Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра автоматизованих систем управління



**Звіт**

до лабораторної роботи №6

з дисципліни

​*Прикладне програмування*

на тему:

*‘Реалізація функціоналу програми’*

Виконав: студент КН-207

**Безменський Іван**

Прийняв: доцент каф. АСУ

**Пелех Ю.М.**

**Львів – 2022**

***Лабораторна робота №6***

• Використовувати можливості ООП: класи, успадкування, поліморфізм, інкапсуляцію.

• Кожний клас повинен мати назву, яка повністю описує його суть, і інформативний склад.

• Успадкування потрібно використовувати тільки тоді, коли воно має сенс.

• При записі програми потрібно використовувати домовленості щодо оформлення коду java code convention.

• Класи потрібно грамотно розкласти по пакетах.

• Для збереження параметрів ініціалізації потрібно використовувати файли.

• Повинне бути консольне меню для вибору різних дій, наприклад: «завантажити з файлу», «довідка», «вихід» тощо.

*Реалізуйте функціонал програми. Обов’язково зверніть увагу на додаткові вимоги до програми*

***Варіант – 18***

**Кредити.** Сформувати набір пропозицій клієнту по цільових кредитах різних банків для оптимального вибору. Враховувати можливість дострокового погашення кредиту і/або збільшення кредитної лінії. Реалізувати вибір і пошук кредиту за певними параметрами.

**Код програми:**

**Menu.java**

package App.Menu;  
  
import App.Bank.Bank;  
import App.Commands.\*;  
import java.util.Scanner;  
import java.util.HashMap;  
import java.util.Map;  
  
public class Menu {  
 private static Bank *bank*;  
 private static Map<String, BaseCommand> *userChoose*;  
  
 public static void start() throws Exception {  
 *userChoose* = new HashMap<>();  
 *userChoose*.put("createbank", new BankCreateCommand());  
 *userChoose*.put("showbank", new ShowBanksCommand());  
 *userChoose*.put("createcredit", new CreateCreditCommand());  
 *userChoose*.put("deletecredit", new DeleteCreditCommand());  
 *userChoose*.put("changecredit", new ChangeCreditCommand());  
 *userChoose*.put("showcredit", new ShowCreditCommand());  
 *userChoose*.put("findcredit", new FindCreditCommand());  
 *userChoose*.put("writeinfile", new WriteToFileCommand());  
 *userChoose*.put("readfromfile", new WriteToFileCommand());  
 *userChoose*.put("exit", new ExitCommand());  
  
 *bank* = new Bank();  
 while (true){  
 *getMenu*();  
 }  
 }  
  
 public static void getMenu() throws Exception {  
 System.*out*.print("""  
 Список всіх доступних команд:  
 \t[1] Вибрати Банк - createbank  
 \t[2] Показати Банк - showbank  
 \t[3] Створити Кредит - createcredit  
 \t[4] Видалити Кредит - deletecredit  
 \t[5] Змінити Кредит - changecredit  
 \t[6] Показати Кредити - showcredit  
 \t[7] Знайти Кредити - findcredit  
 \t[8] Запис у файл - writeinfile  
 \t[9] Зчитування з файлу - readfromfile  
 \t[10] Вихід - exit  
 Введіть ваш вибір:\s""");  
 *userChoose*();  
 }  
  
 private static void userChoose() throws Exception {  
 Scanner scanner = new Scanner(System.*in*);  
 String Input = scanner.nextLine();  
  
  
  
 BaseCommand command = *userChoose*.get(Input);  
 if (command != null) {  
 *userChoose*.get(Input).execute(*bank*);  
 } else {  
 throw new IllegalArgumentException("Ви ввели невірну команду!");  
 }  
 }  
}

**Bank.java**

package App.Bank;  
  
import App.Credit.Credit;  
import App.User.User;  
  
import java.util.Random;  
import java.util.Scanner;  
import java.util.ArrayList;  
import java.util.Objects;  
import java.io.FileNotFoundException;  
import java.io.FileReader;  
import java.io.BufferedReader;  
import java.io.IOException;  
  
public class Bank {  
 Scanner scanner = new Scanner(System.*in*);  
  
 private String bankName;  
 private double creditPercent;  
 private int maxTimeForRepayment;  
  
 protected User user = new User();  
 public ArrayList<Credit> creditList = new ArrayList<>();  
  
  
 public double getCreditPercent() {return creditPercent;}  
 public int getMaxTimeForRepayment() {return maxTimeForRepayment;}  
 public ArrayList<Credit> getCredit() {return creditList;}  
 public String getBankName() {return bankName;}  
 public User getUser() {return user;}  
  
 public int getRandomNumber(int low, int high) {  
 Random random = new Random();  
 return random.nextInt(high-low) + low;  
 }  
  
  
 public Bank banksSet() {  
 Bank temp = new Bank();  
 int name = getRandomNumber(1, 3);  
 int creditPercent = getRandomNumber(1, 3);  
 int MaxTimeForRepayment = getRandomNumber(1, 3);  
 if (name == 1) {  
 temp.bankName = "Privat24";  
 }  
 if (name == 2) {  
 temp.bankName = "MonoBank";  
 } else if (name == 3) {  
 temp.bankName = "Oshchad24";  
 }  
 if (creditPercent == 1) {  
 temp.creditPercent = 1.5;  
 }  
 if (creditPercent == 2) {  
 temp.creditPercent = 3.2;  
 } else if (creditPercent == 3) {  
 temp.creditPercent = 1.7;  
 }  
 if (MaxTimeForRepayment == 1) {  
 temp.maxTimeForRepayment = 12;  
 }  
 if (MaxTimeForRepayment == 2) {  
 temp.maxTimeForRepayment = 24;  
 } else if (MaxTimeForRepayment == 3) {  
 temp.maxTimeForRepayment = 6;  
 }  
 return temp;  
 }  
 public void createBanks(){  
 System.*out*.println("Введтіь логін користувача: ");  
 this.user.setLogin(scanner.next());  
 System.*out*.println("Введтіь пароль користувача: ");  
 this.user.setPassword(scanner.next());  
 ArrayList<Bank> banks = new ArrayList<>();  
 Bank firstBank = banksSet();  
 banks.add(firstBank);  
 Bank secondBank = banksSet();  
 banks.add(secondBank);  
 Bank thirdBank = banksSet();  
 banks.add(thirdBank);  
 for (int i = 0; i < banks.size(); i++){  
 System.*out*.println("Назва Банку: " + banks.get(i).getBankName() + "\nМісячний відсоток: " + banks.get(i).getCreditPercent()  
 + "\nМаксимальний строк кредиту: " + banks.get(i).getMaxTimeForRepayment() + "\n");  
 }  
 bankChoose(banks);  
 }  
 public void showBanks(){  
 System.*out*.println("Назва Банку: " + this.getBankName() + "\nМісячний відсоток: " + this.getCreditPercent()  
 + "\nМаксимальний строк кредиту: " + this.getMaxTimeForRepayment() + "\n");  
 }  
  
 public void bankChoose(ArrayList<Bank> banks) {  
 System.*out*.println("Виберіть з яким банком хочете працювати(По номеру): ");  
 int choose = scanner.nextInt();  
  
 switch (choose) {  
 case (1):  
 this.bankName = banks.get(0).bankName;  
 this.creditPercent = banks.get(0).creditPercent;  
 this.maxTimeForRepayment = banks.get(0).maxTimeForRepayment;  
 break;  
 case (2):  
 this.bankName = banks.get(1).bankName;  
 this.creditPercent = banks.get(1).creditPercent;  
 this.maxTimeForRepayment = banks.get(1).maxTimeForRepayment;  
 break;  
 case (3):  
 this.bankName = banks.get(2).bankName;  
 this.creditPercent = banks.get(2).creditPercent;  
 this.maxTimeForRepayment = banks.get(2).maxTimeForRepayment;  
 break;  
 default:  
 System.*out*.println("Невірний банк");  
 }  
 }  
  
 public void printCredits(){  
 for (int i = 0; i < creditList.size(); i++){  
 System.*out*.println("\n[" + (i+1) + "]");  
 creditList.get(i).toString();  
 }  
 }  
  
 public void createCredit(){  
 System.*out*.println("Введіть назву кредиту: ");  
 String inputName = scanner.next();  
  
 System.*out*.println("Введіть кількість отриманих коштів: ");  
 int inputMoney = scanner.nextInt();  
  
 System.*out*.println("Введіть час на поверненн: ");  
 int inputTime = scanner.nextInt();  
  
 creditList.add(new Credit(inputName, inputMoney, inputTime));  
 }  
  
 public void changeCredit() {  
 System.*out*.println("Введіть номер кредиту який хочете змінити: ");  
 int index = scanner.nextInt();  
  
 System.*out*.println("Введіть назву кредиту: ");  
 String inputName = scanner.next();  
  
 System.*out*.println("Введіть кількість отриманих коштів: ");  
 int inputMoney = scanner.nextInt();  
  
 System.*out*.println("Введіть час на повернення: ");  
 int inputTime = scanner.nextInt();  
  
 creditList.set(index - 1, new Credit(inputName, inputMoney, inputTime));  
 }  
  
 public void deleteCredit(){  
 System.*out*.println("Бажаєте вивести всі доступні кредити? [1 - Так/2 - Ні] ");  
 int choose = scanner.nextInt();  
  
 if (choose == 1){  
 printCredits();  
 }  
  
 System.*out*.println("Введіть номер кредиту який хочете видалити: ");  
 int index = scanner.nextInt();  
 if (index <= creditList.size()){  
 creditList.remove(index - 1);  
 } else {  
 System.*out*.println("Такого кредиту не має.");  
 }  
 }  
  
 public void findCredit(){  
 System.*out*.println("""  
 Виберіть за яким параметром знайти кредит:   
 [1] - Назва Кредиту. [2] - Кількість грошей. [3] - Час на сплату.  
 """);  
 int choose = scanner.nextInt();  
  
 if (choose == 1){  
 firstSearch();  
 }  
 if (choose == 2){  
 secondSearch();  
 }  
 if (choose == 3){  
 thirdSearch();  
 }  
 else{  
 System.*out*.println("Такого параметру немає!");  
 }  
 }  
  
 public void firstSearch() {  
 System.*out*.println("Введіть за якою назвою робити пошук: ");  
 String searchingName = scanner.next();  
  
 for (int i = 0; i < creditList.size(); i++) {  
 if (Objects.*equals*(creditList.get(i).getName(), searchingName)) {  
 creditList.get(i).toString();  
 } else {  
 System.*out*.println("Кредита з такою назвою не існує!");  
 }  
 }  
 }  
 public void secondSearch(){  
 System.*out*.println("Введіть за якою сумою \*ціле число\* робити пошук(Спочатку мінімум, потім максимум): ");  
 int lowerLimit = scanner.nextInt();  
 int upperLimit = scanner.nextInt();  
 //int plusMinusSpread = 1000; //Значення розбросу плюс мінус  
  
 for (int i = 0; i < creditList.size(); i++){  
 if (creditList.get(i).getMoneyAmount() >= lowerLimit /\*- plusMinusSpread\*/ && creditList.get(i).getMoneyAmount() <= upperLimit /\*+ plusMinusSpread\*/){  
 creditList.get(i).toString();  
 }  
 }  
 }  
  
 public void thirdSearch(){  
 System.*out*.println("Введіть час \*у місяцях\* за яким робити пошук(Спочатку мінімум, потім максимум): ");  
 int lowerLimit = scanner.nextInt();  
 int upperLimit = scanner.nextInt();  
  
 for (int i = 0; i < creditList.size(); i++){  
 if (creditList.get(i).getTimeForRepayment() >= lowerLimit && creditList.get(i).getTimeForRepayment() <= upperLimit ){  
 creditList.get(i).toString();  
 }  
 }  
 }  
  
 public void readFromFile() {  
 BufferedReader readingTool;  
 try {  
 readingTool = new BufferedReader(new FileReader("Banks\_Data.txt"));  
 String inputData = readingTool.readLine();  
 while (inputData != null){  
 this.bankName = inputData;  
 inputData = readingTool.readLine();  
 this.creditPercent = Double.*parseDouble*(inputData);  
 inputData = readingTool.readLine();  
 this.maxTimeForRepayment = Integer.*parseInt*(inputData);  
 inputData = readingTool.readLine();  
 this.user.setLogin(inputData);  
 inputData = readingTool.readLine();  
 this.user.setPassword(inputData);  
 inputData = readingTool.readLine();  
 }  
 readingTool.close();  
  
 readingTool = new BufferedReader(new FileReader("Credits\_Data.txt"));  
 String inputData2 = readingTool.readLine();  
 while(inputData2 != null){  
 String name = inputData2;  
 inputData2 = readingTool.readLine();  
 int moneyAmount = Integer.*parseInt*(inputData2);  
 inputData2 = readingTool.readLine();  
 int timeForRepayment = Integer.*parseInt*(inputData2);  
 inputData2 = readingTool.readLine();  
 creditList.add(new Credit(name, moneyAmount, timeForRepayment));  
 }  
 readingTool.close(); }  
 catch (FileNotFoundException e) {throw new RuntimeException(e);}  
 catch (IOException e) {throw new RuntimeException(e);}  
 }  
  
  
}

**User.java**

package App.User;  
  
  
public class User {  
 private String login;  
 private String password;  
  
 public void setLogin(String login) {  
 this.login = login;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public String getLogin() {  
 return login;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
}

**Credit.java**

package App.Credit;  
  
public class Credit {  
 protected String name;  
 protected int moneyAmount;  
 protected int timeForRepayment;  
  
 public Credit(String name, int moneyAmount, int timeForRepayment) {  
 this.name = name;  
 this.moneyAmount = moneyAmount;  
 this.timeForRepayment = timeForRepayment;  
  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getMoneyAmount() {  
 return moneyAmount;  
 }  
  
 public int getTimeForRepayment(){  
 return timeForRepayment;  
 }  
  
 @Override  
 public String toString() {  
 System.out.println("Назва Кредиту: " + getName() +  
 "\nКількість отриманих коштів : " + getMoneyAmount() +  
 "\nЧас на повернення: " + getTimeForRepayment());  
 return "";  
 }  
}

**Висновок:** По закінченню даної лабораторної роботи я розробив функціонал програми.