

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ “КПІ”**

**Кафедра**

**Автоматизованих систем обробки інформації та управління**

**Звіт**

**до лабораторної роботи з дисципліни**

**“** **Технології розроблення програмного забезпечення -1. Основи розробки програмного забезпечення на платформі Microsoft.NET”**

**Комп’ютерний практикум № 3.**

**Варіант - 6**

**Перевірли:**

**доц. Ліщук К. І.**

**Виконала:**

**студентка групи ІС-83**

**Прийдун М.В.**

**м. Київ-2020**

**Комп‘ютерний практикум № 3. Шаблони проектування. Поведінкові шаблони**

**Мета:** ознайомитися з основними шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.

**Завдання:**

**6)** Розробити модель ігрової системи. Передбачити наявність фентезійних персонажів: орки, тролі, пегаси, ельфи, вампіри, гарпії і ін. Врахувати, що деякі персонажі ходять, інші - літають, треті - і ходять і літають. Літати також може група персонажів за допомогою магії.

У роботі використано патерн Стратегія (Strategy), оскільки у роботі є кілька споріднених класів, які відрізняються поведінкою. Можна задати один основний клас, а різні варіанти поведінки винести в окремі класи і при необхідності їх застосовувати

|  |  |
| --- | --- |
| Клас | Призначення |
| FactoryCharacter | Клас фабрика |
| Character | Абстрактний клас від якого наслідуються усі інші персонажі |
| Pegasus, Elf, Harpy, Orc, Troll, Vampire | Класи усіх персонажів |
| Imovable | Це загальний інтерфейс для всіх реалізацій Move. |
| FlyAndGoMove, GoMove, FlyMove | Класи способу руху персонажів |

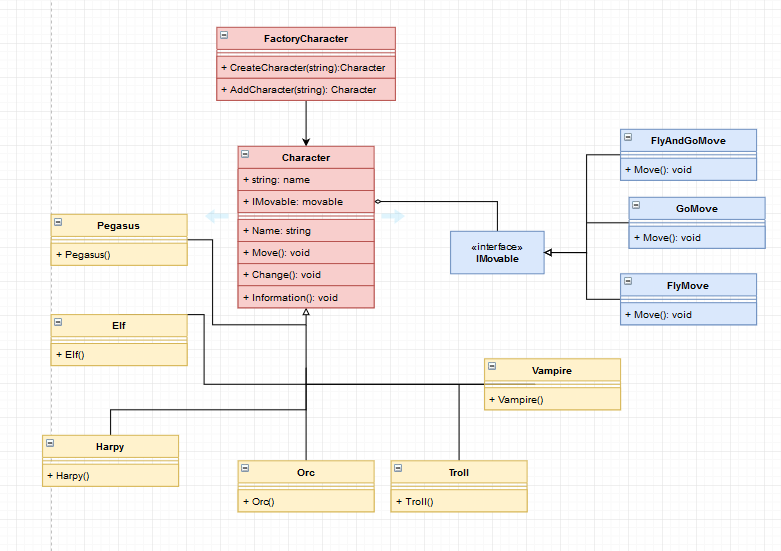


Рис.1 Діаграма класів

## Файл Character.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

abstract class Character

{

public string name;

public IMovable movable;

public string Name

{

get

{

return name;

}

set

{

name = value;

}

}

abstract public void Change();

public void Move()

{

movable.Move();

}

public void Information()

{

Console.WriteLine("Тип персонажа: {0}", Name);

Move();

}

}

}

## Файл Elf.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Elf : Character

{

public Elf()

{

name = "elf";

movable = new FlyAndGoMove();

Console.WriteLine("Створено ельфа.");

}

public override void Change()

{

Console.WriteLine("Виберiть, що має робити персонаж: 1-ходити, 2 - лiтати за допомогою магiї");

string move= Console.ReadLine();

if (move == "1")

movable = new GoMove();

if (move == "2")

movable = new FlyMove();

}

}

}

## Файл FactoryCharacter.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class FactoryCharacter

{

public Character AddCharacter(string type)

{

Character person = CreateCharacter(type);

//person.informationCitizen();

person.Move();

person.Change();

person.Information();

return person;

}

public Character CreateCharacter(string type)

{

if (type == "elf")

return new Elf();

else if (type == "ork")

return new Orc();

else if (type == "pegasus")

return new Pegasus();

else if (type == "troll")

return new Troll();

else if (type == "vampire")

return new Vampire();

else if (type == "harpy")

return new Harpy();

return null;

}

/\*public void informationCivilization()

{

Console.WriteLine("The name of civilization: {0}", name);

Console.WriteLine("Territories: {0}", territories);

Console.WriteLine("Population: {0}", population);

}\*/

}

}

## Файл FlyAndGoMove.cs

using System.Text;

namespace lab3\_v6\_game

{

class FlyAndGoMove : IMovable

{

public void Move()

{

Console.WriteLine("Персонаж може ходити i лiтати.");

}

}

}

## Файл FlyMove.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class FlyMove : IMovable

{

public void Move()

{

Console.WriteLine("Персонаж лiтає.");

}

}

}

## Файл GoMove.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class GoMove : IMovable

{

public void Move()

{

Console.WriteLine("Персонаж ходить.");

}

}

}

## Файл Harpy.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Harpy : Character

{

public Harpy()

{

name = "harpy";

movable = new FlyMove();

Console.WriteLine("Створено гарпiю.");

}

public override void Change()

{

Console.WriteLine("Тип руху персонажа змiнити неможливо.");

}

}

}

## Файл IMovable.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

interface IMovable

{

void Move();

}

}

## Файл Orc.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Orc : Character

{

public Orc()

{

name="orc";

movable = new GoMove();

Console.WriteLine("Створено орка.");

}

public override void Change()

{

Console.WriteLine("Тип руху персонажа змiнити неможливо.");

}

}

}

## Файл Pegasus.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Pegasus : Character

{

public Pegasus()

{

name = "pegasus";

movable = new FlyAndGoMove();

Console.WriteLine("Створено пегаса.");

}

public override void Change()

{

Console.WriteLine("Виберiть, що має робити персонаж: 1-ходити, 2 - лiтати");

string move = Console.ReadLine();

if (move == "1")

movable = new GoMove();

if (move == "2")

movable = new FlyMove();

}

}

}

## Файл Program.cs

using System;

namespace lab3\_v6\_game

{

class Program

{

static void Main()

{

FactoryCharacter My = new FactoryCharacter();

Character character;

// character= My.AddCharacter("elf");

do

{

Console.Clear();

Console.WriteLine(" MENU");

Console.WriteLine("1. Створити ельфа.");

Console.WriteLine("2. Створити гарпiю.");

Console.WriteLine("3. Створити орка.");

Console.WriteLine("4. Створити пегаса.");

Console.WriteLine("5. Створити троля.");

Console.WriteLine("6. Створити вампiра.");

Console.WriteLine("7. Вихiд.");

string option = Console.ReadLine();

switch (option)

{

case "1":

character = My.AddCharacter("elf");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "2":

character = My.AddCharacter("harpy");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "3":

character = My.AddCharacter("ork");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "4":

character = My.AddCharacter("pegasus");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "5":

character = My.AddCharacter("troll");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "6":

character = My.AddCharacter("vampire");

Console.WriteLine("Натиснiть Enter, щоб продовжити...");

Console.ReadLine();

break;

case "7":

Environment.Exit(0);

break;

default:

{

Console.WriteLine("Помилка! Введiть iснуючий пункт меню.");

Console.ReadLine();

break;

}

}

} while (true);

}

}

}

## Файл Troll.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Troll : Character

{

public Troll()

{

name = "troll";

movable = new GoMove();

Console.WriteLine("Створено троля.");

}

public override void Change()

{

Console.WriteLine("Тип руху персонажа змiнити неможливо.");

}

}

}

## Файл Vampire.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab3\_v6\_game

{

class Vampire : Character

{

public Vampire()

{

name = "vampire";

movable =new FlyAndGoMove();

Console.WriteLine("Створено вампiра.");

}

public override void Change()

{

Console.WriteLine("Виберiть, що має робити персонаж: 1-ходити, 2 - перетворитися на летючу мишу i лiтати");

string move = Console.ReadLine();

if (move == "1")

movable = new GoMove();

if (move == "2")

movable = new FlyMove();

}

}

}

**Висновок.** У результаті виконання комп’ютерного практикуму було розроблено модель ігрової системи. Для цього використано патерн Стратегія (Strategy). Також побудовано UML-діаграму класів та описано кожен клас.