

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

КУРСОВА РОБОТА

з дисципліни «Програмування»

на тему

Шаблони проєктування в ООП. Фруктовий сад

Виконав студент

II курсу групи КП-11

Лунев Артем Вадимович

Керівник роботи

доцент, к.т.н. Заболотня Т.М.

Оцінка

(дата, підпис)

КИЇВ 2023

ЗМІСТ

ЗМІСТ	2
ВСТУП	3
1. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ФРУКТОВОГО САДУ	5
1.1. Модульна організація програми	5
1.2. Функціональні характеристики.....	6
2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЄКТУВАННЯ	7
2.1. Обґрунтування вибору та опис шаблонів проєктування для програмного забезпечення фруктового саду	7
2.2. Діаграма класів	15
2.3. Опис результатів роботи програми	16
ВИСНОВКИ.....	20
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	21

ВСТУП

Цей курсовий проєкт присвячений розробці програмного забезпечення для управління фруктовим садом. Метою проєкту є створення програми, яка дозволить користувачам працювати з об'єктами, такими як садівник, сад, дерево та фрукт, забезпечуючи повний цикл догляду за фруктовим садом.

Об'єктом дослідження є процес догляду за фруктовим садом.

Метою роботи є розроблення програмного забезпечення фруктового саду з використанням шаблонів проєктування.

Для досягнення визначеної мети необхідно виконати такі *завдання*:

- абстрагувати об'єкти предметної галузі;
- розробити структурну організацію ПЗ за допомогою застосування основних принципів ООП та шаблонів проєктування;
- визначити та описати функціональні характеристики програми;
- обґрунтувати вибір шаблонів проєктування, використаних для побудови програми;
- розробити користувацький консольний інтерфейс;
- виконати реалізацію програмного забезпечення відповідно до вимог технічного завдання;
- виконати тестування розробленої програми;
- оформити документацію з курсової роботи.

Розроблене ПЗ фруктового саду складається з трьох модулів: модуль взаємодії з користувачем, модуль огляду всіх садів, модуль догляду за конкретним садом модуль перегляду статистики.

Реалізовані шаблони проєктування: Абстрактна фабрика, Фабричний метод, Стан, Шаблонний метод, Замісник, Спостерігач.

До функціональних можливостей програми належать: перевірки коректності вхідних даних, надання інформації щодо створених садів, збереження поточних станів садів та статистики врожаю.

Для функціонування розробленої програми необхідно забезпечити

наявність на комп'ютері 20 Мб вільного дискового простору та встановленого Python.

Розроблене програмне забезпечення може бути використане будь-ким, хто обожнює рослинність.

Пояснювальна записка складається зі вступу, двох розділів, загальних висновків, списку використаних джерел. Робота містить 20 рисунків. Загальний обсяг роботи – 27 друкованих сторінок, з них 24 сторінки основного тексту та 1 сторінка списку використаних джерел.

1. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ФРУКТОВОГО САДУ

1.1. Модульна організація програми

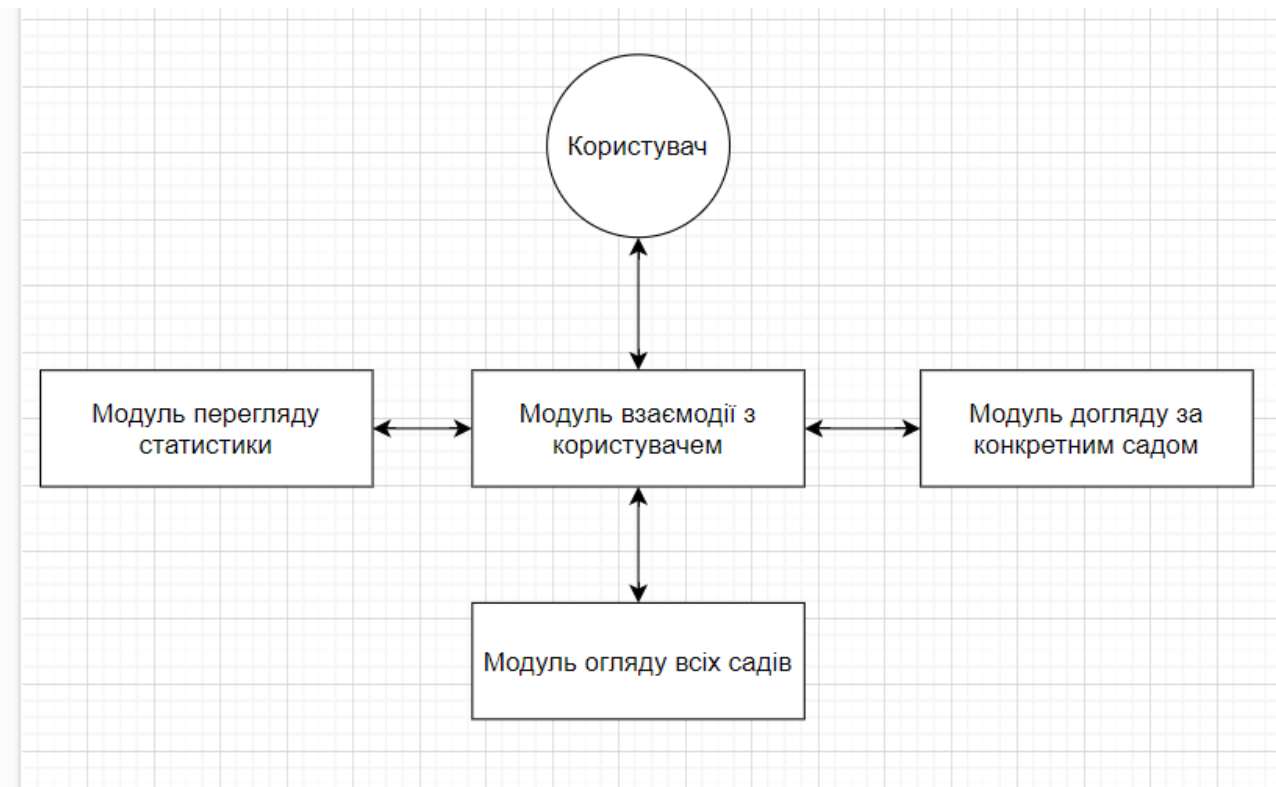


Рис. 1.1.1. Модульна організація програми

Програма може бути розділена на кілька модулів, які взаємодіють між собою та обмінюються даними. Взаємодія між модулями відбувається за допомогою виклику методів та відправлення подій.

Модуль взаємодії з даними користувача виступає посередником між користувачем та іншими функціями системи. Він має простий консольний інтерфейс, який отримує команди від користувача, перевіряє їх коректність та викликає відповідні методи інших модулів. В цьому модулі також отримуються дані від інших модулів та виводяться користувачу. Така архітектура дозволяє забезпечити гнучкість та розширюваність програми, оскільки кожен модуль відповідає за певну логічну частину функціональності і може бути змінений або розширений незалежно від інших модулів.

Модуль взаємодії даних конкретного саду відповідає за догляд над конкретним садом. У цьому модулі виконуються всі можливі дії над садом.

Модуль огляду даних про всі сади відповідає за показ користувачу усіх його створених садів. Цей модуль відображає всі посаджені дерева та їхні характеристики.

Модуль перегляду статистики безпосередньо представляє собою перелік команд для виведення статистики того чи іншого саду, а саме скільки фруктів було зібрано, а скільки впало й не було зібрано.

1.2. Функціональні характеристики

Розроблене ПЗ дозволяє користувачу зручно керувати процесом догляду за допомогою консольного інтерфейсу. При вході в додаток користувач може обрати, за яким садом доглядати. Після цього йому доступні такі можливості:

- перегляд даних про посаджені дерева, їхні характеристики;
- можливість полити дерева або знищити шкідників;
- зібрати врожай;
- переглянути статистику.

2. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЄКТУВАННЯ

2.1. Обґрунтування вибору та опис шаблонів проєктування для програмного забезпечення фруктового саду

1) Фабричний метод + Шаблонний метод

Визначення:

Фабричний метод — це породжувальний патерн проєктування, який визначає загальний інтерфейс для створення об'єктів у суперкласі, дозволяючи підкласам змінювати тип створюваних об'єктів.

Шаблонний метод — це поведінковий патерн проєктування, який визначає кістяк алгоритму, перекладаючи відповідальність за деякі його кроки на підкласи. Патерн дозволяє підкласам перевизначати кроки алгоритму, не змінюючи його загальної структури.

Обґрунтування використання шаблону:

Фабричний метод допомагає з організацією великої кількості дрібних підкласів різних типів дерев.

Шаблонний метод допомагає легко розбити процес створення дерева на декілька кроків для того, щоб класи творці могли виділити особливості створення своїх продуктів без повторного переписування методу.

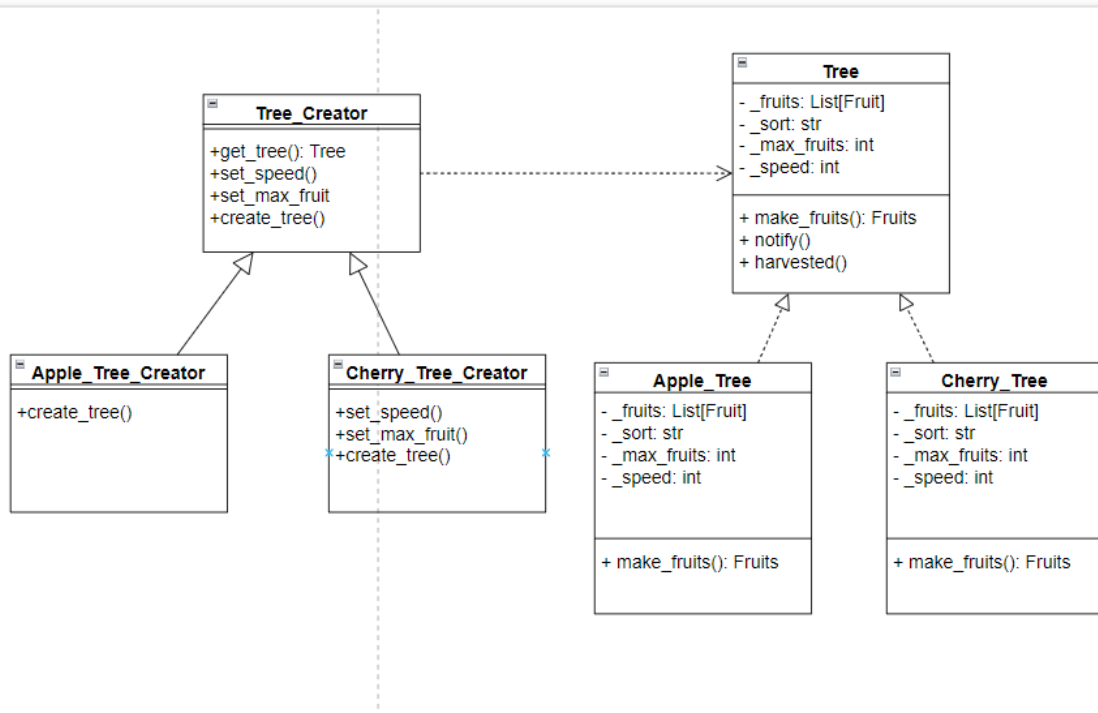


Рис. 2.1.1. UML діаграма класів, які входять до шаблону “Фабричний метод” і “Шаблонний метод”

Учасники шаблону:

- **AbstractClass, Creator** (Tree_Creator) – оголошує інтерфейс для операцій, що створюють конкретні творці;
- **ConcreteCreatorA, B** (Apple_Tree_Creator, Cherry_Tree_Creator) – реалізує методи творця, модифікує їх;
- **AbstractProduct** (Tree) – оголошує інтерфейс для дерев;
- **ConcreteProduct** (Apple_Tree, Cherry_Tree) – визначає дерево певного типу.

2) Стан + Спостерігач

Визначення:

Стан — це поведінковий патерн проектування, що дає змогу об'єктам змінювати поведінку в залежності від їхнього стану. Ззовні створюється враження, ніби змінився клас об'єкта.

Спостерігач — це поведінковий патерн проектування, який створює механізм підписки, що дає змогу одним об'єктам стежити й реагувати на події, які відбуваються в інших об'єктах.

Обґрунтування використання шаблону:

Спостерігач допомагає керувати великою кількістю фруктів, які підв'язані до одного дерева. Якщо з деревом щось відбувається, то це буде впливати на всі фрукти на ньому. Стан допоможе з внутрішньою складовою кожного фрукту, щоб можна було легше визначати що зараз відбувається з кожним фруктом.

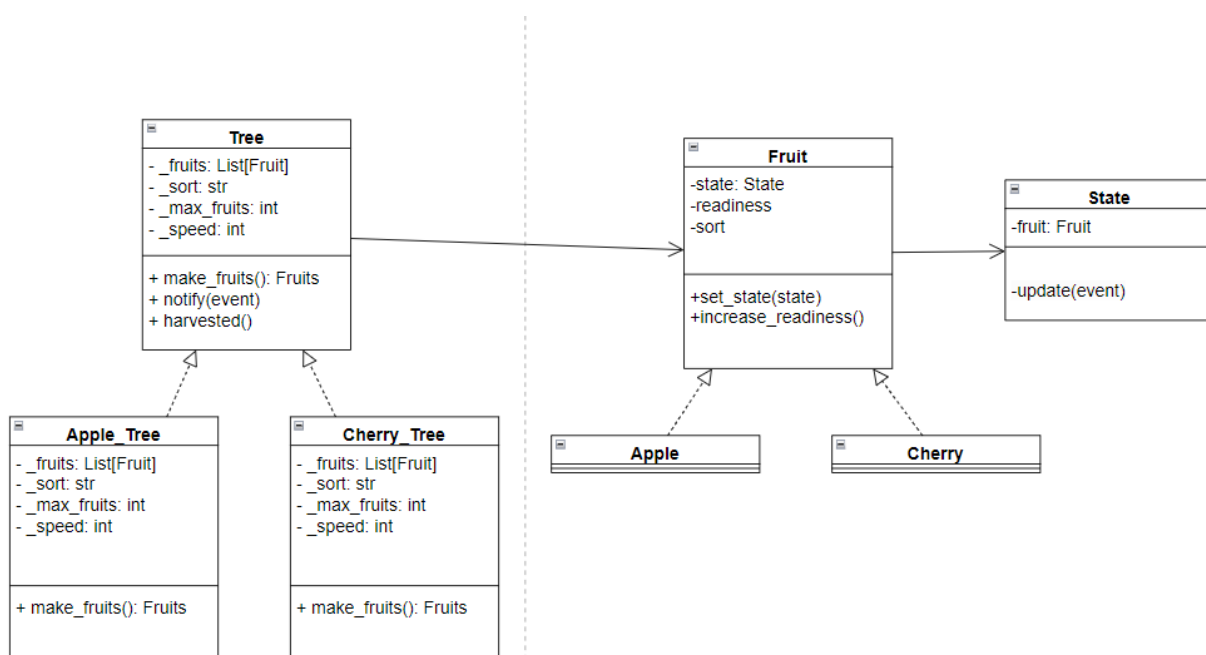


Рис. 2.1.2. UML діаграма класів, які входять до шаблонів “Стан” і “Спостерігач”

Учасники шаблонів:

- **Publisher і Context (Tree)** – оголошує метод повідомлення для

підписників і визначає контекст для зміни станів;

- **Subscriber** (Fruit) – реагує на повідомлення і змінює стан.
- **State** (State) – в залежності від нащадку змінює логіку фрукта.

3) Замісник

Визначення:

Замісник – структурний шаблон проєктування, що дає змогу підставляти замість реальних об’єктів спеціальні об’єкти-замінники. Ці об’єкти перехоплюють виклики до оригінального об’єкта, дозволяючи зробити щось до чи після передачі виклику оригіналові. UML діаграма класів представлена на рис. 2.1.3.

Обґрунтування використання шаблону:

У програмі існує клас Garden, який спочатку не має дерев. Якщо користувач забуде про це і буде поливати сад просто так, то ніякого сенсу в користуванні цього додатку не буде. Тому для класу саду була створена оболонка яка перевіряє наявність хоча б одного дерева в саду для того щоб дозволити виконувати над ним операції.

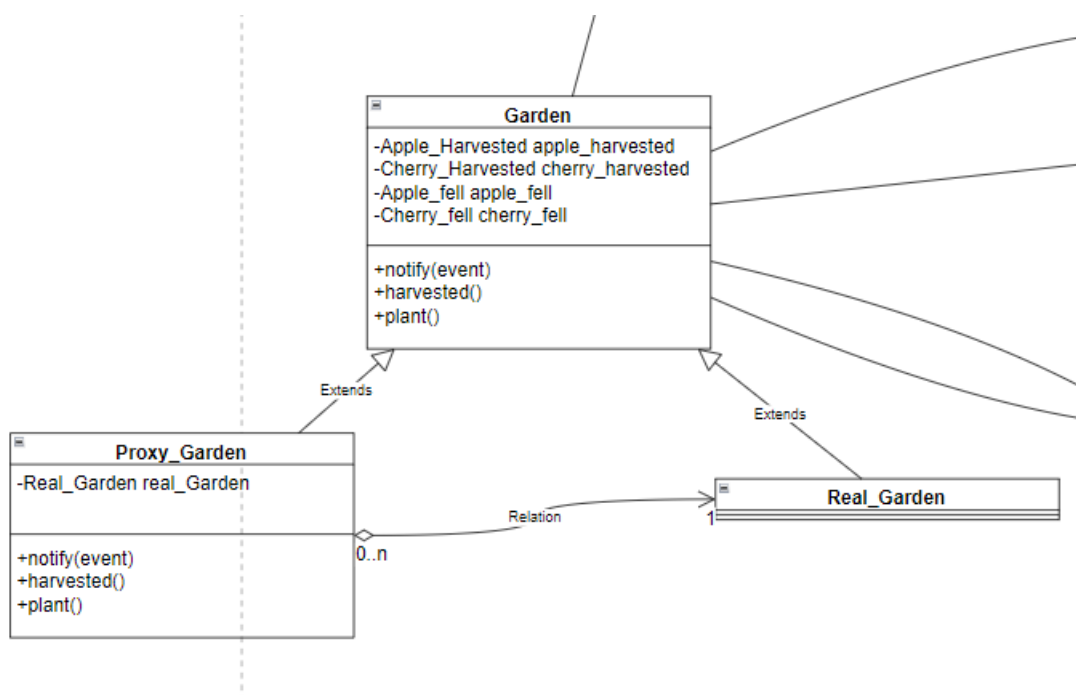


Рис. 2.1.3. UML діаграма класів, які входять до шаблону “Замісник”

Учасники шаблону:

- **Garden (Interface)** – оголошує загальний інтерфейс операцій для саду та його замісника;

- **RealSubject** (Real_Garden) – реалізує інтерфейс операцій саду;
- **Proxy** (Proxy_Garden) – об'єкт-замісник, що зберігає посилання на оригінальний об'єкт магазину та контролює до нього доступ.

4) Абстрактна фабрика

Визначення:

Абстрактна фабрика — це породжувальний патерн проектування, що дає змогу створювати сімейства пов'язаних об'єктів, не прив'язуючись до конкретних класів створюваних об'єктів.

Обґрунтування використання шаблону:

За допомогою абстрактної фабрики були реалізовані класи, які відповідають за статистику. Без абстрактної фабрики було б набагато важче, адже тоді б прийшлося створювати конкретний клас для кожної операції для кожного об'єкту класу.

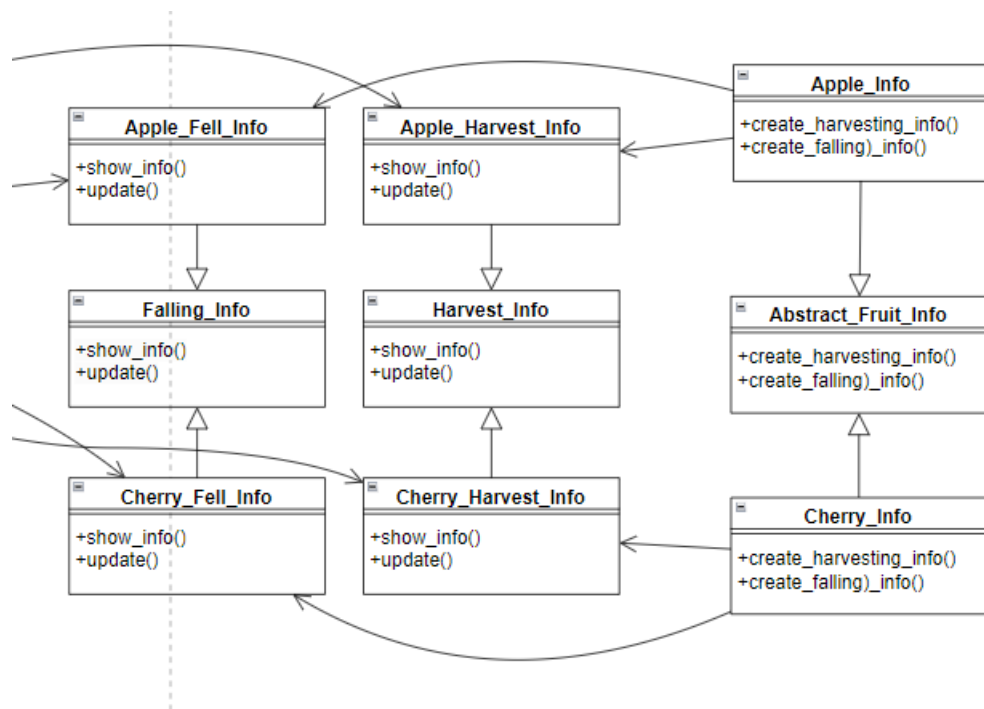


Рис. 2.1.4. UML діаграма класів, які входять до шаблону “Абстрактна фабрика”

Учасники шаблону:

- **AbstractFactory** (**Abstract_Fruit_Info**) – оголошує методи створення різних абстрактних продуктів;
- **ConcreteFactory** (**Apple_Info**, **Cherry_Info**) – дають можливість створювати продукти різної варіації.
- **Product** (**Harvest_Info**, **Fell_Info**) – дають інтерфейс продуктам.
- **ConcreteProducts** (всі інші) – конкретні продукти з різними

варіаціями.

2.2. Діаграма класів

Стан + оглядач

Фабричний метод + Шаблонний метод

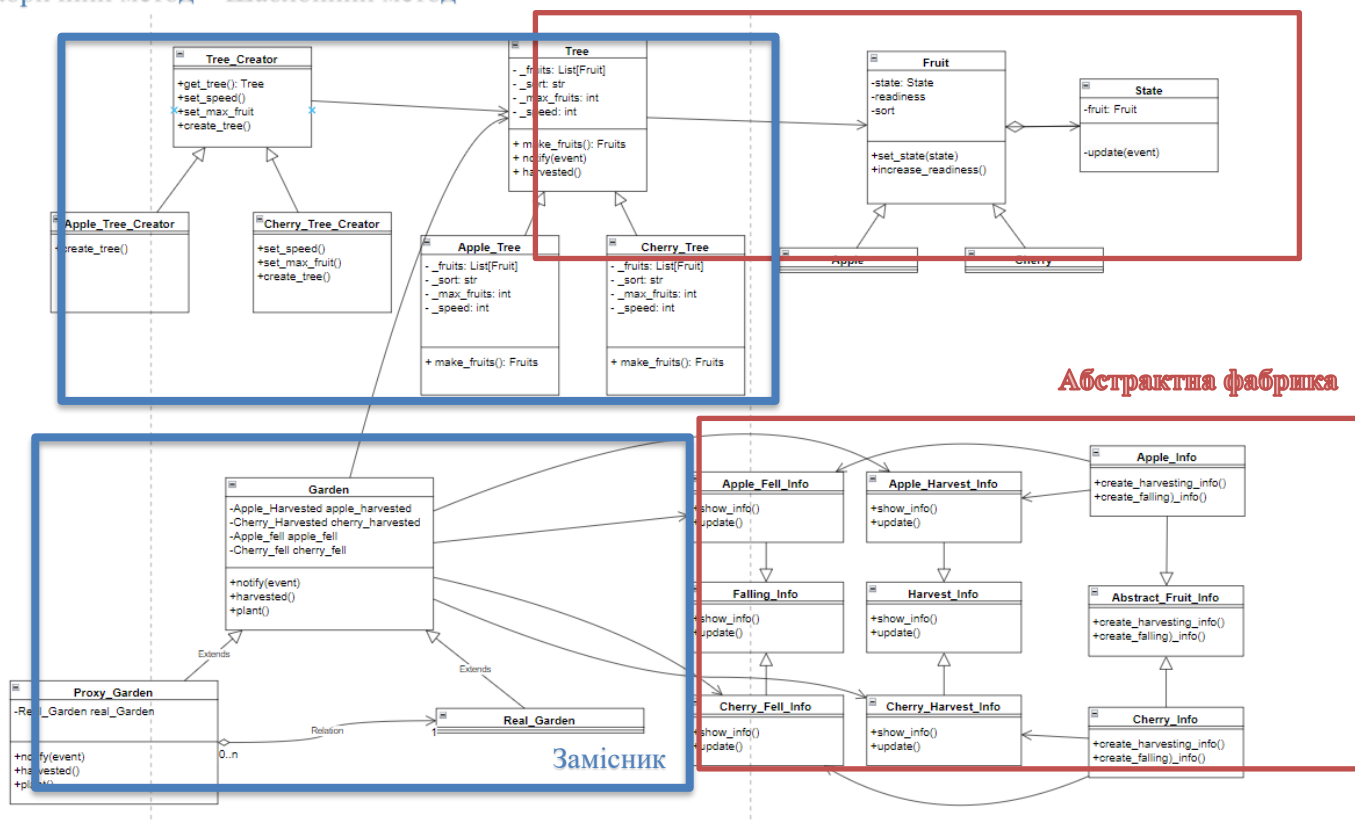
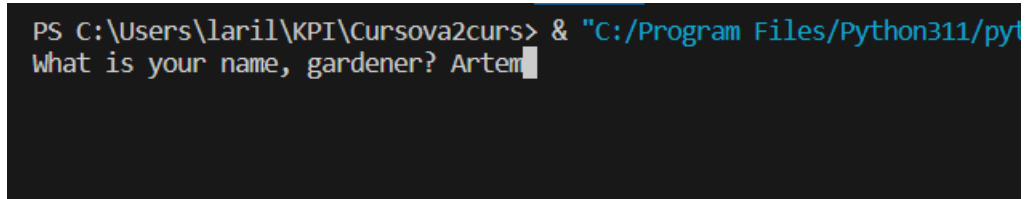


Рис. 2.2 UML діаграма класів фруктового саду

2.3. Опис результатів роботи програми

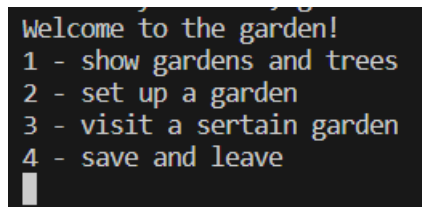
При початку роботи програми користувачу пропонується ввести ім'я:



```
PS C:\Users\laril\KPI\Cursova2curs> & "C:/Program Files/Python311/pyt
what is your name, gardener? Artem
```

Рис. 2.3.1

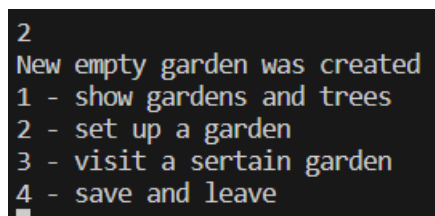
Далі пропонуються 4 команди на вибір: показати усі сади, створити сад, зайнятися окремим садом і вийти та зберегти процес:



```
Welcome to the garden!
1 - show gardens and trees
2 - set up a garden
3 - visit a certain garden
4 - save and leave
```

Рис. 2.3.2

Для створення саду натиснемо 2:



```
2
New empty garden was created
1 - show gardens and trees
2 - set up a garden
3 - visit a certain garden
4 - save and leave
```

Рис. 2.3.3

Тепер зайдемо у цей сад:


```

1 - show gardens and trees
2 - set up a garden
3 - visit a certain garden
4 - save and leave
3
Choose a garden to visit: 1
Choose an option:
1 - fruits harvested in this garden
2 - plant a tree in this garden
3 - clean trees
4 - water trees
5 - harvest fruits
6 - fruits fallen in this garden
7 - exit from a garden

```

Рис. 2.3.4

Тут нам пропонуються усі можливі дії, які ми можемо виконати над садом. Посадимо яблуню і оберемо третій сорт:

```

3 - clean trees
4 - water trees
5 - harvest fruits
6 - fruits fallen in this garden
7 - exit from a garden
2
What type? apple
What sort (from 1 to 3)? 3
Apple planted!
Choose an option:

```

Рис. 2.3.5

Тепер для того, щоб яблуня почала давати врожай, потрібно деяку кількість разів її полити. Також, між поливами потрібно знищувати шкідників командою 3.

```

watering successful!
Choose an option:
1 - fruits harvested in this garden
2 - plant a tree in this garden
3 - clean trees
4 - water trees
5 - harvest fruits
6 - fruits fallen in this garden
7 - exit from a garden

```

Рис. 2.3.6

```

cleaning successful!
Choose an option:
1 - fruits harvested in this garden
2 - plant a tree in this garden
3 - clean trees
4 - water trees
5 - harvest fruits
6 - fruits fallen in this garden
7 - exit from a garden

```

Рис. 2.3.7

Після деякої кількості поливів можемо спробувати зібрати врожай. Для цього використаємо команду 5:

```

7 - exit from a garden
5
Apple 3 sort Harvested - 19
Harvesting successful!
Choose an option:
1 - fruits harvested in this garden
2 - plant a tree in this garden
3 - clean trees

```

Рис. 2.3.8

Як бачимо, нам вдалося зібрати 19 яблук третього сорту. Тепер подивимося скільки яблук впало поки ми поливали сад:

```

3 - clean trees
4 - water trees
5 - harvest fruits
6 - fruits fallen in this garden
7 - exit from a garden
6
Apple 3 sort Fell - 64
Choose an option:

```

Рис. 2.3.9

Як бачимо, 64 яблука ми втратили.

Також ми можемо повернутись у головне меню та подивитись на те які дерева в нашому саду:

```
1 - show gardens and trees
2 - set up a garden
3 - visit a certain garden
4 - save and leave
1
1 garden:
  Apple tree
    sort - 3
    speed - 24
    max fruits - 51
1 - show gardens and trees
2 - set up a garden
3 - visit a certain garden
4 - save and leave
```

Puc. 2.3.10

ВИСНОВКИ

Метою даної курсової роботи було розроблення ПЗ фруктового саду з використанням шаблонів проєктування. Підставою для розроблення стало завдання на виконання курсової роботи з дисципліни «Програмування» студентами II курсу кафедри ПЗКС НТУУ «КПІ ім. І.Сікорського».

Для досягнення поставленої мети у повному обсязі виконано завдання, визначені у аркуші завдання на курсову роботу; розроблено графічні матеріали; реалізовано всі вимоги до програмного продукту, програмного та апаратного забезпечення, наведені у технічному завданні; створено відповідну документацію.

Розроблене програмне забезпечення представляє собою симуляцію фруктового саду, яка дозволяє користувачу ним керувати.

Програму створено на основі використання таких структурних, породжувальних та поведінкових шаблонів проєктування, як: Абстрактна фабрика, Стан, Замісник, Шаблонний метод, Спостерігач, Фабричний метод.

Для розроблення програмного забезпечення використано мову програмування Python.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. C# 3.0 Design Patterns / Judith Bishop – 2007. – 22 с., – 101 с., – 122 с., – 139 с., –158 с., –210 с.
2. Design Patterns via C#. Прийоми об'єктно-орієнтованого проєктування/ Олександр Шевчук, Дмитро Охрименко, Андрій Касьянов – 2015. – 35 с., – 72 с., – 155 с., – 228 с., – 265 с., – 270 с.
3. Шаблоны проектирования. Новый подход к объектно-ориентированному анализу и проектированию / Алан Шаллоуей, Джеймс Р. Тротт. — М. : «Вильямс», 2002. — 288 с.
4. Refactoring Guru [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://refactoring.guru/uk/design-patterns/> (дата звернення 27.06.2022) - Назва з екрана
5. Software Design Patterns [Електронний ресурс]: [Веб-сайт]. - Режим доступу: <https://www.geeksforgeeks.org/software-design-patterns> (дата звернення 27.06.2022) - Назва з екрана