Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

з комп’ютерного практикуму № 2

на тему :

„ Шаблоны проектирования. Структурные шаблоны”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав студент** |  | *ІП-63 Зозуля А.В.* |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  |  |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2018

**ЗМІСТ**

[**1**](#_3dy6vkm) **Мета роботи 3**

[**2**](#_1t3h5sf) **Постановка задачі 4**

[**3**](#_4d34og8) **UML ДІАГРАМА КЛАСУ 5**

[**4**](#_2s8eyo1) **Висновок 6**

[**5**](#_17dp8vu) **Тескт програмного коду 7**

# 1. Мета роботи

*Мета роботи* – ознакомиться с основными шаблонами проектирования, научиться применять их при проектировании и разработке ПО.

# 

# 

# 

# 2. Постановка задачі

Постановка задачи к компьютерному практикуму № 2

При выполнении компьютерного практикума следует реализовать следующие

действия:

1) Реализовать задачу согласно вариантов, приведенных ниже (можно

предложить реализацию собственной задачи)

2) Кратко описать архитектуру проекта

3) Привести UML-диаграмму классов

4) Описать используемые структурные паттерны. Для каждого паттерна

нужно указать составные классы и их назначение

6) Разработать структуру организации армии в игре фэнтези. Армия может состоять из отрядов эльфов, орков, минотавров, кентавров, циклопов, драконов, гидр, рыцарей. Армия может содержать как отряды, так и одиночных воинов, отряд может состоять из других отрядов и одиночных воинов.

# 3. UML ДІАГРАМА КЛАСУ

# 

# 4. Висновок

В ході виконання лабораторної роботи було здійснено розробку програмного забезпечення, використовуючи структурний паттерн Adapter.

# 5. Тескт програмного коду

using System;

namespace lab2

{

class Program

{

static void Main(string[] args)

{

var army = new Army();

var platoon = new Platoon();

var elfPlatoon = new PlatoonAdapter(new Platoon());

elfPlatoon.AddWarrior(new Elf());

elfPlatoon.AddWarrior(new Elf());

platoon.AddWarrior(new Elf());

platoon.AddWarrior(elfPlatoon);

army.AddPlatoon(platoon);

army.AddPlatoon(new Orc());

army.AddPlatoon(new Hydra());

army.AddPlatoon(new Cyclop());

army.AddPlatoon(new Knight());

foreach (var slogan in army.SayAllSlogans())

{

Console.WriteLine(slogan);

}

}

}

}

using System.Collections.Generic;

namespace lab2

{

public class Army

{

private List<IWarrior> \_army;

public Army(List<IWarrior> army)

{

\_army = army;

}

public Army()

{

\_army = new List<IWarrior>();

}

public void AddPlatoon(IWarrior platoon)

{

\_army.Add(platoon);

}

public List<string> SayAllSlogans()

{

List<string> slogans = new List<string>();

foreach (var platoon in \_army)

{

slogans.Add(platoon.SaySlogan());

}

return slogans;

}

}

}

using System.Collections.Generic;

using System.Numerics;

namespace lab2

{

public class Platoon : IWarrior

{

private List<IWarrior> \_warriors;

public Platoon()

{

\_warriors = new List<IWarrior>();

}

public string SaySlogan()

{

if (\_warriors[0] != null)

{

return \_warriors[0].SaySlogan();

}

else

{

return "";

}

}

public void AddWarrior(IWarrior warrior)

{

\_warriors.Add(warrior);

}

}

//Паттерн Adapter

public class PlatoonAdapter : IWarrior

{

private Platoon \_platoon;

public PlatoonAdapter(Platoon platoon)

{

\_platoon = platoon;

}

public void AddWarrior(IWarrior warrior)

{

\_platoon.AddWarrior(warrior);

}

public string SaySlogan()

{

return \_platoon.SaySlogan();

}

}

}

using System;

namespace lab2

/\*

https://proglib.io/p/structural-patterns/

Разработать структуру организации армии в игре фэнтези. Армия может

состоять из отрядов эльфов, орков, минотавров, кентавров, циклопов,

драконов, гидр, рыцарей. Армия может содержать как отряды, так и

одиночных воинов, отряд может состоять из других отрядов и одиночных

воинов.

\*/

{

public interface IWarrior

{

string SaySlogan();

}

public class Elf : IWarrior

{

public string SaySlogan()

{

return "Elf is power!";

}

}

public class Orc : IWarrior

{

public string SaySlogan()

{

return "Orc is power!";

}

}

public class Minotaur : IWarrior

{

public string SaySlogan()

{

return "Minotaur is power!";

}

}

public class Centaur: IWarrior

{

public string SaySlogan()

{

return "Centaur is power!";

}

}

public class Cyclop : IWarrior

{

public string SaySlogan()

{

return "Cyclop is power!";

}

}

public class Dragon : IWarrior

{

public string SaySlogan()

{

return "Dragon is power!";

}

}

public class Hydra : IWarrior

{

public string SaySlogan()

{

return "Hydra is power!";

}

}

public class Knight : IWarrior

{

public string SaySlogan()

{

return "Knight is power!";

}

}

}