**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

Циклова комісія програмування та інформаційних технологій

**КУРСОВА РОБОТА**

з дисципліни «Інструментальні засоби візуального програмування»

на тему: «Автоматизація роботи магазину «Універсам»»

|  |  |  |
| --- | --- | --- |
| ЗАХИЩЕНО З ОЦІНКОЮ  Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_  Кількість балів \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Керівник  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Таран В.В.  (підпис) |  | ВИКОНАЛА |
| студентка 3-го курсу гр. ІПЗ-31 спеціальність 121 «Інженерія програмного забезпечення» |
| Бондарєва Єлизавета Юріївна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис виконавця) |

Члени комісії Таран В.В.

(підпис)

Заболотний В.О.

(підпис)

м. Київ — 2022

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ**

**«ОПТИКО-МЕХАНІЧНИЙ ФАХОВИЙ КОЛЕДЖ КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

ЗАТВЕРДЖУЮ

В.о. директора

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022р.

|  |  |  |
| --- | --- | --- |
| **Дисципліна:** | Інструментальні засоби візуального програмування | |
| **Спеціальність:** | 121 «Інженерія програмного забезпечення» | |
| **Курс ІІІ** | **Група IПЗ-31** | **Семестр** **6** |

**ЗАВДАННЯ  
на курсову роботу студентки**

|  |
| --- |
| Бондарєвої Єлизавети Юріївни |
| **1.Тема курсової роботи:**  Автоматизація роботи магазину «Універсам» |
| Затверджена протоколом циклової комісії №06 від 25 січня 2022 р. |
| **2.Технічні вимоги:** ЄСПД, ЄСКД, ДСТУ 3008-95 |
| **3.Термін здачі студентом закінченої роботи: «**\_\_» червня 2022 р. |
| **4.Вихідні дані до виконання роботи:** |
| * Проаналізувати предметну область, виконати опис проектованої системи |
| і правил її функціонування; |
| * Розробити графічний інтерфейс користувача для роботи з його |
| елементами: головне вікно програми, система меню форм, довідкова система та інформація про програму; |
| **5.Зміст пояснювальної записки** (перелік питань, які підлягають розробці): |
| вступ; постановка задачі; характеристика апаратно-програмної частини; |
| опис мови і середовища програмування; розробка алгоритму та його опис; |
| створення та тестування програми; висновки; перелік умовних позначень; |
| список літератури; програмний лістинг. |
| **6. Дата видачі завдання:** «05» квітня 2022 р. |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Таран Вікторія Валеріївна |
| Завдання прийняв до виконання |  | Бондарєва Єлизавета Юріївна |

«05» квітня 2022 р.

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК**

**виконання курсової роботи з дисципліни**

**«Інструментальні засоби візуального програмування»**

|  |  |  |
| --- | --- | --- |
| **№ з/п** | **Етапи** | **Термін виконання** |
|  | Обрання та затвердження теми, попередня постановка задачі | 01.04-05.04.2022 |
|  | Підбір літератури, веб-джерел, ознайомлення зі стандартами | 06.04-12.04.2022 |
|  | Вивчення та аналіз предметної області об'єкту курсової роботи | 13.04-19.04.2022 |
|  | Підготовка оглядової частини | 20.04-26.04.2022 |
|  | Розробка архітектури та загальної структури системи | 27.04-03.05.2022 |
|  | Програмна реалізація системи | 04.05-10.05.2022 |
|  | Оформлення роботи, передача її на рецензування керівникові | 11.05-17.05.2022 |
|  | Рецензування роботи, уточнення назви, висновків, підготовка презентаційних матеріалів | 18.05-24.05.2022 |
|  | Попередній захист курсової роботи | 25.05-07.06.2022 |
|  | Захист курсової роботи | 08.06.2022 |

|  |  |  |
| --- | --- | --- |
| Керівник роботи |  | Таран Вікторія Валеріївна |
| Студентка |  | Бондарєва Єлизавета Юріївна |
|  |  | « » 2022 р. |

**Зміст**

[**Реферат** 5](#_Toc11986430)

[**Скорочення** 6](#_Toc11986431)

**Вступ**  [7](#_Toc11986432)

[**Розділ 1.Опис напрямків дослідження**](#_Toc11986433)

[1.1 Дослідження предметної області 8](#_Toc11986434)

[1.2 Постановка задачі 1](#_Toc11986435)2

[**Розділ 2. Розробка програми**](#_Toc11986436)

[2.1 Опис алгоритму програми та розробка програмного коду 1](#_Toc11986437)4

[2.2 Опис використання програмних засобів об’єктно-орієнтованого програмування 18](#_Toc11986438)

[**Висновок** 2](#_Toc11986441)2

[**Список використаної літератури** 2](#_Toc11986442)3

[**Додатки** 2](#_Toc11986443)4

# **Реферат**

# Курсова робота присвячена розробці програмного продукту для автоматизації роботи магазину «Універсам».

# Метою курсової роботи є: розробка програми автоматизації магазину «Універсам». В результаті виконання курсової роботи ми отримуємо працюючий програмний продукт магазин «Універсам». В подальшому розвитку передбачується у разі розширення продуктового асортименту, також розширення програмного коду для актуального та правильного функціонування програми.

# Курсова робота складається з 27 сторінок та 18 сторінок додатків. Кількість ілюстрацій – 17 рисунків, 5 додатків, 21 використане джерело.

# **Скорочення**

і т.д.- і так далі

див. – дивитися

ПЗ – програмне забезпечення

ООП- об’єктно-орієнтоване програмування

ІТ- інформаційні технології

Рис.-рисунок

# **ВСТУП**

Темою курсової роботи є автоматизація роботи магазину «Універсам».

Метою курсової роботи є розробка прикладного програмного забезпечення «Магазин» на мові програмування Microsoft Visual Studio С#.

Постановкою задачі є використовуючи принципи об’єктно-орієнтованого програмування (ООП), створити базовий клас і на основі базового класу будувати похідні класи.

Тема курсової роботи є актуальною, оскільки пов'язана сучасними подіями, які кояться у світі, а саме епідемія коронавірусу. Системи управління контентом наразі є прогресивним напрямком в IT сфері, воно надає змогу вирішувати складні завдання з великими обсягами даних.

Сучасний стан проблеми почав турбувати теперішніх гігантів сучасного світу, ще 20 років тому, зарубіжні бізнесмени та підприємці, такі як: Джек Ма, Дже́ффри Престон «Джефф» Бе́зос, Уи́льям Ге́нри Гейтс III, И́лон Рив Маск, Уоррен Эдвард Баффетт, Сторонский Николай Николаевич.

# **Розділ 1.****Опис напрямків дослідження**

## 1.1 Дослідження предметної області

## Клас — це тип, визначений користувачем. Він складається з влаштованих типів, інших типів, визначених користувачем, і функцій. Компоненти, використані при визначенні класу, називаются його членами (members). Клас може містити декілька членів, а може і не мати жодного члена.

## Оголошення класу має таку форму:

## class <сім’я класу> : <список класів-батьків>

## { public: // доступно для всіх відкритий доступ

## <дані, методи, властивості, події>

## protected: // доступно тільки нащадка захищений доступ

## <дані, методи, властивості, події>

## private: // доступно только в классе закритий доступ

## <дані, методи, властивості, події>

## } <список змінних>

## Визначення або оголошення класу починається з ключового слова class. Далі у фігурних дужках ({}) описується тіло класу. Визначення класу закінчується крапкою з комою. Тіло класу включає дані (члени-дані), функції (члени-функції) і специфікатори доступу до членів класу. Специфікатори доступу до членів класу завжди закінчуються двокрапкою (:) і можуть з'являтися у визначенні класу багато разів і у будь-якому порядку.

## Властивість ООП - можливість успадкування даних і функцій. У випадку, коли один клас успадковує інший, базовий клас називають батьківським, а той, що успадковує, — похідним або нащадком.

## Розміщення описів змінних і функції з їх обробки, тобто полів та методів в одному класі називається інкапсуляцією.

## Класи в С# мають три різних рівні доступу до своїх елементів, тобто даних (полів) і функцій (методів):

## • закриті елементи (private);

## • захищені елементи (protected);

## • відкриті елементи (public).

## До даних у закритому розділі (private) мають доступ тільки функції-елементи свого класу. Класам-нащадкам забороняється доступ до закритих даних своїх базових (батьківських) класів. За замовчуванням усі дані-члени класу мають атрибут private(закритий) (Рис.1.1).

## До даних у захищеному розділі (protected) мають доступ функції свого класу і функції класів-нащадків.

## У свою чергу до даних відкритого розділу (public) можуть звертатися будь-які функції(Рис.1.2).

## 

## Рисунок 1.1. Приховані та загальнодоступні класи

## 

## Рисунок 1.2. Синтаксис визначення класу

## Класи є головною складовою об’єктно-орієнтованого програмування. Формально клас – це визначений користувачем тип, який складається із даних полів (часто називають змінні-члени або члени-дані) та членів, які оперують цими даними (конструктори, властивості, методи, події, деструктори). Всі разом поля даних класу представляють стан екземпляра класу (інакше називають його об’єктом).

## Методологія об'єктно-орієнтованого програмування заснована на представленні програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи утворюють ієрархію спадкування.

## Об'єктно-орієнтоване програмування це узагальнення структурного і модульного програмування для оптимізації створення програм і програмних комплексів. ООП дозволяє швидко і з мінімальною кількістю помилок написати програму, код якої легко зрозуміти іншим і самому, і, отже, змінити при необхідності. Мова програмування є об'єктно-орієнтованою, коли виконуються умови: підтримуються об'єкти або абстракції даних, що мають інтерфейс у вигляді іменованих операцій та власні дані з обмеженням доступу до них; об'єкти відносяться до відповідних типів; типи (класи) можуть успадковувати атрибути супертипу (базового класу).

Класи в програмуванні складаються з властивостей (члени-дані, поля, змінні), методів (функції-члени).

Властивості - це будь-які дані, якими можна характеризувати об'єкт класу.

У нашому випадку, об'єктом класу є студент, а його властивостями - ім'я, прізвище, оцінки та середній бал.

Методи - це функції, які можуть виконувати будь-які дії над даними (властивостями) класу.

Члени класів можуть мати три рівні доступу: закритий (private), відкритий (public) чи захищений (protected).

Рівні доступу до членів класу визначають спосіб роботи користувачів із класом. Програмуючи самостійно, ви можете бути як створювачем, так й їхнім користувачем.

Можна сказати, що структура є класом, всі члени якого відкриті. Фактично, це єдина відмінність структури від класу в С#. Структура може містити функції поряд із даними. Але структура не може використовувати модифікатори доступу, оскільки вона має тільки відкритий доступ.

Всі властивості і методи класів мають права доступу. По замовчуванню, весь вміст класу є доступним для читання і запису тільки для нього самого. Для того, щоб дозволити доступ до даних класу ззовні, використовують модифікатор доступу public. Всі функції і змінні, які знаходяться після модифікатора public, стають доступними з усіх частин програми.

Закриті дані класу розміщуються після модифікатора доступу private. Якщо відсутній модифікатор public, то всі функції і змінні, за замовчуванням є закритими.

Зазвичай, приватними роблять всі властивості класу, а публічними - його методи.

Всі дії з закритими властивостями класу реалізуються через його методи.

Абстрактне представлення даних (data abstraction) – це закриття внутрішньої реалізації властивостей класу від погляду ззовні.

Програмний проект (program project) – унікальний і інтегрований комплекс взаємозалежних заходів, орієнтованих на досягнення цілей і задач об’єкта розроблення за визначеними вимогами до строків, бюджету та характеристик очікуваних результатів діяльності від нього. Спосіб виготовлення – інженерія процесу розроблення і менеджменту проекту.

1.2 Постановка задачі

1. Створити прикладне програмне забезпечення „Магазин“, в якому застосовувались би основні принципи об’єктно-орієнтованого програмування.

2. Прикладне програмне забезпечення повинне створюватись на мові програмування Microsoft Visual Studio C#.

3. Додаток повинен виводити на екран категорії товарів і давати змогу порахувати ціни вибраних товарів.

4. По завершенні програми має виводитися ціни вибраних товарів.

# Проектування структури додатку, діаграма класів

Структура нашого додатку складається із базового класу Magazin та похідних класів Instrument, Produktu,Pobut.

Зв’язок між базовим класом Magazin та похідними класами можна спостерігати на діаграмі класів додатку „Магазин“(Рис.1.3).

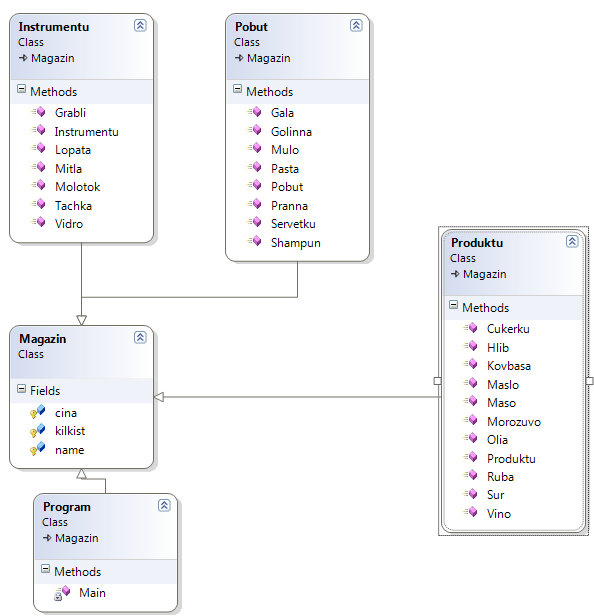


Рисунок 1.3. Діаграма класів

# **Розділ 2.** Опис розробки програми

2.1 Опис алгоритму  
програми та розробка програмного коду

1. Заходимо у Microsoft Visual Studio(Рис.2.1):

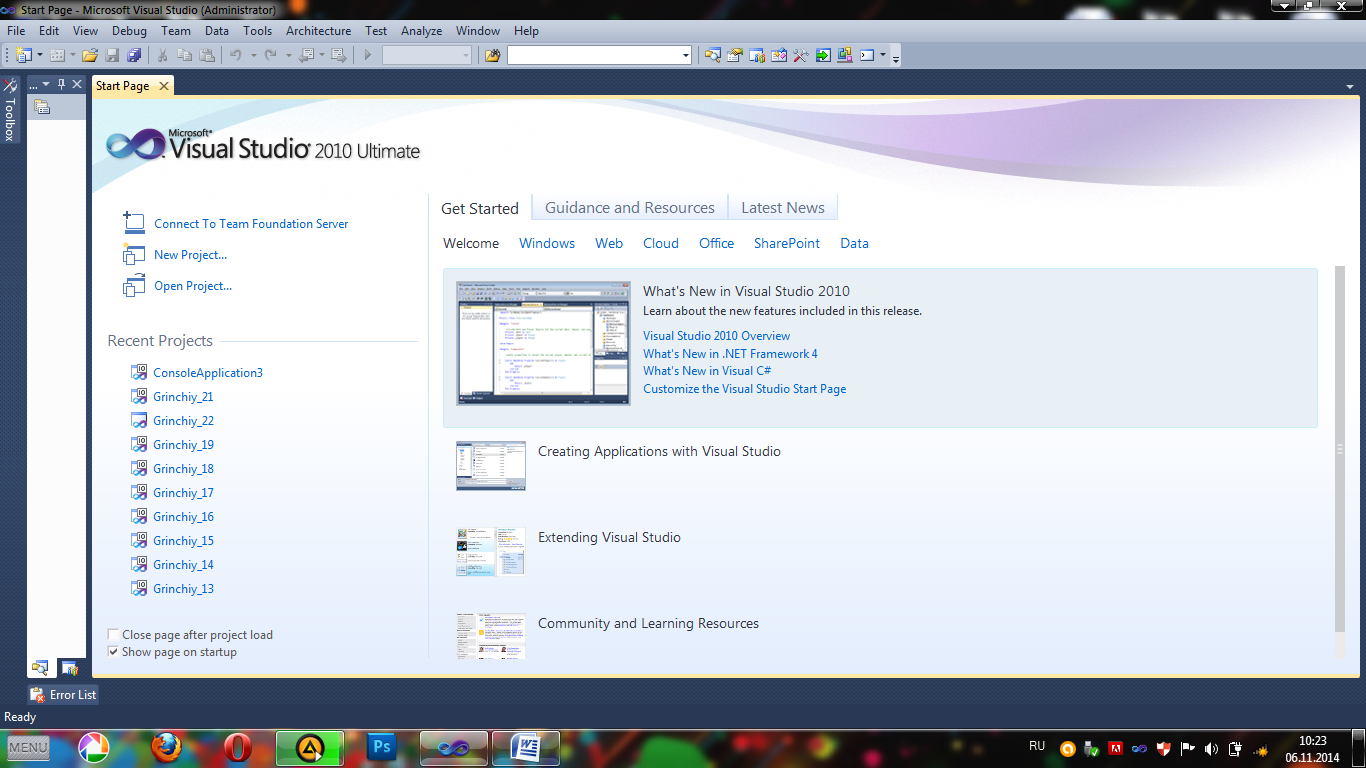


Рисунок 2.1. Вікно програми

1. Створюємо новий проект і даємо йому назву Program.cs(Рис.2.2, 2.3)

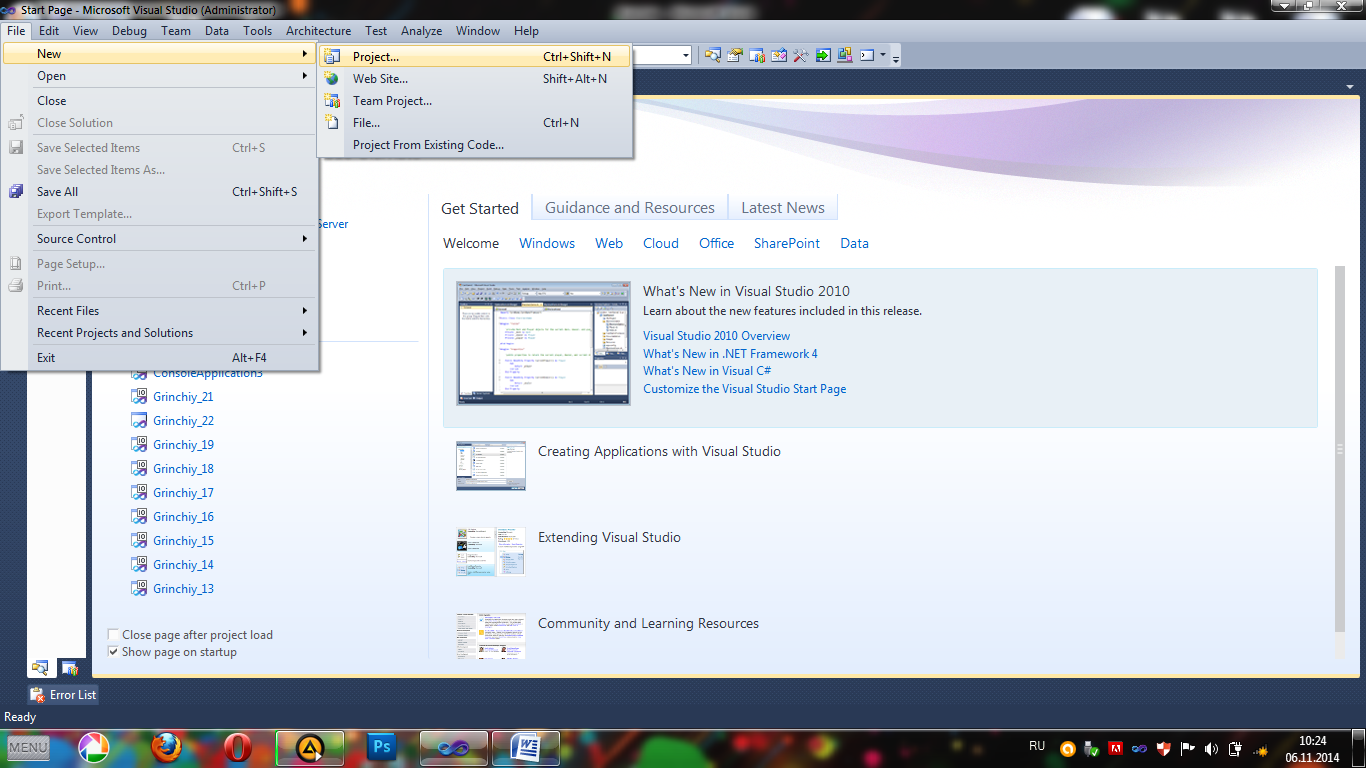


Рисунок 2.2. Вікно створення проекту

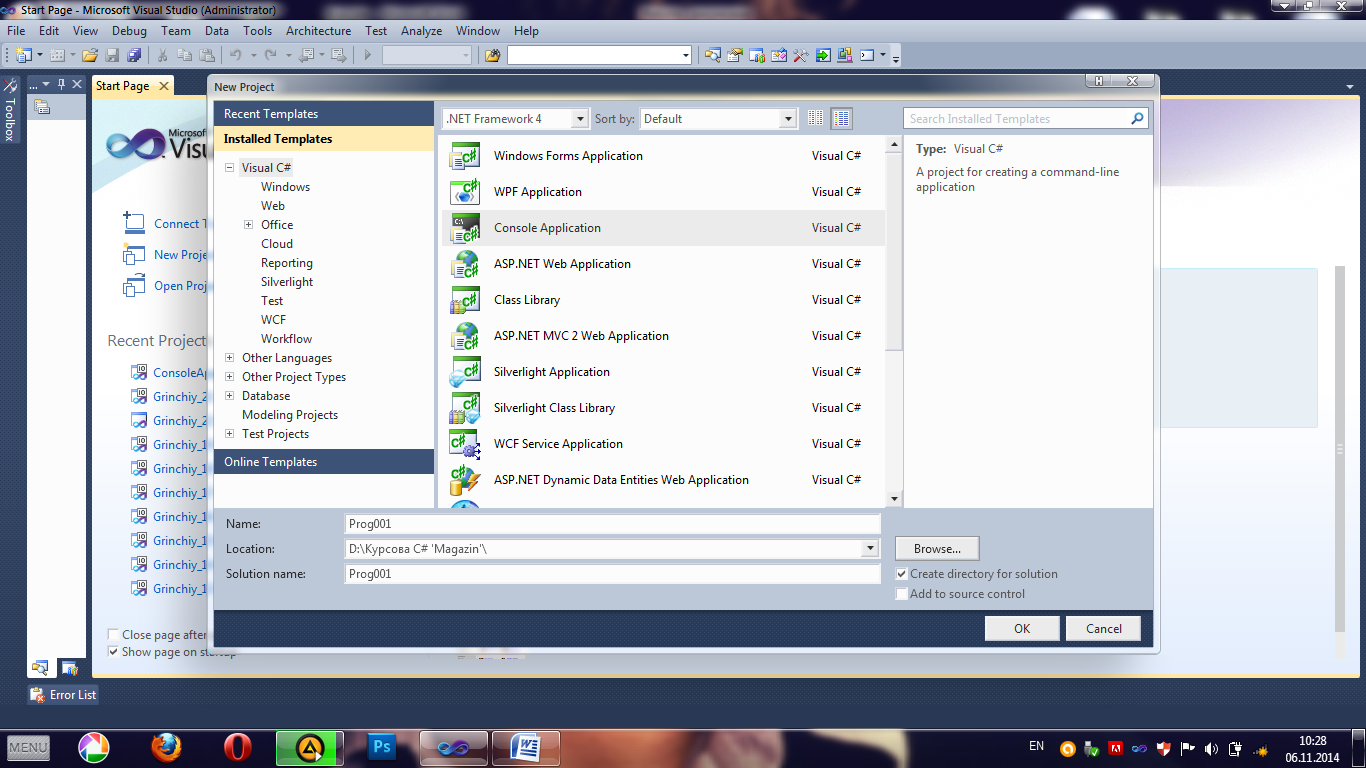


Рисунок 2.3. Вікно створення консольного додатку

Пишемо в ньому програмний код:

Див. додаток 1.

3.Створюємо новий клас і даємо йому назву Magazin.cs(Рис.2.4)

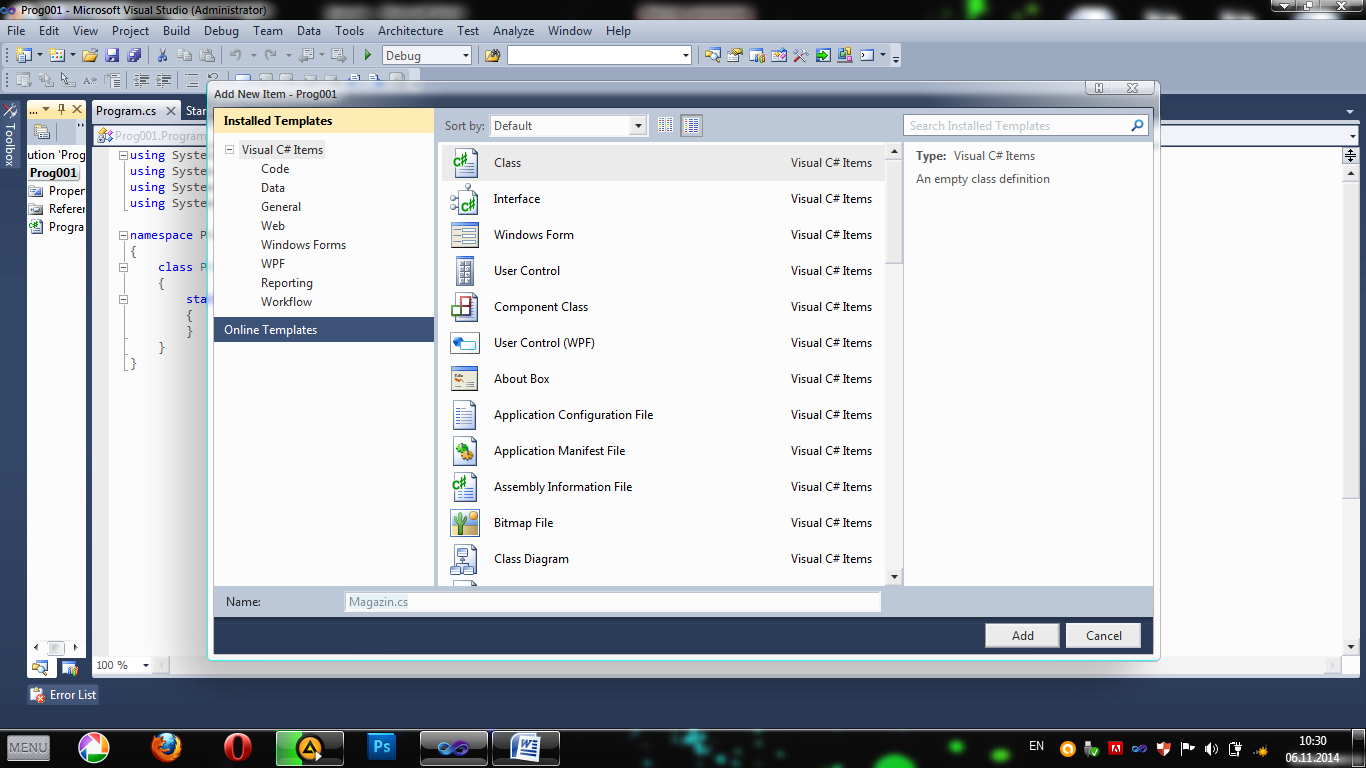


Рисунок 2.4. Вікно створення нового класу і його назви

Пишемо в ньому програмний код:

Див. додаток 2.

4.Створюємо новий клас і даємо йому назву Productu.cs(Рис.2.5)

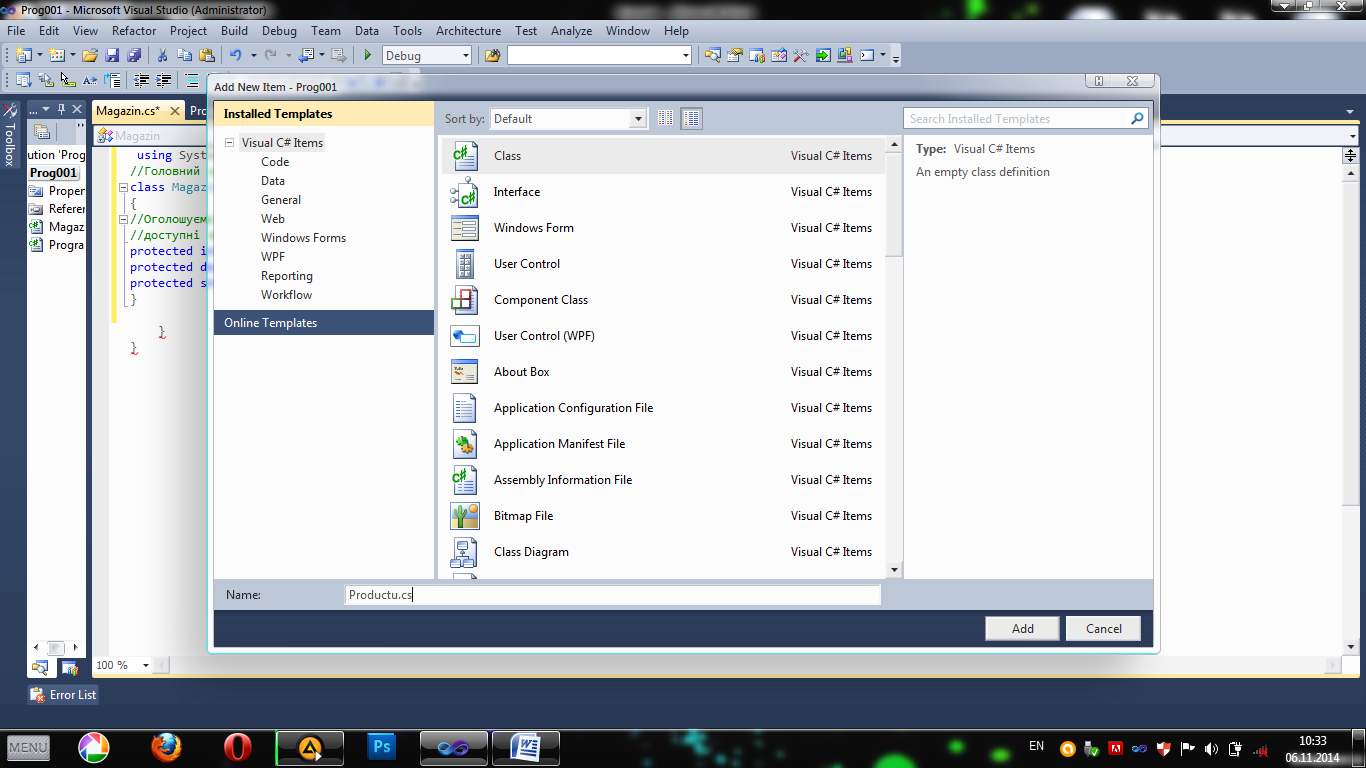


Рисунок 2.5. Вікно створення нового класу і його назви

Пишемо в ньому програмний код:

Див. додаток 3.

5.Створюємо новий клас і даємо йому назву Pobut.cs(Рис.2.6)

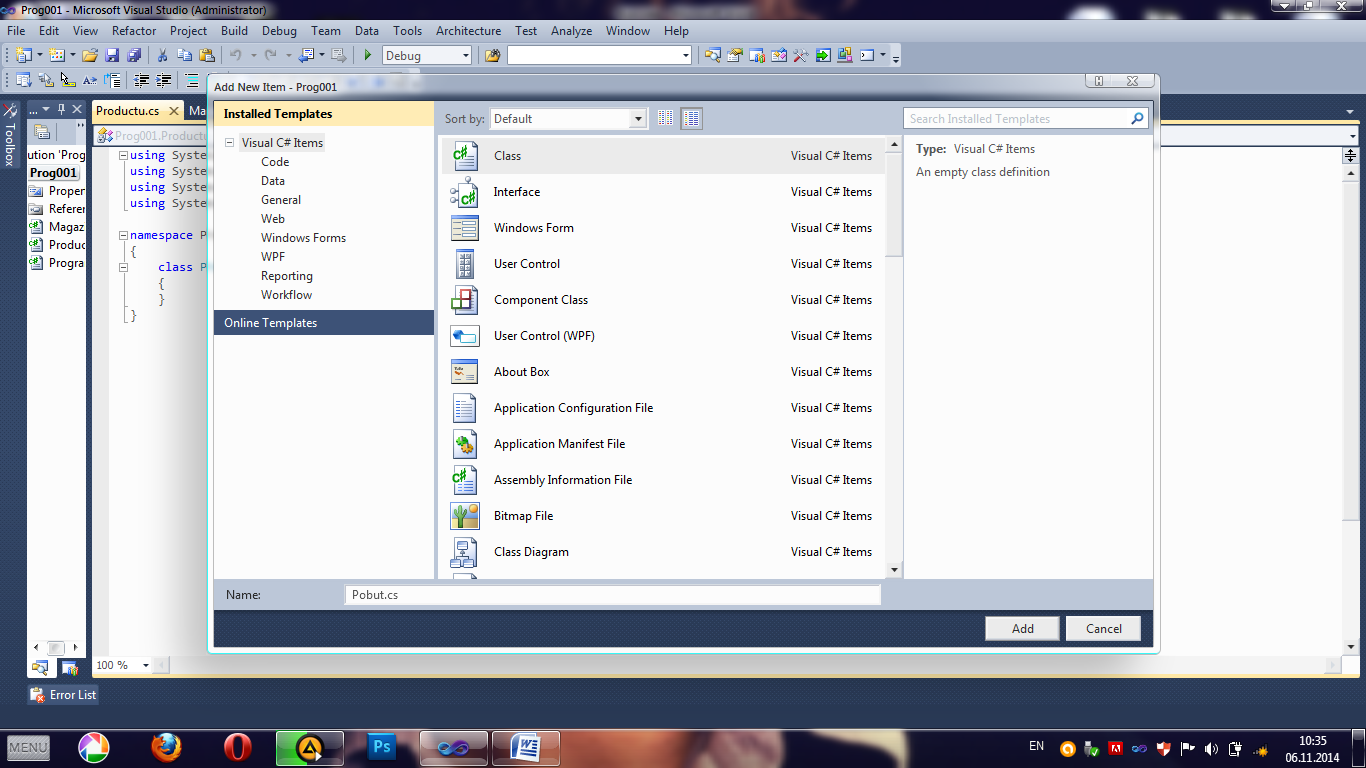


Рисунок 2.6. Вікно створення нового класу і його назви

Пишемо в ньому програмний код:

Див. додаток 4.

6.Створюємо новий клас і даємо йому назву Instrument.cs(Рис.2.7)

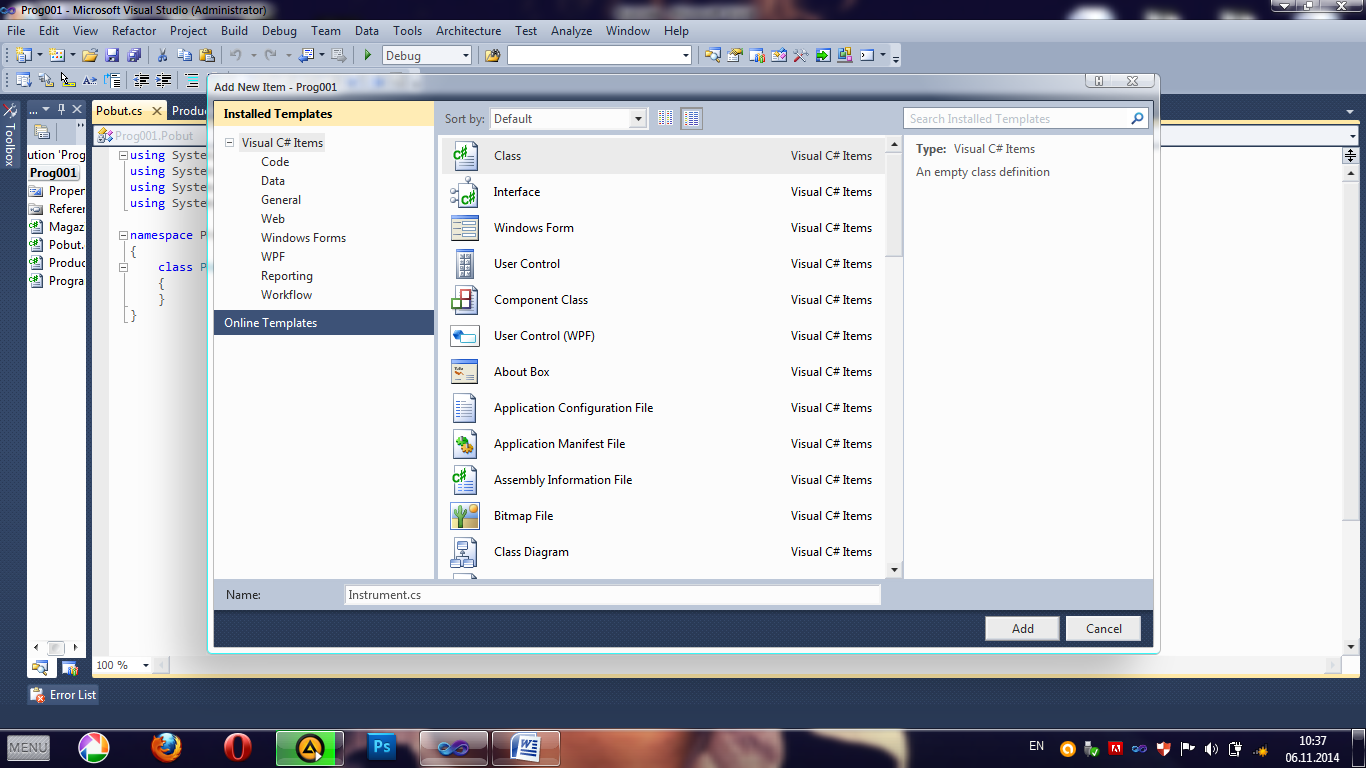


Рисунок 2.7. Вікно створення нового класу і його назви

Пишемо в ньому програмний код:

Див. додаток 5.

Тестування додатку

Запустимо програму:

Див. Рис.2.8-2.12.

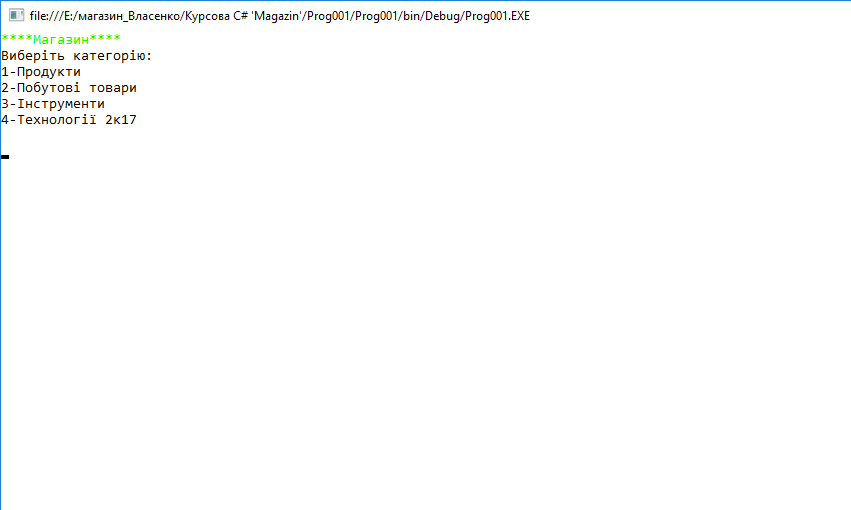


Рисунок 2.8. Початкове вікно програми «Магазин».

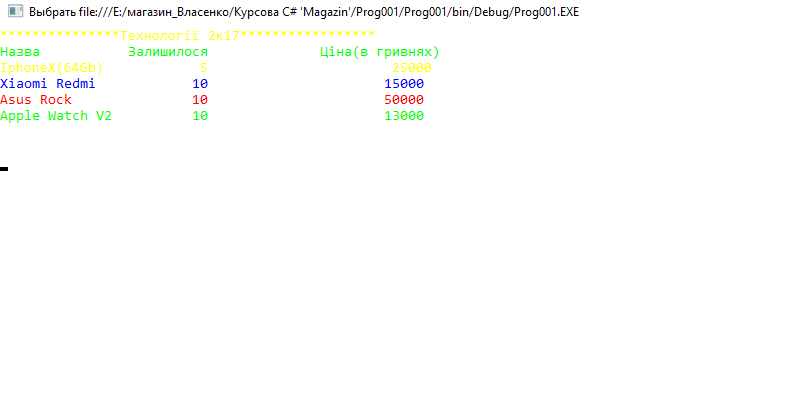


Рисунок 2.9. Вікно програми після вибору категорії

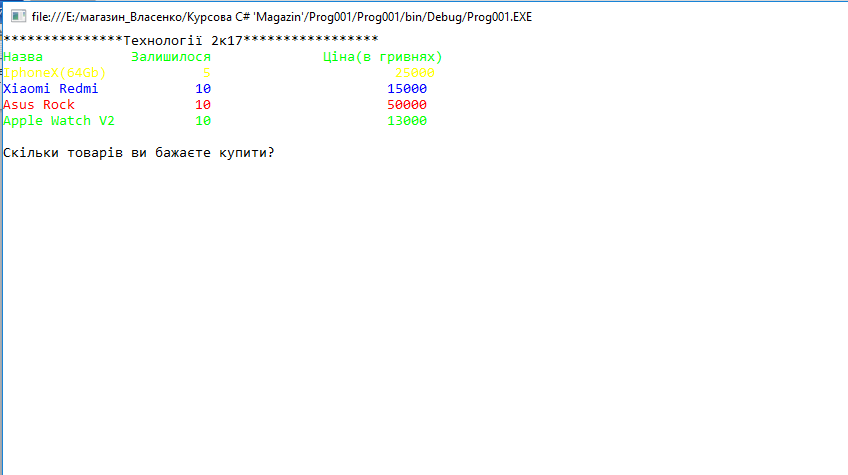


Рисунок 2.10. Вікно програми для вибору товарів

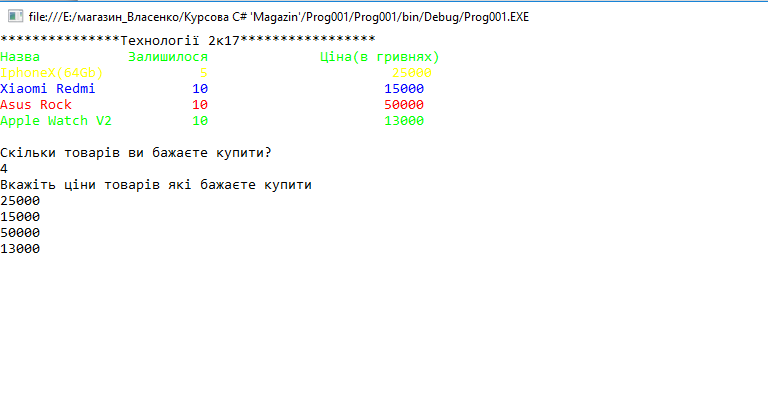


Рисунок 2.11. Вікно програми після вибору товарів

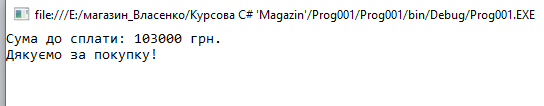


Рисунок 2.12. Кінцеве вікно програми

## 2.2 Опис використання програмних засобів об’єктно-орієнтованого програмування

Основні положення об'єктно-орієнтованого програмування: інкапсуляція; спадкування; поліморфізм.

Інкапсуляція – це механізм, який об’єднує дані і код, який маніпулює цими даними, а також захищає дані і код від стороннього втручання. Інкапсуляція – приховування даних. Інкапсуляція означає, що зовнішнім об’єктам і користувачу прямий доступ до даних заборонено. Це робиться для підтримки цілісності даних і їх несуперечливості(Рис.2.13).

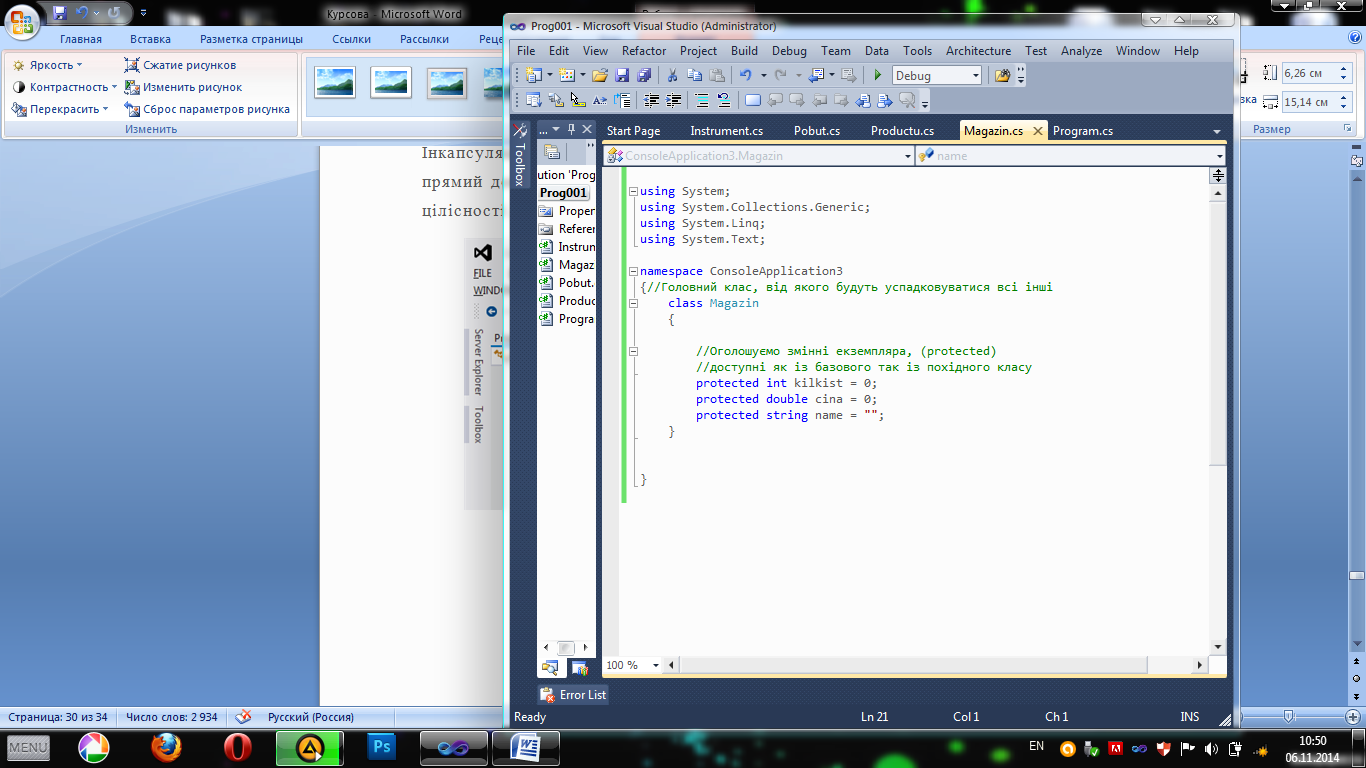


Рисунок 2.13. Частина програмного коду «Магазин».

Спадкування – це процес, в результаті якого один об’єкт може набути властивості іншого. Це означає, що в ООП на основі уже існуючих класів можна будувати похідні класи. При цьому похідний клас (клас-нащадок) наслідує дані-елементи і функції-елементи від своїх батьківських класів (класів-предків), добавляючи до них свої, характерні тільки йому, властивості та методи. Захищені дані-елементи і функції-елементи батьківського класу доступні із похідного класу. Крім того, у похідного класу наслідувані функції можуть бути перевизначені. Таким чином можна побудувати цілу ієрархію класів, зв’язаних між собою відношенням батько – нащадок. Термін базовий клас використовується як синонім батьківському класу в ієрархії класів. Якщо об’єкт наслідує свої атрибути (властивості та методи) від одного батьківського класу, то говорять про просте наслідування. Якщо об’єкт наслідує атрибути від декількох батьківських класів, - множинне наслідування. Наслідування дозволяє в значній мірі скороти визначення класу-нащадка через те, що класи-нащадки являються розширенням батьківських класів.

Основні концепції наслідування

• Якщо програми використовують успадкування, то для породження нового класу необхідний базовий клас, тобто новий клас успадковує елементи уже існуючого базового класу.

• Для ініціалізації елементів похідного класу програма повинна викликати конструктори базового і похідного класів.

• Використовуючи оператор "крапку", програми можуть легко звертатися до елементів базового і похідного класів.

• На додаток до загальних (public) (доступним всім) і приватних (private) (доступним методам класу) елементів C++ надає захищені (protected) елементи, які доступні базовому і похідному класам.

• Для вирішення конфлікту імен між елементами базового і похідного класів програма може використовувати оператор глобального дозволу, указуючи перед ним ім'я базового або похідного класу.

При успадкуванні, за необхідності, можна перевизначити деякі функції батьківського класу в класі-нащадку. Звичайно у базовому класі описуються найбільш загальні властивості, які придатні для всіх похідних від нього класів. Похідний клас успадковує ці загальні властивості і додає нові, характерні тільки для нього.

Простим називається спадкування, при якому похідний клас або клас-нащадок має один базовий клас або одного батька. Для різних методів класу існують різні правила спадкування — наприклад, конструктори, деструктори та операція присвоювання в похідному класі не спадкуються.

***Поля і методи при спадкуванні***

* Клас-нащадок успадковує всі поля та методи батьківського класу.
* Якщо у батьківському класі поле чи метод приватний, то нащадок не має до нього доступу.
* Допускається не тільки успадкування методів базового класу, але також додавання нових і перевизначення існуючих методів.
* Якщо ім’я поля (методу) у похідному і базовому класі співпадають, говорять про перевизначення або перекриття. Для звернення до змінної базового класу використовують операцію доступу до області видимості "**::"**(Рис.2.14)**.**

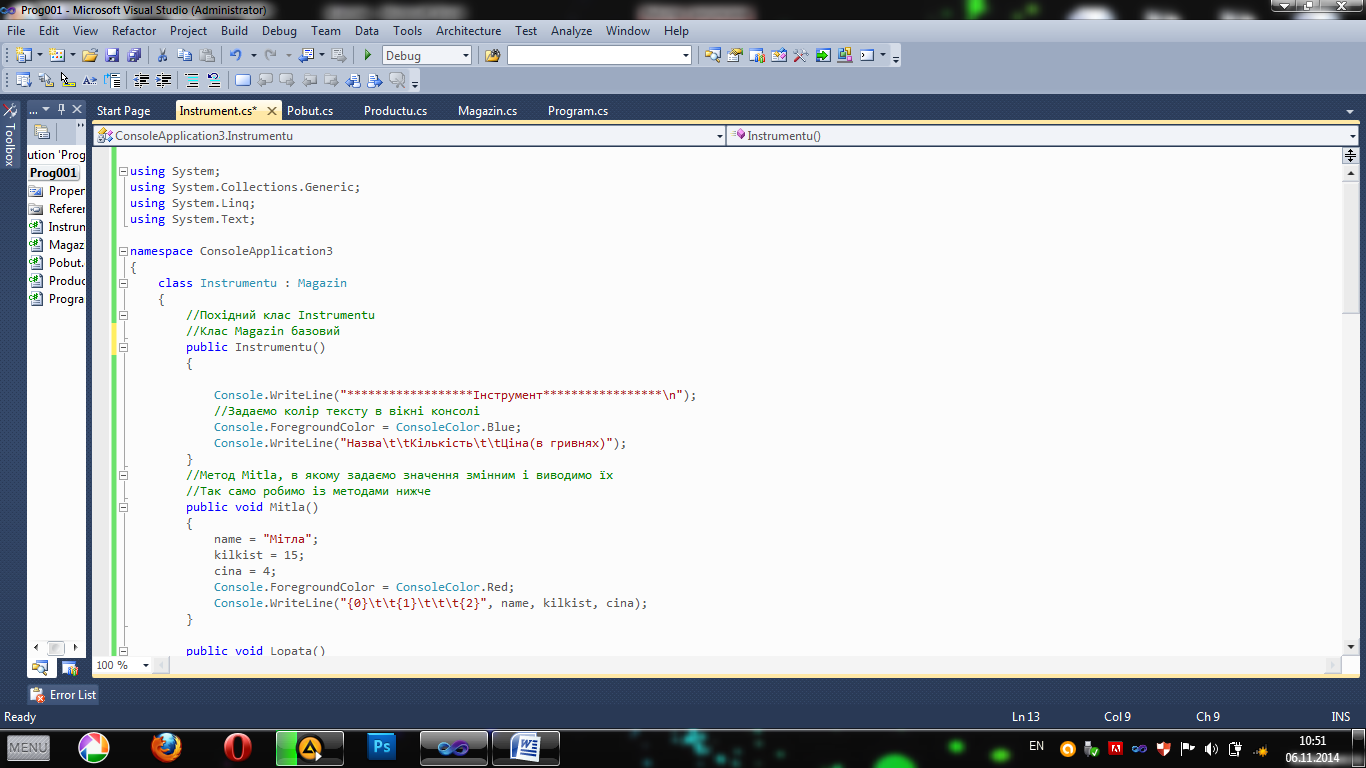


Рисунок 2.14. Клас Magazin з коду програми «Магазин»

Поліморфізм (з грецької «polymorphos» - множинність форм) – це властивість коду вести себе по-різному в залежності від ситуації, яка виникає в момент виконання. Поліморфізм – це не стільки характеристика об’єктів, скільки характеристика функцій-членів класу і проявляється, зокрема, у можливості використання одного імені функції-члена для функцій, які мають різні типи аргументів, якщо виконувані функцією дії визначаються типом переданих їй аргументів. Це називається перевантаження функції.

Поліморфізм може застосовуватись і до операцій. Тобто виконувані операцією дії також можуть залежати від типу даних (операндів). Такий тип поліморфізму називається перезавантаженням операції.

**Висновок**

У результаті виконання курсової роботи мною було опрацьовано ряд джерел інформації, одержані більш глибокі знання з мови програмування Microsoft Visual Studio C#.

В результаті виконання курсової роботи був створений додаток Магазин. В ньому застосовуються головні програмні засоби об’єктно-орієнтованого програмування. А саме: поліморфізм, ікапсуляція, наслідування. Додаток виконує всі поставлені до нього задачі.

Було детально вивчено можливості Microsoft Visual Studio C# щодо реалізації основних принципів об’єктно-орієнтованого програмування.

В результаті проведеного тестування можна переконатися, що додаток „Магазин“ є повністю функціональним, не містить помилок і може в повній мірі використовуватися.

Процес написання курсової роботи дав додаткові теоретичні та практичні навички у застосуванні мови програмування Microsoft Visual Studio C# для використання цього програмного продукту у своїй професійній роботі.

Процес написання курсової роботи дав додаткові теоретичні та практичні навички у застосуванні мови програмування Microsoft Visual Studio C# для використання цього програмного продукту у своїй професійній роботі.

# **С****писок використаної літератури**

1. ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам.
2. ДСТУ 1.5-93 Загальні вимоги до побудови, викладу, оформлення та змісту.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С# . [2-е изд.] / Буч Г. - СПб.: Невский Диалект, 2018. - 560 с.
4. Ковалюк Т. В. Алгоритмізація та програмування: Підручник. — Львів: «Магнолія 2016», 2016. — 400 с., ил.
5. Лавріщева К.М. Програмна інженерія. – Підручник.–К.:Академперіодика, 2008.–415с.
6. Лахатин, А.С. Мови програмування. Навч. посібник / А.С. Лахатин, Л.Ю. Іскакова. - Кропивницький, 2018. - 548с.
7. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Харків, 2013. – 461 с. URL.
8. Уейт М. Мова С. Керівництво для початківців. / М. Уейт, С. Прага, Д. Мартин. - М.: Мир, 2015. - 521с.
9. Фридман, А.Л. Мова програмування С++ / А.Л.Фридман. - М.: Бином, 2016. - 523с.
10. <http://dit.isuct.ru/IVT/BOOKS/DBMS/DBMS6/INDEX.HTM>
11. <http://www.mysql.ru/docs/man/Features.html>
12. <https://uk.wikibooks.org/wiki/SQL/%D0%A2%D0%B88%D1%85_MySQL>
13. <https://docs.microsoft.com/ru-ru/dotnet/csharp/getting-started/>
14. <https://techrocks.ru/2018/07/29/programming-languages-popularity-2018/>
15. <http://apeps.kpi.ua/top10-mov-programuvania-2019>
16. <http://www.sql-tutorial.ru/ru/book_delete_statement.html>
17. <http://www.rldp.ru/mysql/mysql80/index.htm>

# **Додаток 1**

using System;

namespace ConsoleApplication1{

classProgram:Magazin{

staticvoid Main(string[] args) {

link1://Мітка для goto

Console.ForegroundColor = ConsoleColor.Red;

//Задаємо колір тексту в вікні консолі

Console.WriteLine("--------------------Програма Магазин----------------");

//Задаємо колір тексту в вікні консолі

Console.ForegroundColor = ConsoleColor.Yellow;

//Виводимо категорії в меню

Console.WriteLine("Виберiть категорiю:\n"+

"1-Продукти\n"+

"2-Побутовi товари\n"+

"3-Iнструмент\n");

string v = Console.ReadLine();//Змінна вибору

Console.Clear();//Очищає вікно консолі

//Перевіряє що введено

if (v != "1"&& v != "2"&& v != "3" && v!= ‘4’) {

Console.WriteLine("Введiть те, що вказано на екранi!");

goto link1;}

//В залежності від вибіру виводить варіанти

if (v == "1") {

Produktu o1 = newProduktu();//Посилання на об’єкт

o1.Hlib();

o1.Maslo();

o1.Kovbasa();

o1.Morozuvo();

o1.Maso();

o1.Ruba();

o1.Olia();

o1.Vino();

o1.Cukerku();

o1.Sur();}

if (v == "2") {

Pobut o2 = newPobut();

o2.Gala();

o2.Pranna();

o2.Shampun();

o2.Mulo();

o2.Pasta();

o2.Golinna();

o2.Servetku();}

if (v == "3") {

Instrument o3 = newInstrument();//Посилання на о’єкт

o3.Grabli();

o3.Lopata();

o3.Mitla();

o3.Molotok();

o3.Tachka();

o3.Vidro(); }

Console.WriteLine("Скiльки товарiв ви бажаєте купити?");

//Оголошуємо змінні

int comp = 1;

int comp1 = 2;

int comp2 = 3;

int comp3 = 4;

int comp4 = 5;

int comp5 = 6;

int comp6 = 7;

int comp7 = 8;

int comp8 = 9;

int comp9 = 10;

int x = 0;

int c = 0;

//конвертація в цілочисловий тип

x = Convert.ToInt32(Console.ReadLine());

//В залежності від вибору маємо наступний розвиток подій

if (x == comp) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c1);

Console.Write("\nДякуємо за покупку!"); }

elseif (x == comp1) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!\n");

goto link1; }

elseif (x == comp2) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!\n");

goto link1; }

elseif (x == comp3) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!\n"); }

elseif (x == comp4) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!"); }

elseif (x == comp5) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

int c6 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5 + c6;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!"); }

else if (x == comp6) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

int c6 = Convert.ToInt32(Console.ReadLine());

int c7 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5 + c6 + c7;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!"); }

else if (x == comp7) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

int c6 = Convert.ToInt32(Console.ReadLine());

int c7 = Convert.ToInt32(Console.ReadLine());

int c8 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!"); }

else if (x == comp8) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

int c6 = Convert.ToInt32(Console.ReadLine());

int c7 = Convert.ToInt32(Console.ReadLine());

int c8 = Convert.ToInt32(Console.ReadLine());

int c9 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8 + c9;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!"); }

else if (x == comp9) {

Console.WriteLine("Вкажiть цiни товарiв якi бажаєте купити");

int c1 = Convert.ToInt32(Console.ReadLine());

int c2 = Convert.ToInt32(Console.ReadLine());

int c3 = Convert.ToInt32(Console.ReadLine());

int c4 = Convert.ToInt32(Console.ReadLine());

int c5 = Convert.ToInt32(Console.ReadLine());

int c6 = Convert.ToInt32(Console.ReadLine());

int c7 = Convert.ToInt32(Console.ReadLine());

int c8 = Convert.ToInt32(Console.ReadLine());

int c9 = Convert.ToInt32(Console.ReadLine());

int c10 = Convert.ToInt32(Console.ReadLine());

c = c1 + c2 + c3 + c4 + c5 + c6 + c7 + c8 + c9 + c10;

Console.Clear();

Console.Write("Сума до сплати: {0} грн.", c);

Console.Write("\nДякуємо за покупку!");}

Console.ReadLine();//Затримка на екрані консолі}}}

Додаток 2

using System;

//Головний клас, від якого будуть успадковуватися всі інші

classMagazin{

//Оголошуємо змінні екземпляра, (protected)

//доступні як із базового так із похідного класу

protectedint kilkist = 0;

protecteddouble cina = 0;

protectedstring name = "";}

Додаток 3

using System;

//Похідний клас Productu

//Клас Magazin базовий

classProduktu:Magazin{

public Produktu(){

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Продукти\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

//Задаємо колір тексту в вікні консолі

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Назва\t\tКiлькiсть\t\tЦiна(в гривнях)");}

//Метод Hlib, в якому задаємо значення змінним і виводимо їх

//Так само робимо із методами нижче

public void Hlib() {

name = "Хлiб";

kilkist = 15;

cina = 4;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina);}

public void Maslo() {

name = "Масло";

kilkist = 10;

cina = 9;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} почтак\t\t{2}", name, kilkist, cina);}

public void Kovbasa() {

name = "Ковбаса";

kilkist = 7;

cina = 50;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina);}

public void Sur() {

name = "Сир";

kilkist = 25;

cina = 57;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina);}

public void Olia(){

name = "Олiя";

kilkist = 15;

cina = 16;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} л.\t\t\t{2}", name, kilkist, cina);}

public void Vino(){

name = "Вино";

kilkist = 10;

cina = 77;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} пляшок\t\t{2}", name, kilkist, cina);}

public void Morozuvo() {

name = "Морозиво";

kilkist = 10;

cina = 2;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t{1} пачок\t\t{2}", name, kilkist, cina);}

public void Cukerku() {

name = "Цукерки";

kilkist = 25;

cina = 57;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina);}

public void Ruba() {

name = "Риба";

kilkist = 11;

cina = 150;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina); }

public void Maslo() {

name = "М’ясо";

kilkist = 27;

cina = 55;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina); }

}

Додаток 4

using System;

//Похідний клас Pobut

//Клас Magazin базовий

Class Pobut : Magazin{

//Конструктор класу

public Pobut() {

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Побутовi товари\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

//Задаємо колір тексту в вікні консолі

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Назва\t\tКiлькiсть\t\tЦiна(в гривнях)"); }

//Метод Gala, в якому задаємо значення змінним і виводимо їх

//Так само робимо із методами нижче

publicvoid Gala() {

name = "Миючий засiб";

kilkist = 10;

cina = 7;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t{1}\t\t\t{2}", name, kilkist, cina); }

publicvoid Pranna() {

name = "Пральний порошок";

kilkist = 25;

cina = 10;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0} {1} пачок\t\t{2}", name, kilkist, cina);}

publicvoid Shampun() {

name = "Шампунь";

kilkist = 15;

cina =11;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina);}

publicvoid Mulo() {

name = "Мило";

kilkist = 11;

cina = 7;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} кг.\t\t\t{2}", name, kilkist, cina); }

publicvoid Pasta() {

name = "Зубна паста";

kilkist = 28;

cina = 25;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t{1} л.\t\t\t{2}", name, kilkist, cina); }

publicvoid Golinna() {

name = "Гель для голiння";

kilkist = 10;

cina =14;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0} {1} пляшок\t\t{2}", name, kilkist, cina);}

publicvoid Servetku() {

name = "Серветки";

kilkist = 10;

cina = 3;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t{1} пачок\t\t{2}", name, kilkist, cina); }

}

Додаток 5

using System;

//Похідний клас Instrument

//Клас Magazin базовий

classInstrument : Magazin{

public Instrument() {

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Iнструмент\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");//Задаємо колір тексту в вікні консолі

Console.ForegroundColor = ConsoleColor.Blue;

Console.WriteLine("Назва\t\tКiлькiсть\t\tЦiна(в гривнях)");}

//Метод Mitla, в якому задаємо значення змінним і виводимо їх

//Так само робимо із методами нижче

publicvoid Mitla() {

name = "Мiтла";

kilkist = 15;

cina = 4;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina); }

publicvoid Lopata() {

name = "Лопата";

kilkist = 10;

cina = 9;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina); }

publicvoid Vidro() {

name = "Вiдро";

kilkist = 7;

cina = 50;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina); }

publicvoid Grabli() {

name = "Граблi";

kilkist = 25;

cina = 57;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina);}

publicvoid Tachka() {

name = "Тачка";

kilkist = 15;

cina = 16;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1}\t\t\t{2}", name, kilkist, cina); }

publicvoid Molotok() {

name = "Молоток";

kilkist = 10;

cina = 77;

Console.ForegroundColor = ConsoleColor.Red;

Console.WriteLine("{0}\t\t{1} \t\t\t{2}", name, kilkist, cina); }}