Частное учреждение образования

«Колледж бизнеса и права»

Лабораторная работа № 16

Учащийся Е. А. Чужавко

группа: Т-992

2022

**Задание. Вариант 2**

Имеется фабрика по производству телефонов, планшетов и компьютеров трех различных торговых марок. Пользователь выбирает нужную ему категорию.

Результат работы программы представлен на рисунке 1.

package Apple;

import Interface.IAbstractFactory;

import Interface.IComputer;

import Interface.ITablet;

import Interface.Iphone;

public class AppleAbstractFactory implements IAbstractFactory {

@Override

public ITablet getTablet() {

return new AppleTablet();

}

@Override

public Iphone getPhone() {

return new ApplePhone();

}

@Override

public IComputer getComputer() {

return new AppleComputer();

}

}

package Apple;

import Interface.IComputer;

public class AppleComputer implements IComputer {

@Override

public String getComputerType() {

return "Компьютер марки Apple";

}

}

package Apple;

import Interface.Iphone;

public class ApplePhone implements Iphone {

@Override

public String getPhoneType() {

return "Телефон марки Apple";

}

}

package Apple;

import Interface.ITablet;

public class AppleTablet implements ITablet {

@Override

public String getTablet() {

return "Планшет марки Apple";

}

}

package Interface;

public interface IAbstractFactory {

ITablet getTablet();

Iphone getPhone();

IComputer getComputer();

}

package Interface;

public interface IComputer {

String getComputerType();

}

package Interface;

public interface Iphone {

String getPhoneType();

}

package Interface;

public interface ITablet {

String getTablet();

}

package Samsung;

import Apple.AppleTablet;

import Interface.IAbstractFactory;

import Interface.IComputer;

import Interface.ITablet;

import Interface.Iphone;

public class SamsungAbstractFactory implements IAbstractFactory {

@Override

public ITablet getTablet() {

return new SamsungTablet();

}

@Override

public Iphone getPhone() {

return new SamsungPhone();

}

@Override

public IComputer getComputer() {

return new SamsungComputer();

}

}

package Samsung;

import Interface.IComputer;

public class SamsungComputer implements IComputer {

@Override

public String getComputerType() {

return "Компьютер марки Samsung";

}

}

package Samsung;

import Interface.Iphone;

public class SamsungPhone implements Iphone {

@Override

public String getPhoneType() {

return "Телефон марки Samsung";

}

}

package Samsung;

import Interface.ITablet;

public class SamsungTablet implements ITablet {

@Override

public String getTablet() {

return "Планшет марки Samsung";

}

}

package Xiaomi;

import Interface.IAbstractFactory;

import Interface.IComputer;

import Interface.ITablet;

import Interface.Iphone;

import Samsung.SamsungComputer;

import Samsung.SamsungPhone;

import Samsung.SamsungTablet;

public class XiaomiAbstractFactory implements IAbstractFactory {

@Override

public ITablet getTablet() {

return new XiaomiTablet();

}

@Override

public Iphone getPhone() {

return new XiaomiPhone();

}

@Override

public IComputer getComputer() {

return new XiaomiComputer();

}

}

package Xiaomi;

import Interface.IComputer;

public class XiaomiComputer implements IComputer {

@Override

public String getComputerType() {

return "Компьютер марки Xiaomi";

}

}

package Xiaomi;

import Interface.Iphone;

public class XiaomiPhone implements Iphone {

@Override

public String getPhoneType() {

return "Телефрон марки Xiaomi";

}

}

package Xiaomi;

import Interface.ITablet;

public class XiaomiTablet implements ITablet {

@Override

public String getTablet() {

return "Планшет марки Xiaomi";

}

}

import Interface.IAbstractFactory;

import Interface.IComputer;

import Interface.ITablet;

import Interface.Iphone;

import Samsung.SamsungAbstractFactory;

import Xiaomi.XiaomiAbstractFactory;

import Apple.AppleAbstractFactory;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Введите номер фирмы: \n1-Samsung \n2-Apple \n3-Xiaomi");

int firm=0;

firm=in.nextInt();

switch (firm){

case (1):

System.out.println("Введите номер предмета: \n1-Телефон \n2-Планшет \n3-Компьютер");

int item=0;

item=in.nextInt();

IAbstractFactory factory=new SamsungAbstractFactory();

switch (item){

case (1):

Iphone phone=factory.getPhone();

System.out.println(phone.getPhoneType());

break;

case (2):

ITablet tablet=factory.getTablet();

System.out.println(tablet.getTablet());

break;

case (3):

IComputer comp=factory.getComputer();

System.out.println(comp.getComputerType());

break;

default:

System.out.println("Такого предмета нет");

break;

}

break;

case (2):

System.out.println("Введите номер предмета: \n1-Телефон \n2-Планшет \n3-Компьютер");

int item1=0;

item1=in.nextInt();

IAbstractFactory factory1=new AppleAbstractFactory();

switch (item1){

case (1):

Iphone phone=factory1.getPhone();

System.out.println(phone.getPhoneType());

break;

case (2):

ITablet tablet=factory1.getTablet();

System.out.println(tablet.getTablet());

break;

case (3):

IComputer comp=factory1.getComputer();

System.out.println(comp.getComputerType());

break;

default:

System.out.println("Такого предмета нет");

break;

}

break;

case (3):

System.out.println("Введите номер предмета: \n1-Телефон \n2-Планшет \n3-Компьютер");

int item2=0;

item2=in.nextInt();

IAbstractFactory factory2=new XiaomiAbstractFactory();

switch (item2){

case (1):

Iphone phonex=factory2.getPhone();

System.out.println(phonex.getPhoneType());

break;

case (2):

ITablet tabletx=factory2.getTablet();

System.out.println(tabletx.getTablet());

break;

case (3):

IComputer compx=factory2.getComputer();

System.out.println(compx.getComputerType());

break;

default:

System.out.println("Такого предмета нет");

break;

}

break;

default:

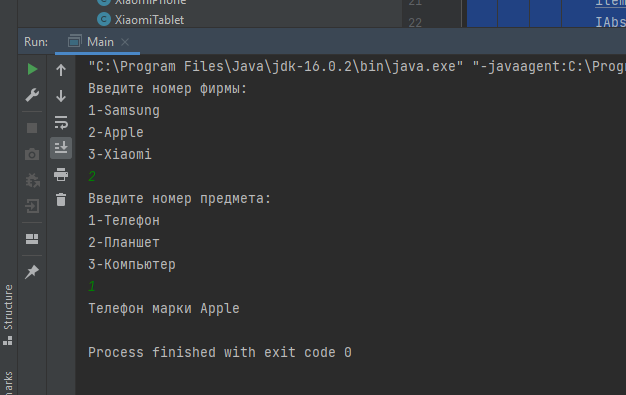
System.out.println("Такой фирмы нет");

break;

}

}

}



*Рисунок 1.*