

Introduction to Computers and Engineering Problem Solving 1.00 / 1.001 Fall 2005

Problem Set 2

Due: 11AM, Friday September 23, 2005

Banking & Investment [100 points]

Introduction

Since the stock markets and other types of investment are relatively difficult to model, we'll be modeling investments as bank accounts. For the purposes of this problem set, we'll assume there are only three types of investment accounts, tabulated below.

| | Investment Type | | |
|---------------------------|-----------------|---------------------|--------------|
| | TYPE 1 | TYPE 2 | TYPE 3 |
| Minimum Investment | \$0 | \$3,000 | \$5,000 |
| Minimum Investment Period | No limit | 60 months (5 years) | No limit |
| Withdrawal Limit | No limit | \$2,000 | No limit |
| Withdrawal Penalties | No penalties | \$500 | No penalties |

A TYPE 1 account has a monthly interest rate of 0.2466% (3% annually). For TYPE 2 accounts, the interest rate varies with the age of the account, as shown below.

| Account Age in months | 0 – 12 | 13 – 36 | 37 to 60 | > 60 |
|----------------------------|-----------------|-------------------|-------------------|-----------------|
| Monthly Rate (Yearly Rate) | 0.1162% (2%) | 0.2060% (2.5%) | 0.2871% (3.5%) | 0.4074% (5%) |

The rate for TYPE 3 accounts is a random number between 0% and 0.7974% monthly (10% annually) and changes at the beginning of each month. To generate a random number (double) greater than or equal to zero, but less than 1, use `Math.random()`. For example, the code fragment below generates a random number greater than or equal to 2, but less than 6.

```
double rand = 2 + 4 * Math.random();
```

For all account types, the interest is calculated at month-end on the **total value** of the investment. The interest is added to the total value of the investment.

$$\begin{aligned}\text{Interest} &= \text{Old Total} \times \text{Rate}/100 \\ \text{New Total} &= \text{Old Total} + \text{Interest}\end{aligned}$$

Equivalently, you may use the formula:

$$\text{New Total} = \text{Old Total} \times (1 + \text{Rate}/100)$$

Assignment

Create files to implement three Java classes:

`Investment.java`, `Portfolio.java`, and `ProblemSet2.java`.

1. `Investment.java`

- This class needs 6 data members. Define data members to represent:
 - i. Type of an `Investment`
 - ii. Money invested
 - iii. Total value of the `Investment` (money reinvested plus interest)
 - iv. Rate
 - v. Minimum possible value of an `Investment`
 - vi. Current age of the investment. An investment's age becomes non-zero after money has been invested in it.

What data types should each of these members be?

2. `Portfolio.java`

- This class needs 5 data members. Define data members to represent:
 - i. Three `Investment` objects
 - ii. The lifetime of the `Portfolio` (in months)
 - iii. The desired final value of the `Portfolio`

What data types should each of these members be?

3. `ProblemSet2.java`

- This class contains the `main()` method.
- Exercise 1
 - i. Within the `main()` method, create three `Investment` objects—one of each type - and one `Portfolio`. Each `Investment` contains \$5,600 at the beginning of the first month. The `Portfolio` has a lifetime of 120 months (10 years) and a desired final value of \$25,000. The `Investment` objects must be data members of the `Portfolio`. Make sure to give all data members an initial value.
 - ii. Looping over the lifetime of the `Portfolio`, change the total value of each `Investment` to incorporate the interest at the end of each month. Recall that the interest rate for a TYPE 2 account varies with the age of the account and that for a TYPE 3 account changes randomly each month. Use branching where appropriate.
 - iii. Print the following details at the end of the `Portfolio` lifetime:
 1. The `Portfolio` age
 2. The type, money invested and total value of each `Investment` in the `Portfolio`
 3. Whether the desired `Portfolio` value has been met.
- Exercise 2
 - i. Similar to Exercise 1 part (i), except that
 1. the new TYPE 1 investment has an initial value of \$1,500

2. the other investments have an initial value of \$0
 3. the new portfolio has a lifetime of 240 months (20 years) and a desired final value of \$500,000.
- ii. A total of \$1,320 is to be invested in the Portfolio every month. Develop an investment strategy which adheres to the minimum investment limits for each investment type and has roughly at least a 50-50 chance of meeting the desired Portfolio value of \$500,000.

One possible strategy deposits to the TYPE 1 account until there is enough money available to transfer some to the TYPE 2 account. Further deposits are then split equally between the TYPE 1 and TYPE 2 accounts, until there is enough money in the TYPE 1 account to transfer some to the TYPE 3 account. Once all three accounts have non-zero balances, all further deposits are split equally between them.

You can assume that transfers between types of investments can only be made at the start of each month and that all transfers are free. At the time a transfer is made, you must print:

1. the month during which the transfer was made
2. the amount transferred
3. the account types involved in the transfer
4. the new balances in each account. [When transferring, both the money invested and total value of the `Investment` are affected.]

- iii. Repeat Exercise 1 part (iii) for the second `Portfolio`.

Turn In

- Turn in **electronic** copies of **all source code** (`.java` files). No printed copies are required.
- Place a **comment** with your full name, MIT server username, tutorial section, TA's name, and assignment number at the beginning of all `.java` files in your solution.
- Remember to **comment your code**. Points will be taken off for insufficient comments.
- Place all of the files in your solution into a **single zip file**. Submit this single zip file under the appropriate section and problem set number.
- Your solution is **due at 11AM**. Your uploaded files should have a time stamp of no later than 11AM on the due date.
- **Do not** turn in compiled byte code (`.class` files) or backup source code (`.java~` files).

Penalties

- **30% off** If you turn in your problem set **after 11AM on Friday** but **before 11AM on the following Monday**.
- **No Credit** If you turn in your problem set **after 11AM on the following Monday**.