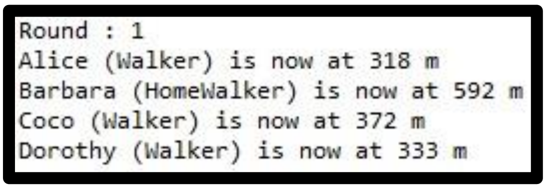
Exercise 2: Generics, Inheritance, Exception

1. Problem Statement: Walk Racing

Barbara wants to organize an event. She invites Alice, Coco and Dorothy to join her event. She wants everyone to play walk racing. Who can get to 10000 meters first is the winner. Barbara will be a home walker. All other participates will be walkers. A home walker will get an additional distance twice per round while other walkers get only once per round. Write a program that calculate distances of each walker from details of walker given (name and speed) and show a winner name when there is a winner. Note that there will be only 1 winner per match.

The outputs of a program:

- If no winner yet, show details of each round including number of round(s), walker’s name and distance of each walker. (Figure 1)



```
Round : 1
Alice (Walker) is now at 318 m
Barbara (HomeWalker) is now at 592 m
Coco (Walker) is now at 372 m
Dorothy (Walker) is now at 333 m
```

Figure 1. Details of each round

2. Implement Details

2.1 Package logic

2.1.1 Class Walker

2.1.1.1 Field

- **protected String name** ; Name of a walker
- **protected int speed** ; A fix distance of walker move per round
- **protected int position** ; A latest position of walker.

2.1.1.2 Constructor

- **Walker(String name, int speed)** ; /* Fill Code */ Initialize name, speed and first position of a walker. (Including HomeWalker)

2.1.1.3 Method

- **public void move()** ; /* Fill Code */ Increase position by a speed and random number from [0,200) meters. (Class RandomGenerator is provided)
- **public String getName()** ; Return name of walker
- **public int getSpeed()** ; Return speed of walker
- **public int getPosition()** ; Return current position of walker

2.1.2 Class HomeWalker extends Walker

2.1.2.1 Constructor

- **HomeWalker(String name, int speed)** ; Initialize name, speed and first position of a Homewalker.

2.1.2.2 Method

- **public void move()** ; /* Fill Code */ Move this walker and also get an additional distance as a home walker.
- **private void homeBuff()** ; /* Fill Code*/ A random additional distance of home walker from [0, 200) meters. (Class RandomGenerator is provided)

2.1.3 Class RaceManager

2.1.3.1 Field

- **private List<Walker> walkers** ; List of walker in the match
- **private final int finishPosition** ; A distance of a winner
- **private int round** ; A round number

2.1.3.2 Constructor

- **RaceManager(List<Walker> walkers)** ; Initialize walkers and round number.

2.1.3.3 Method

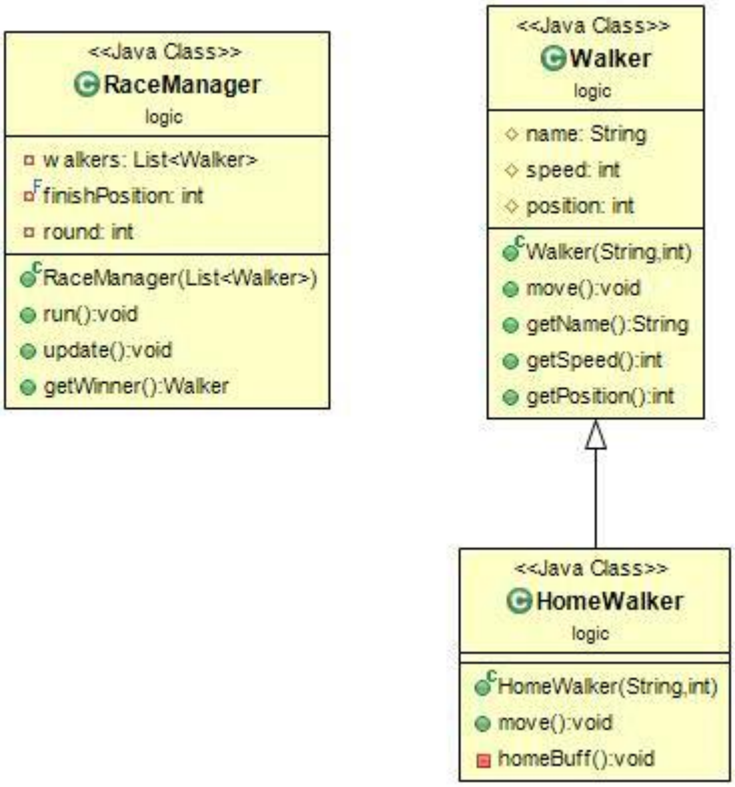
- **public void run()** ; Start and loop the round until there is a winner.
- **public void update()** ; /* Fill Code */ Move each walker.
- **public Walker getWinner()** ; /* Fill Code */ Return the winner of this race. The winner is a walker whose position is greater than or equal to finishPosition. If there is no winner yet, this method return null. (If there is more than 1 walkers cross the finish line, the winner is HomeWalker. If there isn't HomeWalker in the finish line, the first walker in the list wins)

2.2 Package util

2.2.1 Class RandomGenerator (provided)

2.2.1.1 Method

- **public static int random(int from, int to)** ; Return integer within range [from, to)



Class diagram of Walk Racing