

Department of Computer Engineering

University of Peradeniya

Lab 02

CO225 - Software Construction

October 26, 2018

1 Objectives

The main objective of this laboratory class is to introduce you to the idea of Classes and Objects in Java; their variables and how they can be used to model problems.

2 Introduction

In this laboratory class, you are expected to create a class called Ball to model the behavior of a ball on a 2-D plane. Each ball has a position (x,y) and a velocity (speed and a direction) at a given time and moves on a 2-D plane without friction (that is to say the velocity does not change, but the position does). To simplify the mathematics we will assume balls are of negligible radius.

In your system you can have a number of balls and there is a global clock. These balls will be traveling at some velocity. New balls can be introduced to the system after sometime. These new balls can have an arbitrary starting point and a velocity. For example at $t=0$ the system can have two balls at positions (x_1, y_1) and (x_2, y_2) traveling at v_1 and v_2 respectively. Then when $t=10$ we can introduce a 3rd ball with initial position (x_3, y_3) with a velocity v_3 .

3 What to turn in ?

Task

You should implement a Java Class called Ball to capture this behavior. The class should provide the following functions (you should think of right keywords like static, boolean etc.. for each function) . You may use the provided function signatures for your implementation; but will have to modify the return types, access modifiers etc.

Ball(double x, double y, double speed, double angleOfSpeedWithX) : this is used to add a new ball to the simulation. Note that this is the constructor. A ball can be added sometime after starting the simulation. So the system should keep track of the simulation time. Note that velocity is specified by specifying the speed and the angle it makes with the positive direction of x axis of the 2-D plane.

updateTime(double time): Run the simulation for the time specified. For now this would mean increase the system clock by that amount of time. You should use the correct return type and access modifier for this function (static, public,...).

`willCollide(Ball b1)`: This method can be called on a ball and will return whether or not they would collide with the given ball b1. You should not be able to call this method without an object (only `b2.willCollide(b1)`). Note here that two balls might be introduced into the system at two different times; so your function should consider that.

Your `main` method should be able to do the following

- Create Ball object b1 with (x:0.0, y:1.0, speed:10.0, angleOfSpeed:45.0)
- Update global time by +5.0
- Create Ball object b2 with (x:0.0, y:1.0, speed:20.0, angleOfSpeed:45.0)
- Update global time by +5.0
- Call `willCollide()` to find whether b1 and b2 will collide. Print "B1 and B2 will collide" or "B1 and B2 won't collide" depending on the result.
- Create Ball object b3 with (x:-10.0, y:5.0, speed:3.0, angleOfSpeed:30.0)
- Update global time by +20.0
- Call `willCollide()` to find whether b2 and b3 will collide. Print "B2 and B3 will collide" or "B2 and B3 won't collide" depending on the result.

4 Submission

Submit your Ball.java file to FEeLS before the deadline. We will test for correct use of Java keywords (Public, Static, return types etc.) and good programming practices. You will need some mathematics from your A/L as well.

Rename your java file as `E15xxxBall.java` (where xxx is your registration number).

Not following the instructions would give you **zero** marks.

5 Important

Make sure that you understand about java Classes and Objects concepts with variable types 100% during the lab. If you do not understand any concepts, make sure you get help from your friends or instructors.

6 Deadline

The deadline for the submission is Thursday (2nd of November) 23:55h.