Vanier College, Continuing Education
Programming in Java
Winter 2015, Assignment-6

Teacher: Shamima Mithun **Due Date: April 07, 2015**

Objectives

- Class Design

Question 1:

Write a Java class named `Employee` that has the following fields:

- `name` - The `name` field is String type that holds the employee's name.
- `idNumber` - The `idNumber` is an `int` variable that holds the employee's ID number
- `department` – The `department` field is String type that holds the name of the department where the employee works.
- `position` – The `position` field is String type that holds the employee's job title.

The class should have the following constructors:

- A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's `name`, `idNumber`, `department`, and `position`.
- A constructor that accepts the following values as arguments and assigns them to the appropriate fields: employee's `name` and `idNumber`. The `department` and `position` fields should be assigned an empty string (" ").
- A no-arg constructor that assigns empty strings (" ") to the `name`, `department`, and `position` fields, and 0 to the `idNumber` field.

Write appropriate mutator methods that store values in these fields and accessor methods that return the values in these fields. Once you have written the class, write a separate program that creates three Employee objects to hold the following data:

name	idNumber	department	position
Susan Meyers	47899	Accounting	Vice President
Mark Jones	39119	IT	Programmer
Joy Rogers	81774	Manufacturing	Engineer

The program should store this date in three objects and then display the data for each employee on the screen.

Sample Output:

```
Employee #1
Name: Susan Meyers
ID Number: 47899
Department: Accounting
Position: Vice President

Employee #2
Name: Mark Jones
ID Number: 39119
Department: IT
Position: Programmer

Employee #3
Name: Joy Rogers
ID Number: 81774
Department: Manufacturing
Position: Engineer
```

Question 2:

Write a Java class named Car that has the following fields:

- `yearModel` - The `yearModel` field is an `int` that holds the car's year model.
- `make` - The `make` field is `String` type that holds the make of the car.
- `speed` - The `speed` field is an `int` that holds the car's current speed.

In addition, the class should have the following constructor and other methods.

- Constructor. The constructor should accept the car's `yearModel` and `make` as arguments. These values should be assigned to the object's `yearModel` and `make` fields. The constructor should also assign 0 to the `speed` field.
- Accessors. Appropriate accessor methods should get the values stored in an object's `yearModel`, `make`, and `speed` fields.
- `accelerate`. The `accelerate` method should add 5 to the speed field each time it is called.
- `carBreak`. The `carBreak` method should subtract 5 from the speed field each time it is called..

Demonstrate the class in a program that creates a `Car` object, and then calls the `accelerate` method 5 times. After each call to the `accelerate` method, get the current speed of the car and display it. Then call the `carBreak` method 5 times. After each call to the `break` method, get the current speed of the car and display it.

Sample Output:

```
Current status of the car:
Year model: 2004
Make: Porsche
Speed: 0

Accelerating...
Now the speed is 25

Braking...
Now the speed is 0
```

Question 3:

Design a class that holds the following personal information: `name`, `address, age`, and `phone number`. Write appropriate accessor and mutator methods. Demonstrate the class by writing a program that creates three instances of it. One instance should hold your information, and the other two should hold your friends' or family members' information.

Sample Output:

```
My information:
Name: Joe Mahoney
Age: 27
Address: 724 22nd Street
Phone: (555)555-1234

Friend #1's information:
Name: Geri Rose
Age: 24
Address: 149 East Bay Street
Phone: (555)555-5678

Friend #2's information:
Name: John Carbonni
Age: 28
Address: 22 King Street
Phone: (555)555-0123
```

Question 4:

Design a Payroll class that has fields for an `employee's name, ID number, hourly pay rate`, and `number of hours worked`. Write the appropriate accessor and mutator methods and a constructor that accepts the `employee's name, ID number` as arguments. The class should also have a method that returns the employee's gross pay, which is calculated as the number of hours worked multiplied by the hourly pay rate. Write a program that demonstrates the class by creating a Payroll object, then asking the user to enter the data for an employee. The program should display the amount of gross pay earned.

Sample runs of your program should generate the following outputs (user input is shown in blue text):

<u>Sample Output:</u>

```
Enter the employee's name: John
Enter the employee's ID number: 2341
Enter the employee's hourly pay rate: 35
Enter the number of hours worked by the employee: 26

Employee Payroll Data
Name: John
ID Number: 2341
Hourly pay rate: 35.0
Hours worked: 26.0
Gross pay: $910.0
```

Question 5:

Write a Circle class that has the following fields:

- `radius` - a `double`
- `PI` - a `static final double` initialized with the value 3.14159

- The class should have the following methods:
- `Constructor` - Accepts the `radius` of the circle as an argument
- `Constructor` - A no-arg constructor that sets the `radius` field to 0.0
- `setRadius` - A mutator method for the `radius` field
- `getRadius` - An accessor method for the `radius` field
- `getArea` - Return the `area` of the circle, which is calculated as

  ```
  area = PI *radius * radius
  ```
- `getDiameter` - returns the `diameter` of the circle, which is calculated as

  ```
  diameter = radius *2
  ```

- `getCircumference` - returns the `circumference` of the circle, which is calculated as

```
    circumference = 2 * PI * radius
```

Write a program that demonstrates the `Circle` class by asking the user for the circle's `radius`, creating a `Circle` object, and then reporting the circle's `area`, `diameter`, and `circumference`.

Sample runs of your program should generate the following outputs (user input is shown in blue text):

Sample Output:

```
Enter the radius of a circle: 5.4
The circle's area is 91.61
The circle's diameter is 10.8
The circle's circumference is 33.3
```

--End--