

**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 № 1.**

**Перевірли:**

**доц. Ліщук К. І.**

**м. Київ-2020**

**Виконала:**

**студентка групи ІС-83**

**Прийдун М.В.**

**Комп‘ютерний практикум № 1. Шаблони проектування. Породжуючі шаблони**

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

У роботі використаний патерн Абстрактна фабрика (Abstract Factory).

Це пов’язано з тим, що у моєму випадку система повинна залишатися незалежною як від процесу створення нових об’єктів, так і від типів породжуючих об’єктів. При цьому необхідно створювати групи взаємопов’язаних об’єктів, виключаючи можливість одночасного використання об’єктів різних груп в одному контексті. Це означає, що зовнішній вигляд, бойові порядки та характеристики різних Цивілізацій.

|  |  |
| --- | --- |
| Клас | Призначення |
| Civilization | Основний клас (в патерні- Factory) |
| Citizen | Абстактний клас, (в патерні – AbstractProduct) |
| worker | Підклас, , (в патерні – ConcreteProduct) |
| aristocrat | Підклас, , (в патерні – ConcreteProduct) |
| warrior | Підклас, , (в патерні – ConcreteProduct) |

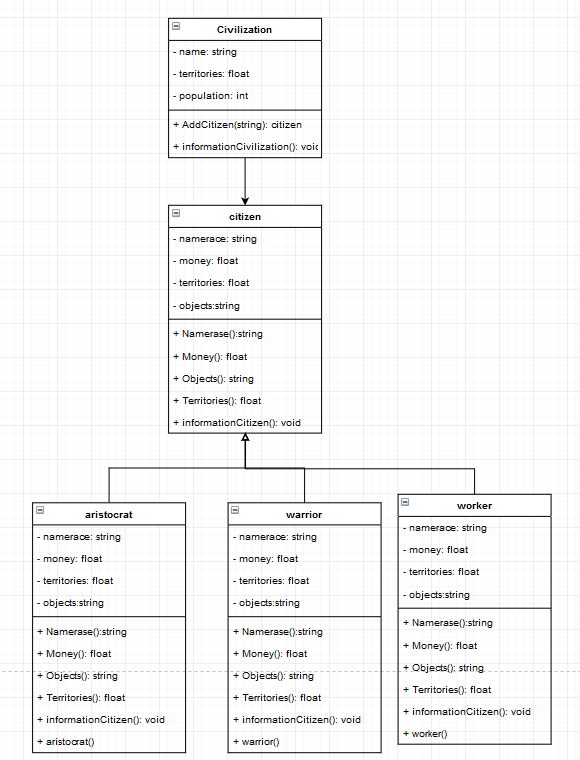


Рис.1 Діаграма класів

## Файл aristocrat.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab1\_v6\_civilization

{

class aristocrat : citizen

{

public aristocrat()

{

money = 10000;

namerace= "aristocrat";

territories=1500;

}

}

}

## Файл citizen.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab1\_v6\_civilization

{

abstract class citizen

{

protected float money;

protected string namerace;

protected float territories;

protected string objects = "forests,fields,dwelling, factories";

public void informationCitizen()

{

Console.WriteLine("The name of the race: {0}", namerace);

Console.WriteLine("Financial stok: {0}", money);

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

Console.WriteLine("Objects are lokated on the territory: {0}", objects);

}

public string Namerase

{

get { return namerace; }

set { namerace = value; }

}

public float Money

{

get { return money; }

set { money = value; }

}

public float Territories

{

get { return territories; }

set { territories = value; }

}

public string Objects

{

get { return objects; }

set { objects = value; }

}

}

}

## Файл Civilization.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab1\_v6\_civilization

{

class Civilization

{

//List<citizen> citizens = new List<citizen>();

private string name;

private float territories;

private int population;

public Civilization()

{

name = "";

territories = 0;

population = 0;

}

public string Name

{

get { return name; }

set { name = value; }

}

public float Territories

{

get { return territories; }

set { territories = value; }

}

public int Population

{

get { return population; }

set { population = value; }

}

public citizen AddCitizen(string type)

{

citizen person = CreateCitizen(type);

population++;

territories += person.Territories;

person.informationCitizen();

return person;

}

public citizen CreateCitizen(string type)

{

if (type == "aristocrat")

return new aristocrat();

else if (type == "warrior")

return new warrior();

else if (type == "worker")

return new worker();

return null;

}

public void informationCivilization()

{

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

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

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

}

}

}

## Файл Program.cs

using System;

using System.Collections.Generic;

namespace lab1\_v6\_civilization

{

class Program

{

static void Main(string[] args)

{

string Name;

Civilization one=new Civilization();

int number = 0,// номер цивілізації

n = 1;//кількість цивілізацій

List<Civilization> settlement = new List<Civilization>();

Console.WriteLine("Enter name of civilization:");

Name = Console.ReadLine();

settlement.Add(one);

settlement[number].Name = Name;

do

{

Console.Clear();

Console.WriteLine("Civilization: " + settlement[number].Name);

Console.WriteLine(" MENU");

Console.WriteLine("1. Create aristocrat.");

Console.WriteLine("2. Create warrior.");

Console.WriteLine("3. Create worker.");

Console.WriteLine("4. Show information about civilization.");

Console.WriteLine("5. Create new civilization.");

Console.WriteLine("6. Change civilization.");

Console.WriteLine("7. Exit.");

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

switch(option)

{

case 1:

settlement[number].AddCitizen("aristocrat");

Console.WriteLine("Press Enter to continue...");

Console.ReadLine();

break;

case 2:

settlement[number].AddCitizen("warrior");

Console.WriteLine("Press Enter to continue...");

Console.ReadLine();

break;

case 3:

settlement[number].AddCitizen("worker");

Console.WriteLine("Press Enter to continue...");

Console.ReadLine();

break;

case 4:

settlement[number].informationCivilization();

Console.WriteLine("Press Enter to continue...");

Console.ReadLine();

break;

case 5:

number++;

n++;

Console.WriteLine("Enter name of civilization:");

Name = Console.ReadLine();

Civilization created = new Civilization();

settlement.Add(created);

settlement[number].Name = Name;

break;

case 6:

for (int i = 0; i < n; i++)

{

Console.WriteLine("=======Press {0} to select civilization===== ", i + 1);

settlement[i].informationCivilization();

}

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

if (ch <= n)

number = ch-1;

else

{

Console.WriteLine("Error! There is no civilization with that number ");

Console.ReadLine();

}

break;

case 7:

Environment.Exit(0);

break;

default:

{

Console.WriteLine("Error! Enter an existing menu item.");

Console.ReadLine();

break;

}

}

} while (true);

}

}

}

## Файл warrior.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace lab1\_v6\_civilization

{

class warrior: citizen

{

public warrior()

{

money = 1000;

namerace = "warrior";

territories = 15;

}

}

}

## Файл worker.cs

namespace lab1\_v6\_civilization

{

class worker:citizen

{

public worker()

{

money = 800;

namerace = "worker";

territories = 15;

}

}

}

**Висновок.** У результаті виконання комп’ютерного практикуму було створено стратегічну гру «Цивілізації». Для цього використано патерн Abstract Factory. Також побудовано UML-діаграму класів та описано кожен клас.