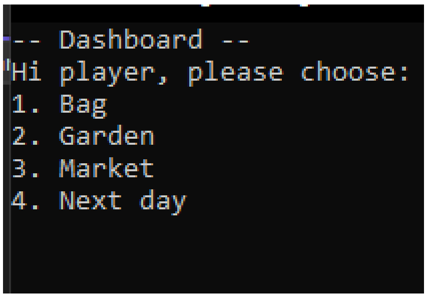
Design Overview for Garden Game

Name: Le Nho Bach

Student ID: 103487884

Summary of Program

This program is a game that allows players to build their own gardens. Players can buy and sell resources like seeds, water or fertilizer, and buy the plant holder like pots or normal plots to raise flowers or fruits. They can also sell the plant after it is fully-growth, or sell the plant holder if players feel like their garden is tight.



```
-- Dashboard --
Hi player, please choose:
1. Bag
2. Garden
3. Market
4. Next day
```

The program uses OOP Design, with multiple principles used to increase the performance and clarity of the code.

Required Roles

Class Bag

Responsibility	Type Details	Notes
AddSeed	void method	Add a seed to the bag
UseSeed	void method	Remove the seed from the bag and add it to a plant holder
UseWater	void method	Lose an amount of water in bag
Use Fertilizer	void method	Lose an amount of

		fertilizer in the bag
AllSeedTypeOf	bool method	Check whether a type of seed (Fruit or Flower) is in the bag or not
Money	double property	Return and change the money in the bag
NumberOfSeed	read-only int property	Return the number of seeds in the bag
Fertilizer	double property	Return and change the amount of fertilizer in the bag
Water	double property	Return and change the amount of water in the bag

Class PlantHolder

Responsibility	Type Details	Notes
AddSeed	void method	Add a seed to an empty plant holder
OneDayPass	void method	Update the plant after a day
CheckDone	virtual void method	Update the price of the plant holder after the plant is fully-growth
CheckWatered	void method	Check whether the plant is watered enough that day or not
Dead	void method	Make the plant dies due to the lack amount of water
SeedType	WaterDemand property	Return the type of the seed

Price	double property	Return the price of the plant holder
DaysRemaining	int property	Return how many days left until the plant is fully-growth
ShortDescription	virtual string property	Return a short description of the plant holder
LongDescription	string property	Return a long description of the plant holder
Seed	Seed property	Return or change the seed in the plant holder
Seeded	boolean property	Return whether the plant holder has a seed inside or not
HolderType	virtual HolderType property	Return the type of the plant holder
Name	string property	Return the name of the plant holder
WaterToday	double property	Return the amount of water has been watered to the plant today
Fully_Growth	boolean property	Return whether the plant is fully-growth or not

Class FlowerPot: A child class of the class PlantHolder

Responsibility	Type Details	Notes
HolderType	override HolderType property	Return the type of the plant holder

Class Plot: A child class of the class PlantHolder

Responsibility	Type Details	Notes
HolderType	override HolderType property	Return the type of the plant holder

Class Garden

Responsibility	Type Details	Notes
AddPlantHolder	void method	Add a plant holder to the garden
RemovePlantHolder	void method	Remove a plant holder from the garden
OneDayPass	void method	Update all the plant holder in the garden after a day pass
Sell	void method	Remove/Reset the plant holder, sell the plant inside and get money
ResetPlantHolder	void method	Reset a plant holder
HolderList	List<PlantHolder> property	Return the list of all plan holders in the garden
Description	string property	Return the description of the garden

Struct Seed

Responsibility	Type Details	Notes
Description	string property	Return the information of the seed

Class BuyingState

Responsibility	Type Details	Notes
SeedShopDescription	string property	Return the description of all seed in the shop
FlowerPotShopDescription	string property	Return the description of all pot in the shop
SeedCollection	List<Seed> property	Return the collection of seeds
PotCollection	List<FlowerPot> property	Return the collection of flower pots

Class System

Responsibility	Type Details	Notes
GetIntBetween	int function	Get the max and the min integer value, then assure that the user would give the valid input

GetDoubleBetween	double function	Get the max and the min double value, then assure that the user would give the valid input
AskIntAmount	int function	Assure the user to give the valid amount of something
AskDoubleAmount	double function	Assure the user to give the valid amount of something
PressEnter()	void function	Let user click a button before clear the screen
CheckMoney	bool function	Test whether the user can buy something
Dashboard	void function	Show the dashboard of the game
BagBoard	void function	Show the bag of the player
GardenBoard	void function	Show the garden of the player
VisitPlantHolder	void function	Allow player to interact with the plant holders in the garden

BuyingPlantHolder	void function	Allow player to buy new plant holders
BuyingSeed	void function	Allow player to buy seeds
BuyingWater	void function	Allow player to buy water
Buying Fertilizer	void function	Allow player to buy fertilizer
VisitSellingState	void function	Allow player to sell plant holders
Start	void function	Start the game

Enum Type

Value	Notes
None	Nothing
Flower	This is a flower seed and can only be planted in a flower pot
Fruit	This is a fruit seed and can only be planted in a plot

Enum WaterDemand

Value	Notes
None	Nothing
NeedWater	This seed need to be watered
DontNeedWater	This seed does not need to be watered

Enum HolderType

Value	Notes
Plot	This is used to plant fruit
FlowerPot	This is used to plant flower

The use of abstraction: GetIntBetween(int start, int end)

This is one of the most useful functions in my whole program. For details, it gets 2 integer values: start and end. Then the program will ask the user to give the valid input, which is an integer between start and end values. After creating this function, I do not need to care about how will I require the user to give a choice and assure that the number is valid anymore

The use of inheritance and polymorphism: The relationship between classes PlantHolder, Plot and FlowerPot.

The FlowerPot and Plot inherit the features, functions, properties... of the class PlantHolder. However, when we call HolderType, the FlowerPot will return HolderType.FlowerPot while the Plot will return the HolderType.Plot