

BÀI TẬP TUẦN 2

Bài 1. Viết một lớp mô tả hình vuông có tên là `Square`, với các thành viên

- Thuộc tính: Kích thước cạnh có kiểu `float`
- Viết các phương thức trả về kiểu `void` để:
 - Hiển thị độ dài cạnh
 - Tính và hiển thị chu vi
 - Tính và hiển thị diện tích
 - Hiển thị tất cả các thông tin cùng lúc

Viết một **lớp khác** có phương thức `main()` để tạo một đối tượng thuộc lớp `Square` và sử dụng tất cả các phương thức.

Lưu ý: các lớp đặt trong cùng gói có tên *samsung.java.week2.square*

Bài 2. Xây dựng chương trình đơn giản để quản lý thông tin môn học, trong đó sử dụng lớp `Subject` trong bài giảng. Các môn học được lưu trữ vào trong một mảng có kích thước tối đa là 50 phần tử. Mỗi khi chương trình bắt đầu, hiển thị menu sau:

```
Subject Management System
-----
1. Append new subject
2. Update subject
3. Display the information of subject
4. Enrol new student
5. Unenrol student
Your choice (1-4, other to quit):
```

Các chức năng 2-5 khi thực hiện đều yêu cầu người dùng nhập vào mã môn học.

Khi thực hiện chức năng 1-2, nếu mã môn học mới hoặc sửa mã môn học trùng với mã của môn khác thì từ chối.

Sau khi thực hiện xong một chức năng nào đó, sử dụng đoạn mã sau để yêu cầu người dùng nhấn phím Enter để quay về menu của chương trình.

```
System.out.println("Press Enter to continue...");
try {
    System.in.read();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
pressKey.nextLine();
```

Trong đó `pressKey` là một đối tượng thuộc lớp `Scanner`

Lưu ý: các lớp đặt trong cùng gói có tên *samsung.java.week2.subject*

Bài 3. Viết một lớp có tên là `SavingAccount` để mô hình hóa một tài khoản tiết kiệm trong ngân hàng, với yêu cầu:

- Lưu thông tin gồm số tài khoản, tên tài khoản, số tiền gốc trong tài khoản, ngày gửi (sử dụng kiểu `Date`), lãi suất tính theo ngày(%).
- Phương thức để tính tiền lãi nhận được như sau:

*Tiền lãi = Tiền gốc * Lãi suất * Số ngày tính từ ngày gửi đến ngày hiện tại*

- Phương thức rút tiền lãi: Hiển thị số tiền lãi nhận được và chuyển ngày gửi thành ngày hiện tại của hệ thống
- Phương thức rút tiền lãi và một phần tiền gốc: Cho phép nhập vào số tiền cần rút từ gốc. Hiển thị số tiền rút được (bao gồm lượng tiền gốc muốn rút và số lãi nhận được). Chuyển ngày gửi thành ngày hiện tại của hệ thống. *Lưu ý: Số tiền cần rút không được vượt quá số tiền gốc*
- Phương thức gửi thêm tiền vào tiền gốc. Hiển thị số tiền lãi nhận được. Cho phép nhập vào số tiền cần gửi thêm vào số tiền gốc. Chuyển ngày gửi thành ngày hiện tại của hệ thống.
- Phương thức tất toán: Hiển thị số tiền nhận được (bao gồm cả gốc và lãi). Chuyển ngày gửi thành ngày hiện tại của hệ thống, số tiền gốc là 0.

Các yêu cầu:

- Số tài khoản và tên không được rỗng. *Gợi ý: Hãy viết phương thức khởi tạo phù hợp để đạt được điều này*
- Khi rút tiền, số tiền nhập vào dương, nếu không đưa thông báo ra màn hình.

Xây dựng chương trình đơn giản để quản lý tài khoản tiết kiệm, trong đó sử dụng lớp `SavingAccount` trong bài giảng. Các tài khoản được lưu trữ vào trong một mảng có kích thước tối đa là 50 phần tử. Mỗi khi chương trình bắt đầu, hiển thị menu sau:

```
Bank Management System
-----
1. Append new account
2. Withdraw interest
3. Partially withdraw
4. Totally withdraw
5. Deposit
6. Display information
Your choice (1-4, other to quit):
```

- Chức năng 1: Thêm thông tin 1 tài khoản. Nếu số tài khoản mới đã có thì thông báo từ chối
- Chức năng 2: Rút tiền lãi
- Chức năng 3: Rút gốc 1 phần
- Chức năng 4: Tất toán
- Chức năng 5: Gửi tiền
- Chức năng 6: Xem thông tin

Các chức năng 2-6 khi thực hiện đều yêu cầu người dùng nhập vào số tài khoản.

Lưu ý: các lớp đặt trong cùng gói có tên *samsung.java.week2.bank*

Gợi ý: Đoạn mã sau chuyển một chuỗi có định dạng dd/MM/yyyy về đối tượng Date

```
String expectedPattern = "dd/MM/yyyy";
SimpleDateFormat formatter = new SimpleDateFormat(expectedPattern);
try
{
    // (2) give the formatter a String that matches the
    //SimpleDateFormat pattern
    String userInput = "22/09/2009";
    Date date = formatter.parse(userInput);
} catch (ParseException e) {
    // execution will come here if the String that is given
    // does not match the expected format.
    e.printStackTrace();
}
```