Sofia University
Department of Mathematics and Informatics

Course : OO Programming Java
Date: January 8, 2010
Student Name:

Lab No. 12

Submit the all Java files developed to solve the problems listed below. Use comments and Modified-Hungarian notation.

Exercises on Client- Server

Задача 1

Create a server that asks for a **password**, then **opens a file** and **sends the file** over the network connection. **Create a client** that **connects to this server**, gives the appropriate **password**, then **captures** and **saves** the **file**.

Test the pair of programs on your machine using the **localhost** (the local loopback IP address **127.0.0.1** produced by calling **InetAddress.getByName("127.0.0.1")**). Write **JFrame** applications that implement the client and the server. See the **`fig24_05_08.rar`** sample code given in lectures.

Задача 2

Develop a **client / server application** to **add** addresses using a **`JFrame`** form at the client side. Declare an **`Address`** class to hold name, **`street`**, **`city`**, **`state`** and **`zip`** in an object. Allow the **client to create `Address`** objects and **send them to the server** using a button "**`Add`**". On receiving an **`Address`** object the server (console application) should **serialize the objects** to a file named **address.ser**.

Задача 3

Develop a **client / server application** to **read addresses** using a JFrame form at the client side. Declare an **`Address class`** to hold **`name`**, street, **`city`**, **`state`** and **`zip`** in an object. Allow the client **to send the name** of the file to the server. In return the **server should send `Address`** objects to the client to be displayed in the **`JFrame`** form at the client side objects On receiving an **`Address`** object the client should **`deserialize`** the objects **before displaying them in the form**.

Задача 4

Разплащателният и спестовният влог са типични примери за **банкови сметки**. ***Внасянето***, ***депозирането*** и ***трансфера*** на парични суми са най- често срещаните трансакции (операции) с банкови сметки, при което

трансакциите *се съхраняват във файл*. Създайте графично приложение на Java по **MVC модела**, което демонстрира тези операции **в описаната по- долу последователност:**

1. **Напишете дефиницията на *class BankAccount*.** Този клас има две променливи (*data members*) (***int intAccNo*** и ***double dblBalance***) и следните методи -
   a) ***double getBalance()->*** връща текущия ***dblBalance***
   b) ***void deposit(double amt)->*** добавя ***amt*** към ***dblBalance***
   c) ***void withdraw(double amt)->*** изважда ***amt*** от ***dblBalance***

   Напишете конструктор, за да инициализирате променливите ***intAccNo*** и ***dblBalance.(dblBalance = 0.0*** by default***)***

   10 точки

2. **Напишете дефиницията на *class UnsufficientFunds,*** който **онаследява *class Exception*. Напишете *general purpose* конструктор** и **използвайте това изключение (*exception*),** за да сигнализирате във функцията ***withdraw()*** на ***class BankAccount*** (*посредством* ***JOptionPane.showMessageDialog()*** в ***try{}catch(){}*** блок***)*** случаите, когато ***amt> dblBalance*** .

   8 точки

3. **Нека *всяка* банкова сметка да бъде *Transferable,* т.е.** да може да прехвърля сума от своя баланс на баланса на друга банкова сметка. **Напишете дефиницията на *interface Transferable,*** който съдържа единствено следния метод

   void transfer(Transferable b, double amt) **{**
   // прехвърля ***amt*** **от баланса на** текущата банкова сметка **в *Transferable b***
   }

   Нека *class BankAccount* онаследява *interface Transferable*

   8 точки

4. **Напишете дефиницията на *class SavingAcc,*** който **онаследява *class BankAccount*.**: В допълнение към онаследените променливи този клас има още ***една*** променлива (data member)- ***double dblIntRate***, която е лихвата по спестовния влог и **тя е една и съща (*static*) за всички обекти** от ***class SavingAcc***. **Напишете *general purpose* конструктор и *toString*()** метод за този клас. Използвайте ***String.format() в toString*()** метода.

   6 точки

5. **Напишете дефиницията на *class CreditCardAcc,*** който **онаследява *class BankAccount*.**: В допълнение към онаследените променливи този клас има още ***една*** променлива (data member)- ***int intTransCount***, която е поредният номер на трансакция и е ***static*** за ***class CreditCardAcc***. **Напишете *general purpose* конструктор и *toString*()** метод за този клас. Използвайте ***String.format()***

   6 точки

6. **Дефинирайте** функцията ***transfer(Transferable b, double amt)*** в ***class SavingAcc*** така че да изпълнявя следния алгоритъм:

```
//1) add the product dblBalance * dblIntRate to dblBalance
//2) withdraw amt from the current dblBalance
//3) deposit amt to the current dblBalance of b
If (b is a CreditCardAcc) then
{
      //4) increment intTransCount of b
}
```

   8 точки

7. **Дефинирайте** функцията ***transfer(Transferable b, double amt)*** в ***class CreditCardAcc,*** така че да изпълнява следния алгоритъм:

```
If(b is a SavingAcc) then
{
      //1) add the product dblBalance * dblIntRate to dblBalance
}
//2) increment intTransCount
//3) withdraw amt from the current dblBalance
//4) deposit amt to the current dblBalance of b
```

8 точки

8. **Напишете дефиницията на *class BankUI,*** който **онаследява *class JPanel.***: Този клас има ***следните*** данни (data members)- референция ***labels*** към **масив** от ***JLabel***, референция ***fields*** към **масив** от ***JTextField,*** референция ***btnOne*** към ***JButton*** и референции към два ***JPanel*** обекта- ***southPanel*** и ***centerPanel***. **Инициализирайте**, тези данни **посредством конструктор**, който приема за аргумент **масив** от ***String[]*** обекти, определящи *текста* на обектите ***JLabel*** от масива ***labels***. **Дължината на този масив съвпада с броят на <u>обектите</u>** в масивите ***labels*** и ***fields***. Разположете **елементите на *labels*** и ***fields*** в две колони в ***centerPanel като използвате Gridlayout().*** Разположете **<u>обекта</u> *btnOne в southPanel***. ***Самите centerPanel*** и ***southPanel*** разположете съответно в центъра и долната част на основния панел като използвате ***BorderLayout() и накрая*** добавете ***команда за обновяване на панела .***

15 точки

9. **Напишете дефиницията на следните методи на *class BankUI:***

- **a)** ***JButton getOne() ->*** връща референция към **обекта *btnOne***
- **b)** ***String[] getFields()->***връща ***текст- полетата*** на ***fields[]***
- **c)** ***void setFields(String s[])->***задава ***текст- полетата*** на ***fields[]***

7 точки

10. **Напишете дефиницията на главната форма на приложението *class FrmMain,*** който **онаследява *class JFrame***. Този клас има ***следните*** данни (data members)- референция към масив ***arrAccounts*** от **50** референции към ***Transferable*** обекти, референция ***output*** към ***ObjectOutputStream***, референции ***creditUI*** и ***savingUI*** към ***class BankUI*** и референции ***btnCUISave, btnSUISave*** и ***btnFile*** към ***class JButton***. В конструктора на ***class FrmMain:***

- **a)** ***създайте creditUI и savingUI*** обекти посредством следните ***String*** масиви ***{"Acc No", "Balance","Trans No.", "Transfer"}*** и ***{"Acc No", "Balance","Interest", "Transfer"}. Добавете*** тези панели към контекста на ***FrmMain***
- **b)** инициализирайте референциите ***btnCUISave*** и ***btnSUISave*** да указват ***обекта btnOne*** съответно от панелите ***creditUI*** и ***savingUI. Задайте*** етикети ***"Create CreditAcc"*** и ***"Create SavingAcc"*** за тези бутони. **При натискане на тези бутони** се създава съответно обект от ***class CreditCardAcc и class SavingAcc, който се добавя*** към масива ***arrAccounts.*** За целта се използвайте текста въведен в текстовите полета на ***creditUI*** и ***savingUI***
- **c)** **инициализирайте *btnFile*** да има етикет ***"To File"*** и **запишете сериализирано** във файл ***"bank.ser"*** банковите сметки събрани в масива ***arrAccounts*** на всеки елемент от този масив при натискане на ***btnFile.***
- **d)** ***Затворете "bank.ser" при затваряне на графичния прозорец***

24 точки