

Exam TDA547, DAT170

Krasimir Angelov, tel. 073-8228-179

2014-03-15

The total number of points is 40. 20 points certainly guarantee a pass. 27p correspond to grade 4 and 32p to grade 5.

No other help materials except an English Dictionary are allowed. Write clean and readable Java code. Trivial syntax errors will be tolerated without affecting the grades. You don't have to comment your code unless if you really want to.

1. Read the following program:

```
public class Question1 {
    public static int delta(int[] a) {
        int min = Integer.MIN_VALUE;
        int max = Integer.MAX_VALUE;
        for (int i = 0; i < a.length-1; i ++) {
            if (a[i] < min)
                min = a[i];
            else if (a[i] > max)
                max = a[i];
        }
        return (max-min);
    }

    public static void main(String[] args) {
        int[] a = {3, 5, -8, 10};
        System.out.println(delta(a));
    }
}
```

What will the program print when it is executed? (4p)

2. In this task we do some simple array processing:

- A harmonic array is an array in which each element is equal to the average of the values of its neighbours. The elements on the border, i.e. the first and the last element are ignored. For example, the array:

2.1 3.3 4.5 5.7

is harmonic because $3.3 = (2.1 + 4.5)/2$ and $4.5 = (3.3+5.7)/2$.

Write a method:

```
public static boolean isHarmonic(double[] a)
```

which returns `true` if the array `a` is a harmonic array. (4p)

- Implement a class `Question2` which can be used to test whether an array is harmonic or not. The class should be possible to run like this:

```
> java Question2 2.1 3.3 4.5 5.7
true
```

i.e. it takes the array of numbers from the command line arguments and prints `true` if the array is harmonic and `false` if it is not. (4p)

3. In group work we have to work in teams. Every team must have a leader who assigns different tasks to the members of the team. To make things simple we assume that the leader of each team is the oldest person in the group. If you already have the class:

```
public class Person {
    public String getName();
    public int getAge();
}
```

then implement a class `Team` with the following methods:

- `void addPerson(Person person)` - to add another person in the team
- `Person getLeader()` - to return the leader of the team, i.e. the oldest person in the group. If there are several people of the same age then just return any of them.
- `Person getVolunteer()` - to return a random member of the team. Each call to this method should return a different member.

- `void print(PrintStream out)` - to print the names of the members in the team. The name of the leader must be followed by a star.

Example

```
Eva
Emil
John *
Christina
```

(10p)

4. A prime number is an integer that is divisible only by one and by itself. The Eratosthenes sieve is a classical algorithm for finding all prime numbers smaller than a fixed `n`. Proceed as follows:
 - (a) Start with the sequence of numbers from 2 (the smallest prime number) to `n`.
 - (b) Look at the first number and strike out all numbers in the rest of the sequence that are multiples of this number.
 - (c) Repeat (b) with the next number in the sequence which is not striked out yet.

At the end the only numbers which are not striked out will be the prime numbers. An example for the first three iterations is this:

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15
2 3 4 5 6 7 8 9 10 11 12 13 14 15
2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Here we start with 2 and strike out all numbers that are even. On the next step the next unstriked number is 3 and then we strike out all numbers divisible by 3.

Hint: represent the sequence of numbers with a boolean array where there is one element for every number in the sequence. If the number is striked out then the element should be `true`, otherwise `false`. Use two nested loops, one which iterates over the whole sequence and another which strikes out the multiples.

Implement:

- (a) a class `Primes` with a constructor which takes as a parameter the upper bound `n`. The constructor must create and initialize the boolean array. (8p for the full solution and 2p if you only define the class and create the array in the constructor)

- (b) a method `boolean isPrime(int i)` which returns true if `i` is a prime number. (2p)
5. Many algorithms for processing text documents rely on statistical information. The simplest example is the frequency with which a single word appears in a given text. Implement a method:

```
HashMap wordCounts(Scanner in)
```

which uses the scanner to read all words in a text and returns a hash map whose keys are the words in the text and the values are integers corresponding to the number of times each word appears in the text. (8p)

Hint: here by word we understand a string returned by the `next()` method for `Scanner`.