”. Визначити, чи належать введені користувачем точки півпростору та півплощині.

8. Створити клас „прямокутник” $b_1 \leq x_1 \leq a_1, b_2 \leq x_2 \leq a_2$ ”.

Описати методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення чи належить введена користувачем точка заданому прямокутнику.

Створити похідний від нього клас „паралелепіпед” $b_1 \leq x_1 \leq a_1, b_2 \leq x_2 \leq a_2, b_3 \leq x_3 \leq a_3$ ”, де $a_i, b_i, i = \overline{1,3}$ – задані числа. Перевантажити відповідні методи:

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- визначення чи належить введена користувачем точка заданому паралелепіпеду.

Створити об'єкти класів „паралелепіпед” та „прямокутник”. Визначити, чи належать введені користувачем точки вказаним об'єктам.

9. Створити клас „система рівнянь $\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1, \\ a_{21}x_1 + a_{22}x_2 = b_2. \end{cases}$ ”.

Створити відповідні методи:

- задання коефіцієнтів рівнянь та вільних членів;
- виведення системи рівнянь на екран;
- визначення, чи задовольняє введений користувачем вектор $X = (x_1, x_2)$ даній системі рівнянь.

Створити похідний від нього клас „система лінійних алгебричних рівнянь

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3. \end{cases}$$

Перевантажити відповідні методи:

- задання коефіцієнтів рівнянь та вільних членів;
- виведення СЛАР на екран;
- визначення, чи задовольняє введений користувачем вектор $X = (x_1, x_2, x_3)$ даній СЛАР.

Створити об'єкти класів „система рівнянь 2×2 ”, „СЛАР 3×3 ”. Визначити, чи задовольняють введені користувачем вектори створеним об'єктам «система рівнянь 2×2 » та «СЛАР 3×3 ».

10. Описати клас „дріб $\frac{1}{ax}$ ”.

Створити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення підхідного дробу у введеній користувачем точці x .

$$\frac{1}{a_1x + \frac{1}{a_2x + \frac{1}{a_3x}}}, \text{ де } a_i$$

Створити похідний від нього клас „тривимірний підхідний дріб виду

$$, i = \overline{1,3} \text{ – задані числа, причому } a_3 \neq 0 .$$

Перевантажити відповідні методи:

- задання відповідних коефіцієнтів;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення підхідного дробу у введеній користувачем точці x .

Створити об'єкти класів „дріб” та „тривимірний похідний дріб”. Обчислити значення дробів у введених користувачем точках.

11. Описати клас „трикутник”, який визначається координатами трьох точок на площині.

Створити відповідні методи:

- задання координат трикутника;
- виведення відповідних координат на екран;
- обчислення площі трикутника.

Створити похідний від нього клас „тетраedr”, який визначається координатами чотирьох точок у просторі і перевантажити відповідні методи:

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

- задання координат вершин тетраедра;
- виведення відповідних коефіцієнтів на екран;
- обчислення об'єму тетраедра.

Створити об'єкти класів „трикутник” та „тетраедр”. Обчислити площу трикутника та об'єм тетраедра.

12. Описати клас „лінійна функція $a_1x + a_0$ ”.

Створити відповідні методи:

- задання коефіцієнтів лінійної функції;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення лінійної функції в точці x .

Створити похідний клас „многочлен $a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ ”, який визначається коефіцієнтами a_i , $i = \overline{0,4}$ з відповідними перевантаженими методами:

- задання коефіцієнтів многочлена;
- виведення відповідних коефіцієнтів на екран;
- обчислення значення многочлена в точці x .

Створити об'єкти класів „лінійна функція” та „многочлен”. Обчислити значення функцій у введеній користувачем точці.

13. Дано клас „рівносторонній трикутник”.

Створити відповідні методи:

- задання значення довжини сторони та кутів;
- знаходження інших характеристик трикутника: довжин сторін;
- обчислення периметра.

Створити похідний клас „трикутник”, який визначається довжиною однієї із сторін та значеннями прилеглих двох кутів з відповідними перевантаженими методами:

- задання значення довжини сторони та двох кутів;
- знаходження інших характеристик трикутника: величини кутів та довжин сторін;
- обчислення периметра.

Створити об'єкти класів „рівносторонній трикутник” та „трикутник”. Знайти інші характеристики створених трикутників та їх периметри.

14. Описати клас „полярна система координат” та створити відповідні методи:

- задання координат в полярній системі координат;
- задання координат в декартовій системі;
- перетворення координат заданої точки з полярної системи в декартову.
- перетворення координат заданої точки з декартової системи в полярну.

Створити похідний клас „циліндрична система координат” і перевантажити відповідні методи:

- задання координат в циліндричній системі координат;
- задання координат в декартовій системі;
- перетворення координат заданої точки з циліндричної системи в декартову.
- перетворення координат заданої точки з декартової системи в циліндричну.

Створити об'єкти класів „циліндрична система координат” та „полярна система координат”. Перевести введені користувачем точки з циліндричної системи в декартову та з декартової в полярну систему.

15. Описати клас „еліпс $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ ”. Створити відповідні методи:

- задання коефіцієнтів;
- виведення коефіцієнтів на екран;

Описати похідний клас „крива другого порядку $a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + b_1x + b_2y + c = 0$ ” з відповідними перевантаженими методами:

- задання коефіцієнтів;
- виведення коефіцієнтів на екран;
- визначення чи належить задана точка (х,у) даній кривій другого порядку.

Створити об'єкт класу „еліпс” і визначити чи належить введена користувачем точка (х,у) даному еліпсу.

16. Створити клас „система двох векторів $A = (a_1, a_2)$, $B = (b_1, b_2)$ ” і описати відповідні методи:

- задання координат векторів;
- виведення координат вектора на екран;
- визначення, чи система векторів A, B є лінійно незалежною.

Описати похідний клас „система 3-х векторів $A = (a_1, a_2, a_3)$, $B = (b_1, b_2, b_3)$, $C = (c_1, c_2, c_3)$ ” з відповідними перевантаженими методами:

- задання координат векторів;
- виведення координат вектора на екран;
- визначення, чи система векторів A, B, C є лінійно незалежною.

Створити об'єкти класів „система 2-х векторів” та „система 3-х векторів”. Визначити чи система даних векторів не є лінійно залежними.

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

17. Описати клас „людина” який містить ім'я, прізвище, по-батькові, число, місяць, рік народження і описати відповідні методи:

- задання відповідних даних;
- визначення за поточною введеною датою віку людини;
- обчислення кількості зустрічань певної літери (літера вводиться користувачем) в прізвищі людини.

Описати похідний клас „студент”, що містить додаткове поле рік вступу до ВУЗу та спеціальність з відповідними перевантаженими методами:

- задання відповідних даних;
- визначення за поточною введеною датою віку студента;
- обчислення кількості зустрічань певної літери (літера вводиться користувачем) в прізвищі людини.

Створити об'єкти класів „людина” та „студент”. За поточною введеною датою визначити вік студента. Визначити кількість зустрічань введеної користувачем літери в прізвищі людини.

18. Дано клас „прямокутний трикутник”.

Створити відповідні методи:

- обчислення периметра.

Створити похідний клас „трикутник”, який визначається довжинами двох сторін та значенням кута між ними та з відповідними перевантаженими методами:

- задання значення довжини сторони та двох кутів;
- знаходження інших характеристик трикутника: величини кутів та довжин сторін;
- обчислення периметра.

Створити об'єкти класів „прямокутний трикутник” та „трикутник”. Знайти інші характеристики створених трикутників та їх периметри.

19. Описати клас „система двох лінійних нерівностей” $\begin{cases} a_{11}x_1 + a_{12}x_2 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 \leq b_2. \end{cases}$, створити відповідні методи:

- задання коефіцієнтів відповідних нерівностей;
- виведення коефіцієнтів на екран;
- визначення, чи задовольняє введений користувачем вектор $X = (x_1, x_2)$ даній системі нерівностей.

Створити похідний клас „система трьох лінійних нерівностей” $\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 \leq b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 \leq b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \leq b_3. \end{cases}$ і перевантажити відповідні методи:

- задання коефіцієнтів відповідних нерівностей;
- виведення коефіцієнтів на екран;
- визначення, чи задовольняє введений користувачем вектор $X = (x_1, x_2, x_3)$ даній системі нерівностей.

Створити об'єкт класу „система лінійних нерівностей” і визначити чи введена користувачем точка задовольняє даній системі.

Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»
Жуковський В.В.

20. Описати клас „практикант”, який містить прізвище та ім'я практиканта; назву ВУЗу, в якому вчиться практикант. Описати відповідні методи:
- задання вказаних даних;
 - визначення чи прізвище є симетричним.
- Створити похідний клас „працівник фірми” з відповідними перевантаженими методам, який також містить дату прийому на роботу в фірму; назву навчального закладу, який закінчив та посаду:
- задання вказаних даних;
 - визначення стажу роботи на фірмі;
 - визначення чи прізвище є симетричним.
- Створити об'єкти класів „працівник фірми” та „практикант”. Для працівника визначити стаж його роботи на фірмі. Для практиканта визначити чи його прізвище є симетричним.
21. Описати клас „квадратне рівняння $b_2x^2 + b_1x + b_0 = 0$ ”, який містить:
- методи задання коефіцієнтів b_2, b_1, b_0 та метод виведення їх на екран;
 - метод визначення, чи задовільняє введене користувачем число x даному рівнянню.
- Поповнити даний клас методом пошуку коренів квадратного рівняння.
- Створити похідний клас „кубічне рівняння $a_3x^3 + a_2x^2 + a_1x + a_0 = 0$ ”. Перевантажити відповідні методи:
- методи задання коефіцієнтів a_3, a_2, a_1, a_0 та метод виведення їх на екран;
 - метод визначення, чи задовільняє введене користувачем число x даному рівнянню.
- Створити об'єкт класів „квадратне рівняння” та „кубічне рівняння”. Знайти корені квадратного рівняння. Визначити, чи задовільняє введене користувачем число x кубічному рівнянню.
22. Створити клас „коло $(x - x_0)^2 + (y - y_0)^2 = R^2$ ” і створити відповідні методи:
- задання координат (x_0, y_0) центру кола та його радіусу R , а також виведення цих даних на екран;
 - визначення довжини кола.
- Створити похідний клас „сфера $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$ ”. Перевантажити відповідні методи:
- задання координат (x_0, y_0, z_0) центру сфери та її радіусу R , а також виведення цих даних на екран;
 - визначення площі поверхні сфери.
- Створити об'єкти класів „коло $(x - x_0)^2 + (y - y_0)^2 = R^2$ ” та „сфера $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2$ ”. Визначити довжину кола та площу поверхні сфери.

23. Описати клас „рухома матеріальна точка”, яка рухається по прямій і координата її

визначається як $x = x_0 + a_1 \sin(t), y = 0, z = 0$.

Створити відповідні методи:

- задання початкового положення точки x_0 ;
- задання коефіцієнтів a_1 та виведення їх на екран;
- визначення координат точки в заданий момент часу t .

Створити похідний клас „рухома матеріальна точка (x, y, z) ”, координати якої визначаються

як $x = x_0 + a_1 \sin(t), y = y_0 + a_2 \cos(t), z = z_0 + a_3 t^2$. Перевантажити відповідні методи:

- задання початкового положення точки (x_0, y_0, z_0) ;
- задання коефіцієнтів a_3, a_2, a_1 та виведення їх на екран;
- визначення координат точки в заданий момент часу t .

Створити об'єкти класів „рухома матеріальна точка x ” та „рухома матеріальна точка (x, y, z) ” і визначити їх положення у введений користувачем момент часу t .

24. Задано клас „коло в просторі”, який буде характеризуватись центром та радіусом.

Описати метод:

- задання вказаних даних та виведення їх на екран;
- метод визначення довжини кола.

Описати похідний клас „конус”, який характеризується координатами вершини, координатами центру основи та твірною. Перевантажити відповідні методи:

- задання вказаних даних та виведення їх на екран;
- метод визначення бічної поверхні конуса;
- метод визначення радіуса основи конуса.

Створити об'єкти класів „коло в просторі” та „конус”. Визначити радіус основи конуса. Визначити довжину кола.

25. Описати клас „перетворення площини $\begin{cases} x' = a_{11}x + a_{12}y + a_{13} \\ y' = a_{21}x + a_{22}y + a_{23} \end{cases}$ ” та створити відповідні методи:

- задання коефіцієнтів перетворення;
- виведення коефіцієнтів перетворення на екран;
- визначення образу заданої точки (x, y) .

Описати похідний клас „перетворення простору $\begin{cases} x' = a_{11}x + a_{12}y + a_{13}z + a_{14} \\ y' = a_{21}x + a_{22}y + a_{23}z + a_{24} \\ z' = a_{31}x + a_{32}y + a_{33}z + a_{34} \end{cases}$ ” з відповідними перевантаженими методами:

- задання коефіцієнтів перетворення;
- виведення коефіцієнтів перетворення на екран;
- визначення образу заданої точки (x, y, z) .

Створити об'єкт класу „перетворення площини” і знайти образ введеної користувачем точки.

Хід виконання

1. Розробити алгоритми створення класів згідно варіантів завдань для самостійної роботи.
2. Написати відповідні програми на одній з мов програмування.
3. Виконати дане завдання на комп'ютері.
4. Захистити лабораторну роботу. оперативна пам'ять 256Мб) встановити операційну

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Що розуміється під „наслідуванням”?
2. Чи допускають мови програмування C++ (C#) множинне наслідування?
3. Чи можливе заміщення конструктора батьківського класу в похідному класі?
4. Чи можливий виклик деструктора батьківського класу в методах похідного класу?
5. Основне призначення наслідування?
6. Чи можливе заміщення методів батьківського класу в похідному класі?
7. Як викликати метод батьківського класу в похідному класі, якщо в похідному класі також існує метод з таким-же іменем?
8. Чи можна створити об'єкт похідного класу, якщо не створено жодного об'єкта батьківського класу?
9. Побудуйте ієрархію класів для поняття „число”.
10. Побудуйте ієрархію класів для поняття „ромб”.

Лабораторна робота №5

Назва роботи: Використання механізму віртуальних методів

Мета роботи: Ознайомитися з особливостями використання віртуальних методів.

Теоретичні відомості

Мова C# включає таку властивість, як **поліморфізм** – можливість для об'єктів різних класів, зв'язаних з допомогою наслідування, реагувати різними способами при виклику однієї функції-елемента. До найважливіших форм поліморфізму можна віднести:

- перевантаження методів та операцій;
- віртуальні методи (функції);

Перевантаження методів та операцій називаються **статичним поліморфізмом**, тому що він підтримується на етапі компіляції (тобто до запуску програми). **Віртуальні методи** відносяться до **динамічного поліморфізму**, тому що він реалізується при виконанні програм.

При використанні віртуального методу запит здійснюється з допомогою вказівника базового класу (або посилання), тобто середовище виконання C# вибирає правильно перевизначений метод у відповідному похідному класі, який зв'язаний з цим об'єктом.

Іноді метод визначається в базовому класі як віртуальний, але просто перевизначений в похідному класі. Якщо такий метод викликається через вказівник похідного класу, то використовується версія похідного класу. Це не поліморфна поведінка.

Завдяки використанню поліморфних методів та поліморфізму виклик функції-елемента може привести до різних дій, які залежать від типу об'єкту, який викликається.

Завдання

Написати програму, описавши батьківський та похідний класи і використавши віртуальні методи. За основу візьміть попередню лабораторну і додайте віртуальні методи. Використайте динамічне створення об'єктів та показники на екземпляр класу.

Продемонструвати роботу віртуальних методів на практичному прикладі коли наперед невідомо який об'єкт буде створено (невідомо на етапі компіляції):

```
//Вибрати режим роботи-запитати в користувача, змінна userChoose
if (userChoose=='1'){
    //Працюємо з одним об'єктом
    //Створюємо, ініціалізуємо ітд
}
else{
    //Працюємо з іншим об'єктом
    //Створюємо, ініціалізуємо ітд
}
//Виклик віртуальної функції через вказівник/показник на базовий клас
```

Змініть віртуальні методи на звичайні та перевірте роботу програми. Проаналізуйте зміни.

Приклад програми:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Celipsoid
    {
        public int a1, a2, a3, b1, b2, b3;
        public double v;
        virtual public void inita()
        {
            System.Console.WriteLine("Введіть півосі еліпсоїда a1, a2, a3:");
            a1 = Convert.ToInt32(Console.ReadLine());
            a2 = Convert.ToInt32(Console.ReadLine());
            a3 = Convert.ToInt32(Console.ReadLine());
        }
        virtual public void initb()
        {
            System.Console.WriteLine("Введіть координати зміщення центру b1, b2,
b3:");
            b1 = Convert.ToInt32(Console.ReadLine());
            b2 = Convert.ToInt32(Console.ReadLine());
            b3 = Convert.ToInt32(Console.ReadLine());
        }
        virtual public void show()
        {
            Console.WriteLine("a1= " + a1);
            Console.WriteLine("a2= " + a2);
            Console.WriteLine("a3= " + a3);
            Console.WriteLine("b1= " + b1);
            Console.WriteLine("b2= " + b2);
            Console.WriteLine("b3= " + b3);
        }
        virtual public double size()
        {
            v = 4.0 * a1 * a2 * a3 / 3.0;
            Console.Write("v еліпсоїда = ");
            Console.WriteLine(v);
            return (v);
        }
    }

    class Cball : Celipsoid
    {
        public int r;
        public override void inita()
        {
            System.Console.Write("Введіть радіус кулі:");
            r = Convert.ToInt32(Console.ReadLine());
            base.initb();
        }
        public override void show()
        {
            Console.WriteLine("r= " + r);
            Console.WriteLine("b1= " + b1);
            Console.WriteLine("b2= " + b2);
        }
    }
}
```

```
        Console.WriteLine("b3= " + b3);
    }
    public override double size()
    {
        v = 4.0 * r * r * r / 3.0;
        Console.Write("v кулі = ");
        Console.WriteLine(v);
        return (v);
    }
}

class Program
{
    static void Main(string[] args)
    {
        int userSelect;
        Celipsoid baseobj = new Celipsoid();
        do
        {
            Console.WriteLine("Enter '0' if you want to work with elipsoid and  
'1' - with ball");
            userSelect = Convert.ToInt32(Console.ReadLine());
            if (userSelect == 0)
            {
                baseobj = new Celipsoid();
                baseobj.initb();
            }
            else if (userSelect == 1)
            {
                baseobj = new Cball();
            }
            else
            {
                return;
            }
            baseobj.inita();
            baseobj.show();
            baseobj.size();
        } while (true);
    }
}
```

Варіанти завдань

Згідно попередньої роботи.

Хід виконання

1. Розробити алгоритми створення класів згідно варіантів завдань.
2. Написати відповідну програму на мові програмування C#.
3. Проаналізувати роботу з віртуальними методами та звичайними. Тобто потрібно дослідити механізм динамічного поліморфізму. Зверніть увагу на приклад коду, зокрема на

**Методичні вказівки до виконання лабораторних робіт з дисципліни
«Об'єктно-орієнтоване програмування»**

Жуковський В.В.

механізм створення об'єкту одного з класів. На етапі компіляції програми невідомо який вибір зробить користувач під час виконання програми.

4. Порівняти результати виконання програми з віртуальними методами та без віртуальних методів.
5. Зробити звіт та захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Що розуміється під „наслідуванням”?
2. Чи допускають мови програмування C++ (C#) множинне наслідування?
3. Що таке поліморфізм?
4. Що таке віртуальний метод (функція)?
5. Який механізм роботи віртуальних методів (функцій)?
6. Для чого можна використовувати віртуальні методи (функції)?
7. Що таке статичний поліморфізм?
8. Що таке динамічний поліморфізм?

Лабораторна робота №6

Назва роботи: Абстрактні класи

Мета роботи: Навчитись використовувати абстрактні класи та інтерфейси.

Теоретичні відомості

Приклад використання абстрактних класів та інтерфейсів

```
using System;

namespace AbstractsANDInterfaces
{
    public abstract class Employee
    {
        //we can have fields and properties
        //in the Abstract class
        protected String id;
        protected String lname;
        protected String fname;
        //properties
        public abstract String ID
        {
            get;    set;
        }

        public abstract String FirstName
        {
            get;    set;
        }

        public abstract String LastName
        {
            get;    set;
        }
        //completed methods
        public String Update()
        {
            return "Employee " + id + " " +
                lname + " " + fname +
                " updated";
        }
        //completed methods

        public String Add()
        {
            return "Employee " + id + " " +
                lname + " " + fname +
                " added";
        }
        //completed methods
        public String Delete()
        {
            return "Employee " + id + " " +
                lname + " " + fname +
```

```
        " deleted";
    }
    //completed methods
    public String Search()
    {
        return "Employee " + id + " " +
            lname + " " + fname +
            " found";
    }

    //abstract method that is different
    //from Fulltime and Contractor
    //therefore i keep it uncompleted and
    //let each implementation
    //complete it the way they calculate the wage.
    public abstract String CalculateWage();
}

//Приклад інтерфейсу
using System;
namespace AbstractsANDInterfaces
{
    public interface IEmployee
    {
        //cannot have fields. uncommenting
        //will raise error!
        //        protected String id;
        //        protected String lname;
        //        protected String fname;

        //just signature of the properties
        //and methods.
        //setting a rule or contract to be
        //followed by implementations.
        String ID
        {
            get; set;
        }
        String FirstName
        {
            get; set;
        }

        String LastName
        {
            get; set;
        }

        // cannot have implementation
        // cannot have modifiers public
        // etc all are assumed public
        // cannot have virtual
        String Update();
        String Add();
        String Delete();
        String Search();
        String CalculateWage();
    }
}
```

}
}

Завдання

За основу візьміть попередню лабораторну, проаналізуйте ваші класи, виділіть загальні особливості і додайте абстрактний клас. В класах повинні бути присутні конструктори та деструктори.

Продемонструйте роботу новостворених класів.

Хід виконання

1. Проаналізувати код.
2. Узагальнити спільні властивості та методи класів у абстрактний клас.
3. Написати програму з використанням абстрактного класу.
4. Створити ще одну версію програми із застосуванням інтерфейсу.
5. Проаналізувати та порівняти створені програми.
6. Спробувати створити екземпляр абстрактного класу. Проаналізувати поведінку IDE.
7. Продемонструвати результати викладачу.
8. Захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Що таке абстрактний клас?
2. Чи допускають мови програмування C++ (C#) створення об'єктів абстрактного класу?
3. Що таке інтерфейс?
4. Яка різниця між інтерфейсом та абстрактним класом?
5. Коли потрібно використовувати інтерфейс замість абстрактного класу?

Лабораторна робота №7

Назва роботи: Простір імен System.Drawing

Мета роботи: Навчитись використовувати класи з простору імен System.Drawing для рисування на формі.

Теоретичні відомості

Клас Graphics - це основний клас, необхідний для рисування на формі в додатках типу Form Application. Даний клас знаходиться в просторі імен System.Drawing.

Універсальний алгоритм рисування на формі включає такі кроки:

1. Отримати об'єкт класу Graphics для керування рисуванням. Це найкраще зробити в конструкторі форми, оголосивши змінну-член класу для цього.
2. Створити екземпляри класів Color та Pen для визначення стилю рисування.
3. Використовуючи змінну-член класу форми, яка представляє об'єкт класу Graphics (див. пункт 1), рисуємо на формі за допомогою методів DrawLine(), DrawCurve(), DrawRectangle() ітд.

Код класу Form1, що рисує лінії графіка при натисканні на кнопку

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;

namespace Graph
{
    public partial class Form1 : Form
    {
        Graphics graph; //declare
        public Form1()
        {
            InitializeComponent();
            graph = CreateGraphics(); //initialization
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //create color
            Color slateBlue = Color.FromName("SlateBlue");
            float width = 3; // set width of the pen
            // create pen
            Pen pen = new Pen(slateBlue, width);
            int y, y2, x;
            y = 0;
            x = 0;
            for (int x2 = 0; x2 < 1000; x2++)
            {
```

```
y2 = x2 * x2;  
//draw line  
graph.DrawLine(pen, x, y, x2, y2);  
x = x2;  
y = y2;  
}  
}  
}
```



Рис. 1. Результат виконання програми

Завдання

Використовуючи класи з простору імен System.Drawing нарисувати графік на формі. Потрібно реалізувати програму, яка буде рисувати графік на формі в залежності від розмірів вікна. Тобто при зміні розмірів вікна форми - графік повинен бути перерисований в новому масштабі.

Варіанти завдань отримаєте у викладача.

Варіанти завдань для самостійної роботи

1. $y = \frac{x + \cos 2x}{x + 2}$, $0,2 \leq x \leq 10$, $\Delta x = 0,8$;
2. $y = \frac{\lg 0,5x}{x^3 + 7,5}$, $0,1 \leq x \leq 1,2$, $\Delta x = 0,1$;

3. $y = \frac{6x+4}{\sin 3x-x}$, $2,3 \leq x \leq 7,8$ $\Delta x = 0,9$;
4. $y = \frac{2,5x^3}{e^{2x}+2}$, $0 \leq x \leq 0,5$, $\Delta x = 0,1$;
5. $y = \frac{1,5x - \ln 2x}{3x+1}$, $2,5 \leq x \leq 9$, $\Delta x = 0,8$;
6. $y = \frac{3x+1}{\arctg x}$, $0,1 \leq x \leq 1,5$, $\Delta x = 0,2$;
7. $y = \frac{\cos^2 x}{x^2+1}$, $3,8 \leq x \leq 7,6$, $\Delta x = 0,6$;
8. $y = \frac{e^{2x}-8}{x+3}$, $-1 \leq x \leq 2,3$, $\Delta x = 0,7$;
9. $y = \frac{x + \cos 2x}{3x}$, $2,3 \leq x \leq 5,4$, $\Delta x = 0,8$;
10. $y = \frac{\cos^3 t^2}{1,5t+2}$, $2,3 \leq t \leq 7,2$, $\Delta t = 0,8$;
11. $y = \frac{x^3+2x}{3\cos\sqrt{x}+1}$, $0 \leq x \leq 2$ $\Delta x = 0,4$;
12. $z = \frac{t + \sin 2t}{t^2-3}$, $2,4 \leq t \leq 6,9$ $\Delta x = 0,4$;
13. $y = \frac{x^3-2}{3\ln x}$, $4,5 \leq x \leq 16,4$ $\Delta x = 2,2$;
14. $y = \frac{\arccos x}{2x+1}$, $0,1 \leq x \leq 0,9$ $\Delta x = 0,1$;
15. $y = \frac{5tg(x+7)}{(x+3)^2}$, $1,2 \leq x \leq 6,3$ $\Delta x = 0,2$;
16. $y = \frac{5\lg x}{x^2-1}$, $1,2 \leq x \leq 3,8$ $\Delta x = 0,4$;
17. $z = \frac{2\sin^2(x+2)}{x^2+1}$, $7,2 \leq x \leq 12$ $\Delta x = 0,5$;
18. $y = \frac{(3x+2)^2}{\sin x+3}$, $4,8 \leq x \leq 7,9$ $\Delta x = 0,4$;
19. $y = \frac{tg 2t-3t}{t+3}$, $0,2 \leq t \leq 0,8$ $\Delta t = 0,1$;
20. $y = \frac{(x+2)^2}{\sqrt{x^2+1}}$, $2,3 \leq x \leq 8,3$ $\Delta x = 0,6$;

21. $y = \frac{t - \ln 2t}{3t + 1}$, $2,1 \leq t \leq 8,5$ $\Delta t = 0,7$;
22. $y = \frac{x^2 + 2x}{\cos 5x + 2}$, $-2 \leq x \leq 4,5$ $\Delta x = 0,5$;
23. $y = \frac{\ln|x+1| + 5}{2x + 3}$, $0,2 \leq x \leq 0,9$ $\Delta x = 0,15$;
24. $y = \frac{2t + 8}{|\cos 3t| + 1}$, $2 \leq t \leq 6,5$ $\Delta x = 0,8$;
25. $y = 1/\cos^3(2x) - \sqrt{x^2 - 4}$, $2 \leq x \leq 7$, $\Delta x = 0,05$.

Хід виконання

1. Прочитати теоретичні відомості.
2. Створити програму для побудови графіка.
3. Продемонструвати результати викладачу.
4. Оформити звіт
5. Захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи

Завдання.

Текст програми.

Висновки

Контрольні запитання

1. Які класи з простору імен System.Drawing можуть використовуватись для рисування?

Лабораторна робота №8

Назва роботи: Регулярні вирази

Мета роботи: Навчитись використовувати класи з простору імен RegularExpression для операцій з текстом.

Теоретичні відомості

Регулярні вирази - це один із способів пошуку підрядків (відповідностей) в рядках. Здійснюється за допомогою перегляду рядки в пошуках деякого шаблону. Застосування регулярних виразів дає значне збільшення продуктивності, оскільки бібліотеки, що інтерпретують регулярні вирази, зазвичай пишуться на низько-рівневих високопродуктивних мовах (C, C++, Assembler).

Завдання

Використовуючи класи з простору імен RegularExpression виконати завдання згідно варіанту. Варіанти завдань отримаєте у викладача.

Варіанти завдань для самостійної роботи

1. Заданий текст. Визначити чи входить в нього задане слово і скільки разів.
2. Заданий текст. Визначити чи є він кодом HTML: містить теги <html>, <form>, <h1>.
3. Заданий текст. Визначити чи є він текстом англійською мовою.
4. Заданий текст. Після кожної літери «о» вставити сполучення «OK».
5. Заданий текст. Визначити чи є він текстом російської мови.
6. Заданий текст. Визначити чи містить він цифри.
7. Заданий текст. Визначити скільки варіантів починається зі слова "Інформатика".
8. Заданий текст. Вибрати з нього всі 7-значні номери телефонів.
9. Заданий текст. Визначити чи містить він цифри.
10. Заданий текст. Вибрати з нього всі e-mail адреси.
11. Задано текст, що містить букви і цифри. Знайти всі числа в тексті.
12. Задано речення. Роздрукувати всі слова в стовпчик.
13. Заданий текст. Визначити кількість приголосних букв в ньому і роздрукувати їх.
14. Заданий текст. Визначити чи є він текстом української мови.

Хід виконання

1. Прочитати теоретичні відомості.
2. Створити програму для вирішення завдання.
3. Продемонструвати результати викладачу.
4. Оформити звіт.

5. Дати відповіді на контрольні запитання.
6. Захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи.

Завдання.

Текст програми.

Висновки.

Контрольні запитання

1. Що таке регулярні вирази. Застосування регулярних виразів.
2. Задання регулярного виразу. Пошук підрядків за допомогою регулярного виразу.
3. Клас Regex і його методи.
4. Пошук першого входження зразка з текст.
5. Пошук всіх входжень зразка в текст.
6. Заміна зразка в тексті.
7. Розбір тексту на слова.
8. Властивості класу Match.
9. Класи Match і MatchCollection та їх властивості.
10. Класи Group і GroupCollection та їх властивості.

Лабораторна робота №9

Назва роботи: Породжуючі паттерни

Мета роботи: Навчитись використовувати породжуючі паттерни абстрактна фабрика, будівельник та фабричний метод.

Теоретичні відомості

Основним завданням породжуючих паттернів є спрощення створення об'єктів.

Іноколи ви працюєте із певним набором об'єктів через групу інтерфейсів. А тоді хочете створювати об'єкти із іншого набору, щоб пристосувати ваш код до інших умов. Звичайно, група інтерфейсів, через які ви оперуєте, залишається та ж сама. Спростити створення відповідного набору об'єктів допоможе Абстрактна Фабрика.

Бувають випадки коли структура деякого об'єкта дуже складна і залежить від багатьох чинників. Щоб спростити створення такого об'єкту зазвичай використовують Будівельника.

А щоб зручно вибрати одну реалізацію та інстанціювати її, відштовхуючись від простої умови, можна використати Фабричний Метод.

Нерідко постає завдання отримати копію уже існуючого об'єкта або отримати можливість швидко генерувати багато подібних екземплярів. У такому випадку Прототип якраз згодиться.

Завдання

Для студентів спеціальностей 113 Прикладна математика та 122 Комп'ютерні науки завдання наступне:

За основу візьміть лабораторну роботу «Використання механізму віртуальних функцій». Модифікуйте її так, щоб при створенні об'єктів використовувався механізм породжуючих паттернів.

Абстрактну фабрику реалізують ті варіанти, у яких остача від ділення на 3 рівна 0 (3, 6, 9, 12, ...)

Будівельника реалізують ті варіанти, у яких остача від ділення на 3 рівна 1 (1, 7, 10, 13, ...)

Фабричний метод реалізують ті варіанти, у яких остача від ділення на 3 рівна 2 (2, 8, 14, 17, ...)

Для студентів спеціальності 121 Інженерія програмного забезпечення завдання наступне:

1. Потрібно переглянути приклади практичної реалізації паттернів проектування за даним посиланням <https://www.globallogic.com/ua/insights/blogs/dive-into-the-selenium-patterns/>.
2. Ознайомитися з фреймворком тестування cypress <https://www.cypress.io/>
3. Реалізувати патерн проектування на вибір (Page Object, Page factory, Builder) при тестуванні конкретного веб-ресурсу вказаного викладачем.

Прикладом такого ресурсу для тестування може бути даний проект

https://drive.google.com/file/d/1jEjgvEGPOyIuQB2hqq62VrgT_4kLsoaf/view?usp=sharing.

Для запуску даного проекту необхідно розпакувати архів Angular додатку, відкрити директорію за допомогою редактора коду, наприклад Visual Code

(<https://code.visualstudio.com/>). Наступним кроком буде оновлення бібліотек `npm i --force` та запуск проекту командою `ng serve`. Потім потрібно перейти за

посиланням, що з'явиться в консолі. Якщо не з'явиться, понизьте версію Node.js до 16. Для запуску тестів cypress використовуйте команду `npm cypress open`. Зверніть увагу, що даний проект повинен використовувати тестовий backend сервер і, відповідно, вам потрібно знати логін і пароль користувача для авторизації (запитайте викладача про цей логін та пароль).

Хід виконання

1. Прочитати теоретичні відомості.
2. Створити програму для вирішення завдання.
3. **Нарисувати UML діаграму згідно програми.**
4. Продемонструвати результати викладачу.
5. Оформити звіт.
6. Дати відповіді на контрольні запитання.
7. Захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи.

Короткі теоретичні відомості по патерну.

Завдання.

Текст програми.

UML діаграма.

Висновки.

Корисні посилання

<https://www.globallogic.com/ua/insights/blogs/dive-into-the-selenium-patterns/>

<https://www.youtube.com/watch?v=1LQ4XiCNmvA>

Контрольні запитання

1. Що таке паттерни проектування.
2. Як класифікуються паттерни проектування.
3. Що таке породжуючі паттерни проектування.
4. Для чого використовується абстрактна фабрика. Нарисуйте UML діаграму.
5. Наведіть приклад практичного застосування патерну.
6. Для чого використовується фабричний метод. Нарисуйте UML діаграму.
7. Для чого використовується будівельник. Нарисуйте UML діаграму.

Лабораторна робота №10

Назва роботи: Структурні паттерни

Мета роботи: Навчитись використовувати структурні паттерни згідно поставленої задачі

Теоретичні відомості

Основним завданням структурних патернів є формування найбільш оптимальної структури та взаємодії між класами для виконання поставлених завдань.

Якщо потрібно, щоб один об'єкт міг бути зрозумілим під іншим інтерфейсом, використовується **Адаптер**.

Якщо ви хочете розділити абстракцію та імплементацію так, щоб на одному боці мати абстракцію, а на іншому декілька реалізацій, причому, всі доступні до модифікацій, то потрібно задуматися над поєднанням таких незалежних абстракції та реалізацій за допомогою патерну **Міст**.

Якщо елемент містить собі подібні елементи, а вони, в свою чергу, також можуть містити елементи, то таку структуру найлегше реалізувати за допомогою **Компонувальника**.

Для швидкої та динамічної можливості розширення існуючої функціональності, без зміни її носія, можна скористатися **Декоратором**.

Якщо ваша система використовує багато об'єктів, що мають спільні дані, то такі дані можна винести та зробити загальнодоступними для економії пам'яті за допомогою патерну **Легковаговик**.

Якщо відсутня можливість працювати із об'єктом напряду, використовуйте **Проксі**, що дозволить донести ваші команди до пункту призначення.

Завдання

Для студентів спеціальностей 113 Прикладна математика та 122 Комп'ютерні науки завдання наступне:

За основу візьміть довільну лабораторну роботу. Модифікуйте її так, щоб продемонструвати роботу структурного патерну згідно варіанту завдання.

Адаптер реалізують ті варіанти, у яких остача від ділення на 6 рівна 0 (6, 12, ...)

Міст реалізують ті варіанти, у яких остача від ділення на 6 рівна 1 (1, 13, ...)

Компонувальник реалізують ті варіанти, у яких остача від ділення на 6 рівна 2 (2, 14, ...)

Декоратор реалізують ті варіанти, у яких остача від ділення на 6 рівна 3 (3, 15, ...)

Легковаговик реалізують ті варіанти, у яких остача від ділення на 6 рівна 4 (4, 16, ...)

Проксі реалізують ті варіанти, у яких остача від ділення на 6 рівна 5 (5, 17, ...)

Для студентів спеціальності 121 Інженерія програмного забезпечення завдання наступне:

4. Потрібно переглянути приклади практичної реалізації патернів проектування за даним посиланням <https://www.globallogic.com/ua/insights/blogs/dive-into-the-selenium-patterns/>.
5. Ознайомитися з фреймворком тестування cypress <https://www.cypress.io/>
6. Реалізувати патерн проектування на вибір (Singleton, Facade, Strategy, Decorator) при тестуванні конкретного веб-ресурсу вказаного викладачем

Хід виконання

1. Прочитати теоретичні відомості.
2. Створити програму для вирішення завдання.
3. Нарисувати UML діаграму згідно програми.
4. Продемонструвати результати викладачу.
5. Оформити звіт.
6. Дати відповіді на контрольні запитання.
7. Захистити лабораторну роботу.

Зміст звіту:

Назва та мета роботи.

Короткі теоретичні відомості по патерну.

Завдання.

Текст програми.

UML діаграма.

Висновки.

Контрольні запитання

1. Що таке паттерни проектування.
2. Як класифікуються структурні паттерни?
3. Яке призначення структурних паттернів проектування?
4. Для чого використовується Адаптер. Нарисуйте UML діаграму.
5. Для чого використовується Міст. Нарисуйте UML діаграму.
6. Для чого використовується Компонувальник. Нарисуйте UML діаграму.
7. Для чого використовується Декоратор. Нарисуйте UML діаграму.
8. Для чого використовується Легковаговик. Нарисуйте UML діаграму.
9. Для чого використовується Легковаговик. Нарисуйте UML діаграму.
10. Наведіть приклад практичного застосування патерну.