НТУУ «КПИ»

ФИВТ

Кафедра ВТ

Лабораторная работа №6

Выполнил:

студент группы ИВ-34

Мозговой Иван

Проверил:

Невдащенко М. В.

Номер зачетной книжки:

3416

Киев 2014г.

Определение задания:

3416 mod 13 = 10 => создать иерархию тарифов мобильной компании.

Описание программы:

В данной программе используется 9 классов. 8 для каждой функции и один исполнительный. Работа каждого класса описана в коде.

Код программы:

**public** **class** mainLab5 {

/\*\*

\*

\* **@version** 1.1 29 May 2014

\* **@author** Ivan Mozgovyi

\*

\*/

/\*\*

\* Это исполнительный класс.

\*

\*/

**public** **static** **void** main(String[] args) {

System.*out*.println("3416%13="+3416%13);

Tariffs T = **new** Tariffs();

T.sortTariffsDecrease();

T.outP();

T.search(60);

T.numOfCustomers();

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для хранения ввода и вывода информации о тарифах.

\*

\*/

**public** **class** Atnt {

**private** **int** payPerMonth;

**private** **double** textPrice;

**private** **double** talkPrice;

**private** **int** internetPrice;

**private** **int** customersOnThisTariff;

**private** String name;

**public** Atnt(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**this**.customersOnThisTariff=customersOnThisTariff;

**this**.internetPrice=internetPrice;

**this**.name=name;

**this**.payPerMonth=payPerMonth;

**this**.talkPrice=talkPrice;

**this**.textPrice=talkPrice;

}

**public** **double** getPriceTalk(){

**return** talkPrice;

}

**public** **double** getPriceText(){

**return** textPrice;

}

**public** **double** getPriceInternet(){

**return** internetPrice;

}

**public** **int** getCustomers(){

**return** customersOnThisTariff;

}

**public** **int** getPricePerMonth(){

**return** payPerMonth;

}

**public** String getNameOfTariff(){

**return** name;

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для оплаты тарифов.

\*

\*/

**public** **class** EverythingPayed **extends** Atnt {

**public** EverythingPayed(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**super**(payPerMonth, textPrice, talkPrice, internetPrice, customersOnThisTariff, name);

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для вывода и обработки информации о тарифах и клиентах.

\*

\*/

**public** **class** Tariffs {

**private** Atnt atntCompany[];

**public** Tariffs(){

atntCompany=**new** Atnt[]{

**new** EverythingPayed(0, 0.10, 0.15, 10, 5000, "Nothing Unlim"),

**new** UnlimInternetTalkText(60, 0, 0, 0, 35000, "Unlim Talk+Text+Internet"),

**new** unlimTalkPayedTextInternet(10, 0.10, 0.15, 10, 10000, "Unlim Talk"),

**new** UnlimTextPayedTalkInternet(5, 0, 0.15, 10, 2000, "Unlim text" ),

**new** UnlimTextTalkPayedInternet(20, 0, 0, 15, 6000, "Unlim Talk+Text"),

};

}

/\*\*

\* Output the table of all possible tariffs

\*/

**public** **void** outP(){

System.*out*.format("======================================================================================\n");

System.*out*.printf("| Plan name |Cost PM|Inter. 1Gig |Text 1msg |Talk 1min |Customers |\n");

System.*out*.format("======================================================================================\n");

String leftAlighName="| %-28s |";

String leftAlignCost=" %-5d |";

String leftAlignPrices=" %-10s |";

String leftAlignText="%-9s |";

**for**(**int** i=0; i<atntCompany.length; i++){

System.*out*.format(leftAlighName, atntCompany[i].getNameOfTariff());

System.*out*.format(leftAlignCost, atntCompany[i].getPricePerMonth());

String internetPrice = String.*format*("%.2f", atntCompany[i].getPriceInternet());

System.*out*.format(leftAlignPrices, internetPrice);

String textPrice = String.*format*("%.2f", atntCompany[i].getPriceText());

System.*out*.format(leftAlignText, textPrice);

String talkPrice = String.*format*("%.2f", atntCompany[i].getPriceTalk());

System.*out*.format(leftAlignText, talkPrice);

System.*out*.format(leftAlignText, atntCompany[i].getCustomers());

System.*out*.println();

}

System.*out*.format("======================================================================================\n");

}

/\*\*

\* Find every tariff that is less than entered price

\* **@param** price - max price you are ready to pay

\*/

**public** **void** search(**int** price){

**if**(price<0){

System.*out*.println("Wrong price entered. ");

}**else**{

System.*out*.println("Tariffs that are in your price range: ");

**boolean** check=**false**;

**for**(**int** i=0; i<atntCompany.length; i++){

**if**(atntCompany[i].getPricePerMonth()<=price){

System.*out*.println(atntCompany[i].getNameOfTariff());

check=**true**;

}

}

**if**(check==**false**){

System.*out*.println("None found.");

}

}

}

/\*\*

\* Add all the customers that use different tariffs to find the total amount of customers

\*/

**public** **void** numOfCustomers(){

System.*out*.print("Total amount of customers: ");

**int** temp=0;

**for**(**int** i=0; i<atntCompany.length; i++){

temp+=atntCompany[i].getCustomers();

}

System.*out*.println(temp);

}

/\*\*

\* Sort tariffs prices per month in increasing order

\*/

**public** **void** sortTariffsIncrease(){

Atnt temp;

**for**(**int** i=0; i<atntCompany.length; i++){

**for**(**int** j=i+1; j<atntCompany.length; j++){

**if**(atntCompany[j].getPricePerMonth()<=atntCompany[i].getPricePerMonth()){

temp=atntCompany[i];

atntCompany[i]=atntCompany[j];

atntCompany[j]=temp;

}

}

}

}

/\*\*

\* Sort tariffs prices per month in decreasing order

\*/

**public** **void** sortTariffsDecrease(){

Atnt temp;

**for**(**int** i=0; i<atntCompany.length; i++){

**for**(**int** j=i+1; j<atntCompany.length; j++){

**if**(atntCompany[j].getPricePerMonth()>=atntCompany[i].getPricePerMonth()){

temp=atntCompany[i];

atntCompany[i]=atntCompany[j];

atntCompany[j]=temp;

}

}

}

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для хранения тарифа UnlimInternetTalkText

\*

\*/

**public** **class** UnlimInternetTalkText **extends** Atnt{

**public** UnlimInternetTalkText(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**super**(payPerMonth, textPrice, talkPrice, internetPrice, customersOnThisTariff, name);

}

}

/\*\*

\*

\* **@author** Ivan

\*класс тарифа для хранения тарифа unlimTalkPayedTextInternet.

\*

\*/

**public** **class** unlimTalkPayedTextInternet **extends** Atnt{

**public** unlimTalkPayedTextInternet(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**super**(payPerMonth, textPrice, talkPrice, internetPrice, customersOnThisTariff, name);

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для хранения тарифа UnlimTextPayedTalkInternet.

\*

\*/

**public** **class** UnlimTextPayedTalkInternet **extends** Atnt{

**public** UnlimTextPayedTalkInternet(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**super**(payPerMonth, textPrice, talkPrice, internetPrice, customersOnThisTariff, name);

}

}

/\*\*

\*

\* **@author** Ivan

\* класс для хранения тарифа UnlimTextTalkPayedInternet.

\*

\*/

**public** **class** UnlimTextTalkPayedInternet **extends** Atnt{

**public** UnlimTextTalkPayedInternet(**int** payPerMonth, **double** textPrice, **double** talkPrice, **int** internetPrice, **int** customersOnThisTariff, String name){

**super**(payPerMonth, textPrice, talkPrice, internetPrice, customersOnThisTariff, name);

}

}

Выводы:

Данная лабораторная работа демонстрирует работу с классами. В этой лабораторной работе были созданы 9 классов. Один исполнительный и 8 для каждой определенной функции. Были использованы конструкторы для определения полей, которые являлись характеристиками каждого тарифа, с директивой private для инкапсуляции. Программа демонстрирует также объектно-ориентированный принцип программирования как полиморфизм.