60/60 = (100/100)

CS 1401, Exam #3, MW version

Date: Wednesday, November 19, 2014
Name (please type legibly, ideally in block letters): [redacted]

Exactly 45 years ago, on November 19, 1969, the legendary Brazilian football (soccer) player Pele scored his 1,000-th goal.

10/10

1a. Write a method named *average* that, given the total number of goals scored by the player and the starting and ending years, returns the average number of goals per year scored by the player.

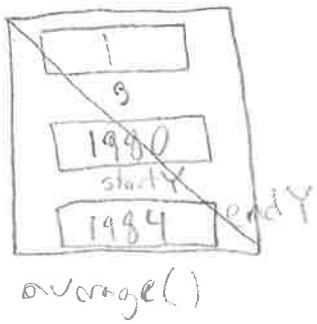
1b. In the *main* method, call your method *average* to compute the average number of goals scored by Vladik Kreinovich, who played (when at school) for 4 years and scored 1 goal.

1c. Trace, step-by-step, how the computer will perform all the needed computations, and check that the result is indeed correct.

1a.
```
public static double average(int g, int startY, int endY) {
    return 1.0 * g / (endY - startY);
}
```

b.
```
main
    double averageVK = average(1, 1980, 1984);
```

1c.



2a. Define a class *Footballer* whose objects are different football players. The description of each player should contain his/her name, country, number of years played, and number of goals scored. Your class should contain a constructor method, get- and set-methods, and a method for computing the average number of goals scored per year.

2b. Use your class in the *main* method to define a new object *pele* of type *Footballer* describing Pele (Brazil) at the end of 1969, when he scored 1,008 goals during the 14 years of his career (he started in 1955). Compute and print the average number of goals that Pele scored per year. Then, update the values based on the year 1970, during which he has scored 72 more goals, and compute and print the new average. 15 1080

2c. Trace your program step-by-step.

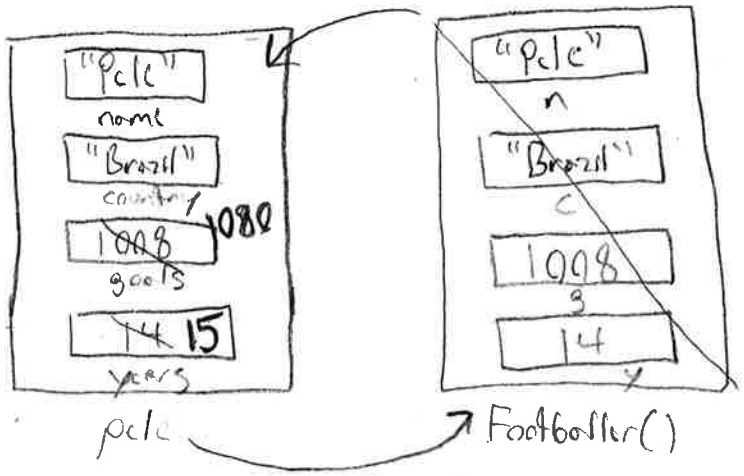
2a. 10/10

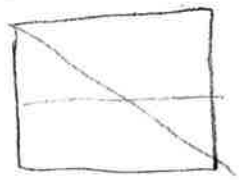
```
public class Footballer {
    private String name;
    private String country;
    private int years;
    private int goals;
    public Footballer(String n, String c, int y, int g){
        name = n;
        country = c;
        years = y;
        goals = g;
    }
    public String getName(){ return name; }
    public void setName(String n){ name = n; }
    public String getCountry(){ return country; }
    public void setCountry(String c){ country = c; }
    public int getYears(){ return years; }
    public void setYears(int y){ years = y; }
    public int getGoals(){ return goals; }
    public void setGoals(int g){ goals = g; }
    public double average(){ return 1.0 * goals/years; }
}
```

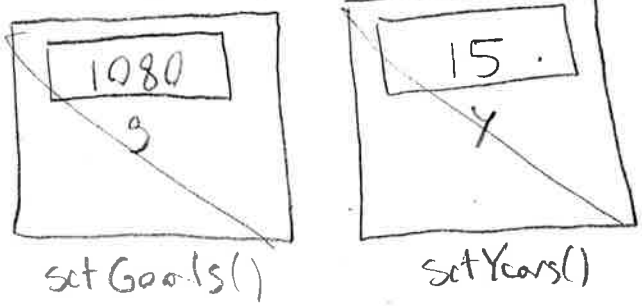
26. main

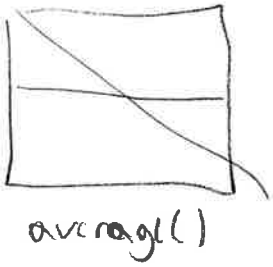
```
Footballer pele = new Footballer("Pele","Brazil", 1008, 14);
System.out.println("Pele's goal/year average after 1969 is" + average());
pele.setGoals(1080);
pele.setYears(15);
System.out.println("Pele's goal/year average after 1970 is" + average());
```

2c.









10/10

3a. Which football player scored the most goals? To answer this question, write a method that, given an array *g* of goals scored by different players and an array *n* of their names, returns the name of the highest-scoring player.

3b. To check the correctness of your method, test it on the example of two arrays: array *players* with elements Pele and Kreinovich, and array *goals* with values 1,008 and 1.

3c. Trace step-by-step how the piece of code you wrote in Part 3b finds the name of the player who scored the most goals.

3a.

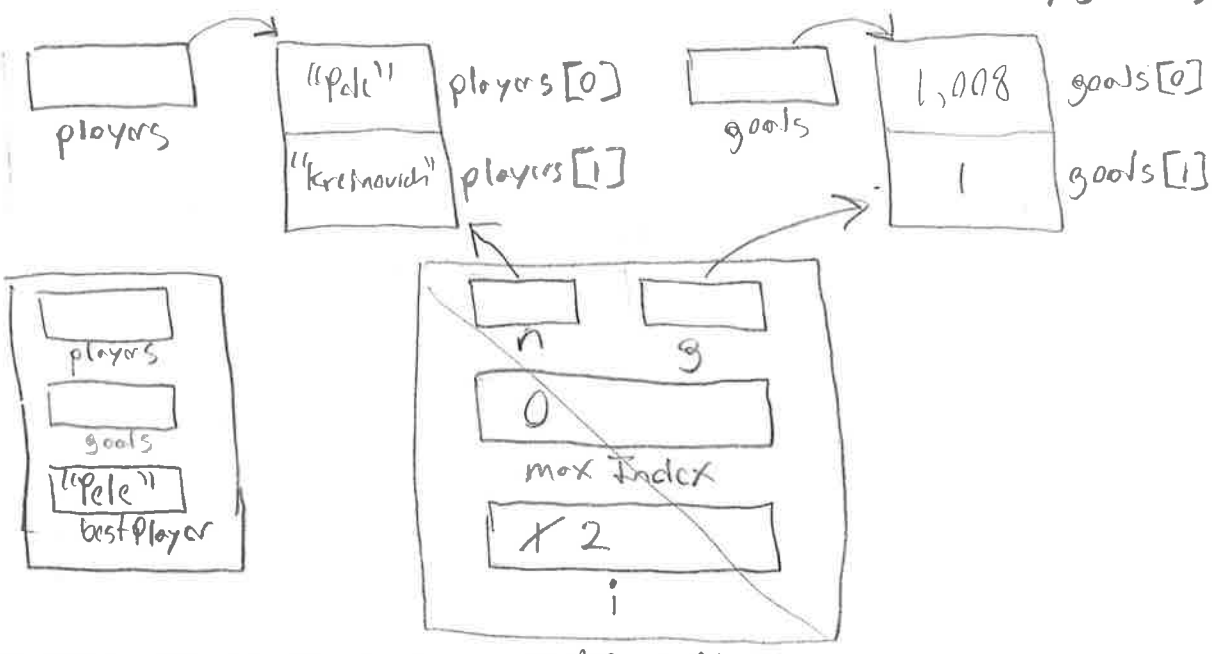
```
public static String mostGoals(int[] g, String[] n){
    int maxIndex = 0;
    for(int i = 1; i < g.length; i++){
        if(g[i] > g[maxIndex])
            maxIndex = i;
    }
    return n[maxIndex];
}
```

3b.

mom

```
String[] players = {"Pele", "Kreinovich"};
int[] goals = {1,008, 1};
String bestPlayer = mostGoals(players, goals);
```

3c.



10/10

4a. Suppose that a football player gets a certain monetary bonus for each goal. Write a method that, given an array of goals scored by different players and a bonus-per-goal value, returns a new array with the amounts given to each player.

4b. To check the correctness of your method, test it in the main program on the example of an array *g* consisting of 1,008 and 1 goals, and a bonus of $100 per goal.

4c. Trace step-by-step what happens when you run the piece of code you wrote in Part 4b. The result should be an array consisting of values $100,800 and $100.

4a.

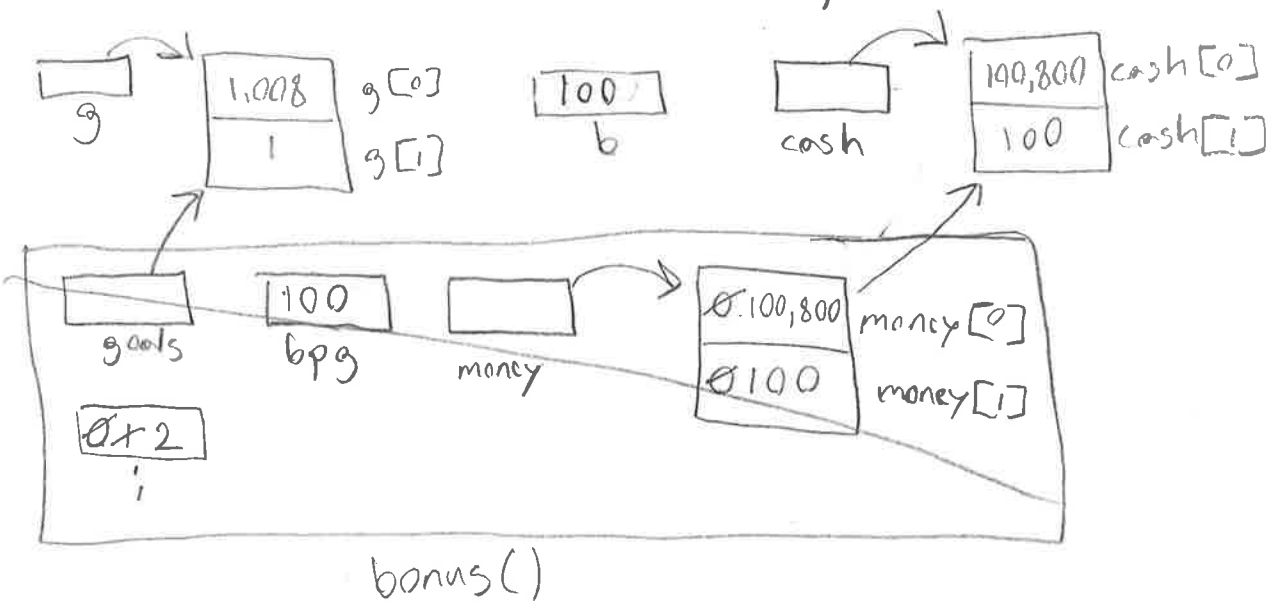
```
public static int[] bonus(int[] goals, int bpg){
    int[] money = new int[goals.length];
    for(int i=0; i<goals.length; i++){
        money[i] = goals[i] * bpg;
    }
    return money;
}
```

4b.

main

```
int[] g = {1,008, 1};
int b = 100;
int[] cash = bonus(g, b);
```

4c.



10/10

5. Describe the main teamwork roles, and what are the main tasks of the corresponding persons.

* time keeper
  - keeps track of available time
* record keeper
  - sole writer of group
  - records discussion between group members
* leader
  - instigates discussion among group members
  - makes sure everybody has a say
  - makes sure group stays on topic
  - makes sure nobody dominates entire discussion, everybody gives input

10/10

6. Describe what is white-box testing and what is black-box testing. Describe the main rules for testing; give an example for each rule.

White-box testing is a testing method in which the tester is fully aware of the internal structure of the source code.

Black-box testing is an alternative where the tester does not know contents of the source code and only knows the input and output.

The main rules for testing are:

- Test all branches
- Test simple values
- Test the boundaries or limits
- Test random values

branches → Test "for" loop limits

simples values → 1, 2, 3, 4. ...

boundaries → $\frac{1}{0}$ (infinity) or 0

random values → 234.658 (some value that is not simple)