

Zadanie 1 (max. 4 pkt.)

Konta bankowe

Napisać program, który symuluje operacje na kontach bankowych. Konta są obiektami klasy Account. Każde konto ma aktualny stan (balance) i można:

- wpłacać pieniądze (metoda deposit())
- wypłacać pieniądze (metoda withdraw())
- przelewać na inne konto (metoda transfer(...))
- dodawać do stanu konta odsetki w skali rocznej (metoda addInterest()).

Stopa oprocentowania jest wspólna dla wszystkich kont i ustalana za pomocą metody setInterestRate(...).

Konta posiadają klienci banku (obiekty klasy BankCustomer). W roli klienta występuje osoba (obiekt klasy Person).

Dostarczyć takich definicji w/w klas, aby poniższy program:

```
public class BankingTest {  
  
    public static void main(String[] args) {  
  
        Person janP = new Person("Jan"),  
            alaP = new Person("Ala");  
  
        BankCustomer jan = new BankCustomer(janP);  
        BankCustomer ala = new BankCustomer(alaP);  
  
        jan.getAccount().deposit(1000);  
        ala.getAccount().deposit(2000);  
        jan.getAccount().transfer(ala.getAccount(), 500);  
        ala.getAccount().withdraw(1000);  
  
        System.out.println(jan);  
        System.out.println(ala);  
  
        Account.setInterestRate(4.5);  
        jan.getAccount().addInterest();  
        ala.getAccount().addInterest();  
  
        System.out.println(jan);  
        System.out.println(ala);  
    }  
}
```

wyprowadził następującą informację na standardowe wyjście (out):

Klient: Jan stan konta 500.0

Klient: Ala stan konta 1500.0

Klient: Jan stan konta 522.5

Klient: Ala stan konta 1567.5

Postać wydruku jest obowiązkowa.

Program powinien dobrze działać dla dowolnych wpłat-wypłat. To znaczy, że trzeba się zabezpieczyć przed ich niedopuszczalnymi wartościami (np. ujemne wartości lub też wypłaty przekraczające stan konta). Najprościej będzie ignorować niedopuszczalne operacje.

Jeśli natomiast będą zgłaszane wyjątki - to koniecznie trzeba je obsłużyć. Przy obsłudze nie wolno wypisywać komunikatów do standardowego strumienia błędów *err* (co najwyżej na standardowe wyjście - *out*).

Uwaga: utworzonej przez generator projektów klasy BankingTest nie wolno zmieniać.

Zadanie 2 (max. 6 pkt.)

Zadanie: Finder (6 punktów)

Napisać program, który wczytuje plik o nazwie "Test.java" z katalogu {user.home} z kodem źródłowym Javy (poprawnym, czyli kompilującym się bez błędów) i wyszukuje w nim instrukcje *if* oraz napisy "wariant". Wyniki działania programu w postaci:

Liczba instrukcji if: n

Liczba napisów wariant: m

należy wypisać na konsoli.

Obowiązkowa klasa Main, służąca do testowania programu winna mieć następującą postać:

```
public class Main {  
  
    public static void main(String ... args) throws Exception {  
        String home = System.getProperty("user.home");  
        Finder finder = new Finder(home + "/Test.java");  
        int nif = finder.getIfCount();  
        int nwar = finder.getStringCount("wariant");  
        System.out.println("Liczba instrukcji if: " + nif);  
        System.out.println("Liczba napisów wariant: " + nwar);  
    }  
}
```

Uwagi:

1. Wszystkie klasy w programie winny być publiczne (i w oddzielnych plikach)
2. Klasy Main (utworzonej przez generator projektów) nie wolno w żaden sposób modyfikować
3. Plik winien znajdować się w katalogu {user.home} czyli tym który zwróci wywołanie

```
System.getProperty("user.home");
```

4. Plik tworzymy sobie sami .
5. Proszę nie dołączać pliku do projektu.
6. Forma wydruku na konsoli jest obowiązkowa.

Za niespełnienie tych warunków nie będą przyznawane punkty.

Pomoc:

instrukcja if ma swoje wymagania składniowe (ale zakładamy, że program jest

poprawny). Nie jest instrukcją if coś co tak wygląda, ale jest umieszczone w komentarzu lub w literale napisowym.

Zadanie 3 (max. 5 pkt.)

Napisać program, który z pliku `{user.home}/tab.txt` wczytuje liczby całkowite (liczby w pliku są rozdzielone dowolnymi białymi znakami) i tworzy tablicę, zawierającą te liczby.

W tak zainicjowanej tablicy program znajduje wartość maksymalną oraz wszystkie indeksy w tablicy gdzie taka wartość się znajduje.

Program wypisuje na konsoli:

w pierwszym wierszu - wszystkie elementy tablicy, rozdzielone spacjami
w drugim wierszu - wartość maksymalną,
w trzecim wierszu - indeksy tablicy gdzie ta wartość się znajduje.

Przykładowo, jeśli plik `{user.home}/tab.txt` zawiera:

1 5 5 3

-1 2 5 4

to na konsoli otrzymujemy:

1 5 5 3 -1 2 5 4

5

1 2 6

W przypadku wystąpienia błędów (np. brak pliku, wadliwe dane) należy obsłużyć błąd - wypisać na konsoli TYLKO trzy gwiazdki:

i zakończyć działanie programu.

Nazwa pliku, jego umiejscowienie i postać oraz postać wyniku na konsoli jest obowiązkowa. Rozwiązanie nie spełniające tych wymagań otrzymują 0 punktów. Uwaga: proszę w żadnym razie nie linkować pliku jako zasobu do projektu Eclipse.

Uwagi dodatkowe:

- należy się przygotować na to, że w pliku jest dowolna ilość liczb,
- wobec tego należy użyć klasy `ArrayList` do wczytywania danych, ale wymaganie stworzenia tablicy też jest obowiązkowe.

Utworzona przez generator projektów klasa `Main` zawiera fragment pomocny dla uzyskania wymaganej nazwy pliku.

Klasę `Main` należy uzupełnić, tak, aby uzyskać właściwe rozwiązanie.

Zadanie 4 (max. 10 pkt.)

Wczytać tekst z pliku `{user.home}/textforwords.txt` i policzyć liczbę wystąpień poszczególnych słów.

Do zliczania wystąpień przygotować klasę `CountWords` w taki sposób, aby następujący program, zapisany w pliku `Main.java`:

```
import java.util.*;

public class Main {

    public static void main(String[] args) throws Exception {
        String fname = System.getProperty("user.home") +
"/textforwords.txt";
        CountWords cw = new CountWords(fname);
        List<String> result = cw.getResult();
        for (String wordRes : result) {
            System.out.println(wordRes);
        }
    }
}
```

wyprowadził w kolejnych wierszach kolejne słowa (w kolejności ich występowania w pliku) z następującą po spacji liczbą ich wystąpień.

Na przykład dla tekstu z pliku:

abc def abc ghi def

na konsoli winniśmy zobaczyć następujące wiersze:

abc 2
def 2
ghi 1

Uwagi:

1. Pliku `Main.java` nie wolno modyfikować.
2. Nazwa i umiejscowienie (katalog) pliku z tekstem są obowiązkowe
3. Słowa - to ciągi znaków rozdzielone białymi znakami i/lub znakami interpunkcji