# SE 2AA4 Winter 2007

# Software Design Exercise 4

Instructor: William M. Farmer

| | |
|---|---|
| Assigned: | 2 March 2007 |
| Files due: | 16 March 2007 |
| Lab report due: | 23 March 2007 |
| Revised: | 14 March 2007 |

The purpose of this software design exercise is to write a Java program that creates, uses, and tests an abstract data type (ADT) of vectors stored as a sorted linked list. The program will consist of a package named `exercise4` that contains the following four files:

1. `Vector.java`
2. `VectorPlus.java`
3. `TestVectorPlus.java`
4. `ZeroVectorException.java`

Each file will contain a module represented as a Java class.

## Background

A *vector* is a mathematical entity that has direction and magnitude. A two-dimensional vector is represented by a pair $V = (a, b)$ of real numbers where $a$ and $b$ are the x- and y-coordinates of $V$. Let $r$ be a real number and $V_1 = (a_1, b_1)$ and $V_2 = (a_2, b_2)$ be vectors. The *scalar multiple* of $r$ and $V_1$ is the vector $rV_1 = (ra_1, rb_1)$. The *sum* of $V_1$ and $V_2$ is the vector $V_1 + V_2 = (a_1 + a_2, b_1 + b_2)$. And the *dot product* of $V_1$ and $V_2$ is the real number $V_1 \bullet V_2 = a_1 \cdot a_2 + b_1 \cdot b_2$.

## Step 1

Write a first module that is a class named `Vector`. It should:

1. Define an object of type `Vector` that stores a 2-dimensional vector whose components are represented by values of type `float`.
2. Define four class methods for constructing objects of type `Vector`.

The `Vector` class should contain no public fields and only the following public methods:

- Two selectors

  ```
  public float getX();
  ```

  and

  ```
  public float getY();
  ```

  that return the x and y coordinates of an object of type `Vector`, respectively.

- Two constructors

  ```
  public static Vector iVector();
  ```

  and

  ```
  public static Vector jVector();
  ```

  that return the units vectors in the x and y directions, respectively.

- A constructor

  ```
  public static Vector mul(float r, Vector v);
  ```

  that returns the scalar multiple of `r` and `v`.

- A constructor

  ```
  public static Vector add(Vector u, Vector v);
  ```

  that returns the sum of `u` and `v`.

The implementation of your module is required to:

1. Store all constructed objects of type `Vector` in a linked list. Create your own linked list data structure. In particular, do not use the Java class `java.util.LinkedList`.

2. The objects in the linked list should be sorted lexicographically. This means, if $V_1$ and $V_2$ represent vectors $(a_1, b_1)$ and $(a_2, b_2)$, respectively, then $V_1$ comes before $V_2$ in the linked list iff $a_1 < a_2$ or ($a_1 = a_2$ and $b_1 < b_2$).

3. For all $a, b$ of type `float`, store in the linked list at most one `Vector` object that represents the vector $(a, b)$. This means that, if a method is required to return a `Vector` object, representing a vector $(a, b)$, that has not already been created, the procedure should create the `Vector` object and insert it into linked list at the correct place. Otherwise, the `Vector` object should be found in the linked list.

4. Nothing in your implementation should limit the number of `Vector` objects that can be stored in the linked list.

## Step 2

Write an exception named `ZeroVectorException` that will be thrown by the `getAngle` method described below.

## Step 3

Write a second module that is a subclass of `Vector` named `VectorPlus`. It should contain no public fields and only the following public methods:

- A selector

  ```
  public static float getMagnitude(Vector v);
  ```

  that returns the length of `v`.

- A selector

  ```
  public static float getAngle(Vector v);
  ```

  that returns the angle $\alpha$ from the x-axis to `v` where $\alpha$ is measured in radians such that $0 \leq \alpha < 2\pi$. `getAngle` must throw a `ZeroVectorException` when it is applied to the zero vector.

- A constructor

  ```
  public static Vector zeroVector();
  ```

  that returns the `Vector` object that represents the zero vector.

- A method

  ```
  public static float dot(Vector u, Vector v);
  ```

  that returns the dot product of `u` and `v`.

## Step 4

Write a third module that is a class named `TestVectorPlus`. It should "black box" test the services provided by the second module. (Note that the services of the second module includes the services of the first module.) Record the results of using this module to test the second module.

## Step 5

Submit your package of four files using subversion. E-mail your `Vector.java` file to your assigned partner (your partner will be posted on the course web site on Thursday, March 15, 2007). Your partner will likewise e-mail his or her `Vector.java` file to you. You must finish this step no later than 23:59 on Friday, March 16, 2007.

## Step 6

After you have received your partner's `Vector.java` file, replace your `Vector.java` file with your partner's. Do not make any modifications to any of the code. Run your test module and record the results.

## Step 7

Write a report that includes the following:

1. Your name and MAC ID.
2. The four files of your package.
3. Your partner's `Vector.java` file.
4. The results of the test of your second module.
5. The results of the test of your second module using your partner's first module.
6. A discussion of the test results and what you learned doing the exercise. List any problems you found with (1) your program, (2) your partner's `Vector.java` file, and (3) the specification of the three modules. Discuss how these problems could have been avoided.
7. A copy of the part of your log book relevant to this software design exercise.

The lab report is due no later than the beginning of the tutorial session on Friday, March 23.

**Notes**:

1. Please put your name and MAC ID at the top of each of your source files.
2. Your program must work on birkhoff when compiled by `javac`.
3. All fields and methods in your classes should be designated as either `private` or `public` except for the constructor for the `Vector` class which should have no designation.
4. If your partner fails to provide you with his or her `Vector.java` file by the deadline, please tell the instructor via e-mail as soon as possible.