Object Oriented Programming TDA547

Krasimir Angelov, tel. 070-234-24-78

2015-04-14

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 void fill(int d, int[] a) {
        for (int i = 0; i < a.length; i++) {
            a[i] = d;
            d += 2;
        }
    }

    public static void main(String[] args) {
        int d = 0;
        int[] a = new int[4];
        fill(d,a);
        for (int i = 0; i < a.length; i++) {
            System.out.println(a[i]);
        }
        System.out.println(d);
    }
}
```

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

2. In this task we do simple array processing:

- An arithmetic progression is a sequence of numbers such that the difference between every two consecutive numbers is a constant. For example 1 3 5 7 is an arithmetic progression because you get the next number by adding two to the previous. Similarly 8 7 6 5 is a progression because we substract one from each number to get the next one. Implement the static method:

  ```
  public static boolean isProgression(int[] a)
  ```

  which returns true if the array *a* contains an arithmetic progression. (4p)

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

  ```
  > java Question2 1 3 5 7
  true
  ```

  i.e. it takes the sequence of numbers from the command line arguments and prints `true` or `false`. (4p)

3. The software on your laptop allows programs to monitor the current battery power. You need a simple component that can be used to simulate this. Assume that there is the following interface:

```
public interface BatteryListener {
    void updateStatus(int power);
}
```

- Write a class called `Battery` with the following definitions:
  - a constructor which creates a battery with a given power level:
    `public Battery(int power)`
  - a method to register a new BatteryListener:
    `public void addListener(BatteryListener l);`
    Note! There might be more than one listeners.
  - a method to charge/discharge the battery:
    `public void charge(int delta);`
    This method increases the current power with `delta`. A positive number can be used to *charge it* and a negative to *discharge it*. Every time when the power changes the method `charge` should call the method `updateStatus` for every listener registered with `addListener`. The argument to `updateStatus` is the current battery power.

(8p)

- Write a test class called `TestBatteryListener` which implements `BatteryListener` and simply prints the current power. (2p)

4. One of the oldest methods for encryption of text messages is to replace every letter in the message with another letter. For example, the following is an encryption table where every letter in the first row should be replaced with the corresponding letter in the second row.

Plaintext alphabet:	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext alphabet:	ZEBRASCDFGHIJKLMNOPQTUVWXY

In this way the word ”exam” can be encoded as ”awzj”. Your task is to write a static method that can generate a new random encryption table:

```
public static char[] generateCipher()
```

The table is an array with 26 characters. Each character represents the encoding of the corresponding letter in the original alphabet. The table should consist of only the letters from `A` to `Z` and each letter should be used exactly once. The order of the letters must be random. (10p)

Hint: You can generate a random letter by using one array with the letters `A`, `B`, `C` ... `Z` from which you pick a random element.

5. Many web sites ask their users to create an account with password. The password should not be very easy and a simple way to check this is to lookup the password in a file with easy passwords. Write a static method:

```
public static boolean isEasy(File dictionary, String password)
          throws FileNotFoundException
```

that returns true if the password is in the file referenced by the argument `dictionary`. The passwords in the file are just words separated by spaces. (8p)

(check the cheat sheet for the methods of Scanner).