Министерство образования Республики Беларусь

Учреждение образования «Брестский государственный технический университет»

Кафедра ИИТ

Отчёт по лабораторной работе №6

Выполнил:

студент 3 курса

Группы ПО-5

Поздняков Д.А.

Проверил:

Крощенко А.А.

Брест 2021

**Цель работы:** приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка C#

**Вариант 5**

**Задание 1.** Завод по производству смартфонов. Обеспечить создание нескольких различных моделей мобильных телефонов с заранее выбранными характеристиками.

**Задание 2.** Проект «Электронный градусник». В проекте должен быть реализован класс, который дает возможность пользоваться аналоговым градусником так же, как и электронным. В классе «Аналоговый градусник» хранится высота ртутного столба и границы измерений (верхняя и нижняя).

**Задание 3.** Проект «Банкомат». Предусмотреть выполнение основных операций (ввод пин-кода, снятие суммы, завершение работы) и наличие различных режимов работы (ожидание, аутентификация, выполнение операции, блокировка – если нет денег). Атрибуты: общая сумма денег в банкомате, ID.

**Код программы (задание 1):**

Program.cs

using System;

namespace lab6.\_1.\_5

{

class Program

{

static void Main(string[] args)

{

Phone phone1 = new Phone(new AndroidFactory());

phone1.OperationSystem();

phone1.installCamera();

phone1.installCPU();

Console.WriteLine();

Phone phone2 = new Phone(new IOSFactory());

phone2.OperationSystem();

phone2.installCamera();

phone2.installCPU();

}

}

}

Phone.cs

using System;

namespace lab6.\_1.\_5

{

abstract class OS

{

public abstract void OperationSystem();

}

abstract class Camera

{

public abstract void InstallCamera();

}

abstract class CPU

{

public abstract void InstallCPU();

}

class Android : OS

{

public override void OperationSystem()

{

Console.WriteLine("Установлена Android");

}

}

class IOS : OS

{

public override void OperationSystem()

{

Console.WriteLine("Установлена IOS");

}

}

class Front : Camera

{

public override void InstallCamera()

{

Console.WriteLine("Установлена фронтальная камера");

}

}

class Main : Camera

{

public override void InstallCamera()

{

Console.WriteLine("Установлена основная камера");

}

}

class M1 : CPU

{

public override void InstallCPU()

{

Console.WriteLine("Установлена процессор М1");

}

}

class Mi\_MIX : CPU

{

public override void InstallCPU()

{

Console.WriteLine("Установлена процессор Mi\_MIX");

}

}

abstract class PhoneFactory

{

public abstract Camera installCamera();

public abstract OS createOS();

public abstract CPU installCPU();

}

class AndroidFactory : PhoneFactory

{

public override Camera installCamera()

{

return new Front();

}

public override OS createOS()

{

return new Android();

}

public override CPU installCPU()

{

return new Mi\_MIX();

}

}

class IOSFactory : PhoneFactory

{

public override Camera installCamera()

{

return new Main();

}

public override OS createOS()

{

return new IOS();

}

public override CPU installCPU()

{

return new M1();

}

}

class Phone

{

private OS operationSystem;

private Camera install\_Camera;

private CPU install\_CPU;

public Phone(PhoneFactory factory)

{

operationSystem = factory.createOS();

install\_Camera = factory.installCamera();

install\_CPU = factory.installCPU();

}

public void installCamera()

{

install\_Camera.InstallCamera();

}

public void OperationSystem()

{

operationSystem.OperationSystem();

}

public void installCPU()

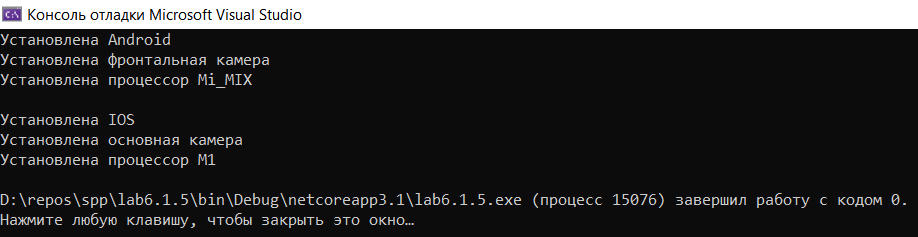
{

install\_CPU.InstallCPU();

}

}

}



**Код программы (задание 2):**

Program.cs

namespace lab6.\_2.\_5

{

class Program

{

static void Main(string[] args)

{

Person person = new Person();

ElectronicThermometer electronic\_Thermometer = new ElectronicThermometer();

person.MeasureYourTemperature(electronic\_Thermometer);

MercuryThermometer mercuryThermometer = new MercuryThermometer();

Thermometer mercury\_Thermometer = new MercuryThermometerToElectronicThermometer(mercuryThermometer);

person.MeasureYourTemperature(mercury\_Thermometer);

}

}

}

AnalogThermometer.cs

namespace lab6.\_2.\_5

{

public interface AnalogThermometer

{

public void RoughlyMeasureTheTemperature();

}

}

ElectronicThermometer.cs

using System;

namespace lab6.\_2.\_5

{

public class ElectronicThermometer : Thermometer

{

private int Temperature = 0;

public int GetTemperature()

{

return Temperature;

}

public void SetTemperature(int temperature)

{

Temperature = temperature;

}

public void MeasureTheTemperature()

{

Random rand = new Random();

SetTemperature(rand.Next(35, 40));

Console.WriteLine($"Температура тела: {Temperature}");

}

}

}

MercuryThermometer.cs

using System;

namespace lab6.\_2.\_5

{

public class MercuryThermometer : AnalogThermometer

{

private int HeightOfTheMercuryColumn = 0;

private int UpperBound = 100, BottomLine = 0;

public int GetTemperature()

{

return HeightOfTheMercuryColumn;

}

public void SetTemperature(int heightOfTheMercuryColumn)

{

HeightOfTheMercuryColumn = heightOfTheMercuryColumn;

}

public int GetUpperBound()

{

return UpperBound;

}

public void SetUpperBound(int upperBound)

{

UpperBound = upperBound;

}

public int GetBottomLine()

{

return HeightOfTheMercuryColumn;

}

public void SetBottomLine(int bottomLine)

{

BottomLine = bottomLine;

}

public void RoughlyMeasureTheTemperature()

{

Console.WriteLine($"Высота ртутного столба: {HeightOfTheMercuryColumn}. Нижняя граница градусника: {BottomLine}, верхняя граница градусника {UpperBound}");

}

}

}

MercuryThermometerToElectronicThermometer.cs

using System;

namespace lab6.\_2.\_5

{

public class MercuryThermometerToElectronicThermometer : Thermometer

{

MercuryThermometer mercuryThermometer;

public MercuryThermometerToElectronicThermometer(MercuryThermometer thermometer)

{

mercuryThermometer = thermometer;

}

public void MeasureTheTemperature()

{

Random rand = new Random();

mercuryThermometer.SetTemperature(rand.Next(35, 40));

Console.WriteLine($"Температура тела: {mercuryThermometer.GetTemperature()}");

}

}

}

Person.cs

namespace lab6.\_2.\_5

{

public class Person

{

public void MeasureYourTemperature(Thermometer thermometer)

{

thermometer.MeasureTheTemperature();

}

}

}

Thermometer.cs

namespace lab6.\_2.\_5

{

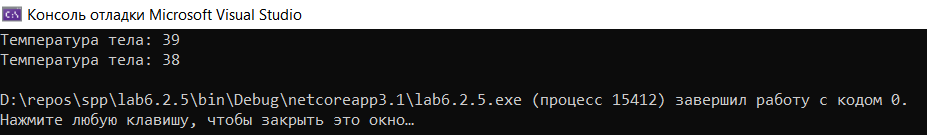
public interface Thermometer

{

void MeasureTheTemperature();

}

}



**Код программы (задание 3):**

ATM.cs

namespace lab6.\_3.\_5

{

public class ATM

{

private ATMState State;

public ATMState GetState()

{

return State;

}

public void SetState(ATMState state)

{

State = state;

}

public ATM(ATMState state)

{

State = state;

}

public void Expectation()

{

State.Expectation\_(this);

}

public void Authentication()

{

State.Authentication\_(this);

}

public void PerformingOperation()

{

State.PerformingOperation\_(this);

}

public void Blocking()

{

State.Blocking\_(this);

}

}

}

ATMState.cs

namespace lab6.\_3.\_5

{

public interface ATMState

{

void Expectation\_(ATM atm);

void Authentication\_(ATM atm);

void PerformingOperation\_(ATM atm);

void Blocking\_(ATM atm);

}

}

Authentication.cs

using System;

namespace lab6.\_3.\_5

{

public class Authentication : ATMState

{

Person person;

public Authentication(Person per)

{

person = per;

}

public void Expectation\_(ATM atm)

{

if (person.GetIsBlocked() == true)

{

Blocking\_(atm);

return;

}

Console.WriteLine("Ожидание ввода данных...");

}

public void Authentication\_(ATM atm)

{

if (person.GetIsBlocked() == true)

{

Blocking\_(atm);

return;

}

int pin, numTry = 0;

do

{

Console.WriteLine("Введите пароль:");

pin = Convert.ToInt32(Console.ReadLine());

if (person.GetPassword() == pin)

{

Console.WriteLine("Пароль введён верно!");

return;

}

else

{

Console.WriteLine("Пароль введён неверно!");

numTry++;

if (numTry == 3)

{

Blocking\_(atm);

return;

}

}

}

while (person.GetPassword() != pin);

}

public void PerformingOperation\_(ATM atm)

{

if (person.GetIsBlocked() == true)

{

Blocking\_(atm);

return;

}

Console.WriteLine("Введите сууму для снятия: ");

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

if (sum > person.GetBill())

{

Console.WriteLine("Недостаточно денег на счёте!");

}

if (sum <= person.GetBill())

{

person.SetBill(person.GetBill() - sum);

Console.WriteLine("Выдача денег...");

Console.WriteLine("Остаток на счёте: " + person.GetBill());

}

}

public void Blocking\_(ATM atm)

{

person.SetIsBlocked(true);

Console.WriteLine("Ваша карта заблокирована!");

}

}

}

Person.cs

namespace lab6.\_3.\_5

{

public class Person

{

string Name;

int Bill, Password;

bool isBlocked = false;

public Person(string name, int bill, int password)

{

Name = name;

Bill = bill;

Password = password;

}

public void SetName(string name)

{

Name = name;

}

public string GetName()

{

return Name;

}

public void SetBill(int bill)

{

Bill = bill;

}

public int GetBill()

{

return Bill;

}

public void SetPassword(int password)

{

Password = password;

}

public int GetPassword()

{

return Password;

}

public void SetIsBlocked(bool is\_Blocked)

{

isBlocked = is\_Blocked;

}

public bool GetIsBlocked()

{

return isBlocked;

}

}

}

Program.cs

namespace lab6.\_3.\_5

{

class Program

{

static void Main(string[] args)

{

Person Bob = new Person("Bob", 1000, 1234);

ATM atm = new ATM(new Authentication(Bob));

atm.Expectation();

atm.Authentication();

atm.PerformingOperation();

}

}

}



Вывод: в ходе выполнения лабораторной работы приобрел навыки применения паттернов проектирования при решении практических задач с использованием языка C#