МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Технология программирования»

**Лабораторная работа №4.**

**Фабричный метод (Factory Method)**

Выполнил:

студент группы ИВТАПбд-21

Кондратьев П. С.

Проверил:

Фролов В. А.

Ульяновск, 2018

Фабричный метод (Factory Method) - это паттерн, который определяет интерфейс для создания объектов некоторого класса, но непосредственное решение о том, объект какого класса создавать происходит в подклассах. То есть паттерн предполагает, что базовый класс делегирует создание объектов классам-наследникам.

**Когда надо применять паттерн**

* Когда заранее неизвестно, объекты каких типов необходимо создавать
* Когда система должна быть независимой от процесса создания новых объектов и расширяемой: в нее можно легко вводить новые классы, объекты которых система должна создавать.
* Когда создание новых объектов необходимо делегировать из базового класса классам наследникам

**Техническое задание**

Реализовать паттерн и продемонстрировать его работу с различными типами объектов.

Исходный код

class Program

{

static void Main(string[] args)

{

Info facility = new Horde();

facility.GetUnit("Horseman");

Console.WriteLine();

facility.GetUnit("Archer");

Console.WriteLine();

facility.GetUnit("Infantryman");

Console.ReadLine();

}

} class SimpleFactory

{

public Warrior GetUnit(string type)

{

Warrior warrior = new Warrior();

if (type == "Horseman")

warrior = new AllianHorseman();

else if (type == "Archer")

warrior = new AllianceArcher();

else if (type == "Infantryman")

warrior = new AllianInfantryman();

return warrior;

}

} class Alliance : Info

{

public override Warrior CreateUnit(string type)

{

Warrior warrior = new Warrior();

if (type == "Horseman")

warrior = new AllianHorseman();

else if (type == "Archer")

warrior = new AllianceArcher();

else if (type == "Infantryman")

warrior = new AllianInfantryman();

return warrior;

}

} class Horde : Info

{

public override Warrior CreateUnit(string type)

{

Warrior warrior = new Warrior();

if (type == "Horseman")

warrior = new HordeHorseman();

else if (type == "Archer")

warrior = new HordeArcher();

else if (type == "Infantryman")

warrior = new HordeInfantryman();

return warrior;

}

} abstract class Info

{

public Warrior GetUnit(string type)

{

Warrior warrior = CreateUnit(type);

warrior.Configure();

return warrior;

}

public abstract Warrior CreateUnit(string type);

class Warrior

{

protected string Name = "[]";

protected string Attack = "0";

protected string Armor = "0";

protected string HP = "0";

protected string MP = "0";

public void Configure()

{

Console.WriteLine("Info {0}", Name);

Console.WriteLine("Health Points {0}", HP);

Console.WriteLine("Mana Points {0}", MP);

Console.WriteLine("Attack {0}", Attack);

Console.WriteLine("Armor {0}", Armor);

}

} class AllianceArcher : Warrior

{

public AllianceArcher()

{

Name = "AllianArcher";

Attack = "18";

Armor = "7";

HP = "14";

MP = "21";

}

} class HordeHorseman : Warrior

{

public HordeHorseman()

{

Name = "HordeHorseman";

Attack = "22";

Armor = "8";

HP = "30";

}

}