**Занятие № 19**

**Дата выполнения работы:** 12.05.2023

# **Тема работы:** «Создание программной системы, использующей порождающие шаблоны»

**Ход работы**

**Задание 1**

Шаблон “Фабричный метод”. Проект “Фабрика смартфонов”. В проекте

должно быть реализовано создание смартфонов с различными

характеристиками.

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace App

{

public partial class Form1 : Form

{

private SmartphoneFactory smartphoneFactory;

public Form1()

{

InitializeComponent();

}

private void btnIPhone\_Click(object sender, EventArgs e)

{

smartphoneFactory = new IPhoneFactory();

smartphoneFactory.DisplaySmartphoneInfo();

}

private void btnSamsungGalaxy\_Click(object sender, EventArgs e)

{

smartphoneFactory = new SamsungGalaxyFactory();

smartphoneFactory.DisplaySmartphoneInfo();

}

}

public interface ISmartphone

{

string GetModel();

string GetProcessor();

int GetRAM();

}

// Конкретный продукт

public class IPhone : ISmartphone

{

public string GetModel()

{

return "iPhone X";

}

public string GetProcessor()

{

return "Apple A11 Bionic";

}

public int GetRAM()

{

return 3;

}

}

// Конкретный продукт

public class SamsungGalaxy : ISmartphone

{

public string GetModel()

{

return "Samsung Galaxy S20";

}

public string GetProcessor()

{

return "Qualcomm Snapdragon 865";

}

public int GetRAM()

{

return 8;

}

}

// Создатель

public abstract class SmartphoneFactory

{

public abstract ISmartphone CreateSmartphone();

public void DisplaySmartphoneInfo()

{

ISmartphone smartphone = CreateSmartphone();

MessageBox.Show($"Model: {smartphone.GetModel()}\nProcessor: {smartphone.GetProcessor()}\nRAM: {smartphone.GetRAM()} GB");

}

}

// Конкретный создатель

public class IPhoneFactory : SmartphoneFactory

{

public override ISmartphone CreateSmartphone()

{

return new IPhone();

}

}

// Конкретный создатель

public class SamsungGalaxyFactory : SmartphoneFactory

{

public override ISmartphone CreateSmartphone()

{

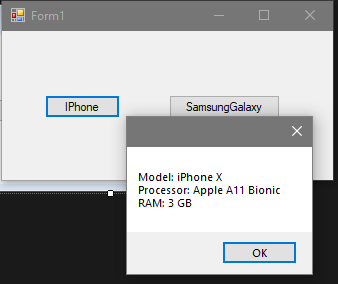
return new SamsungGalaxy();

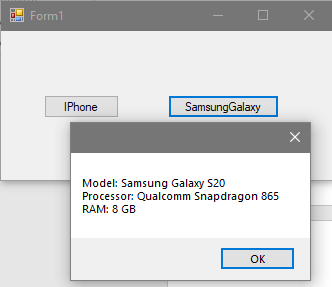
}

}

}

**Результат:**

****

****

**Вывод:** Порождающие шаблоны упрощают создание объектов, уменьшают связанность компонентов, централизуют процесс создания и обеспечивают расширяемость и поддержку изменений в программной системе. Они помогают создавать гибкие и легко поддерживаемые приложения.