Exercises module 17 – Inheritance

17.1 – Extract base class

1. Given these two classes:

```
class Processor
{
    public int id;
    public String productName;
    public boolean isForSale;
    public int frequency;
    public int numberOfCores;
}

class Keyboard
{
    public int id;
    public String productName;
    public boolean isForSale;
    public boolean gotNumericKeypad;
    public String color;
    public boolean isWireless;
}
```

For example, you can call the Processor class using code like:

```
Processor cpu = new Processor();

cpu.id = 1040;
cpu.productName="Intel Core i7";
cpu.isForSale=true;
cpu.frequency=3000;
cpu.numberOfCores=8;
```

2. Extract a suitable base-class named **Product** and let the **Processor** and **Keyboard** classes **inherit** from it

3. Create a new class named **Monitor** that inherits from the **Product** class and adds the following two fields:
    a. int size;
    b. float weight;

4. Create an instance of the Monitor class and you will notice that the fields from both the **Product** and the **Monitor** class is available for you to use.

5. We also need to create two sub-classes of the Monitor class, one called **LCDMonitor** and one called **CRTMonitor** with the following fields:
    a. LCDMonitor

        i. boolean touchEnabled;
        ii. boolean vesaMount;
        iii. boolean hasSpeaker;
    b. **CRTMonitor**
        i. int refreshRate;
        ii. boolean environmentFriendly;

6. Create an instance of the **LCDMonitor** class and explore what fields are available to you to set.

7. To the **LCDMonitor** class add a **print** method that uses some of the fields from both the Product and Monitor class, like:

```
public void print(){
    System.out.printf("ProductID {0}\r\n", id);
    System.out.printf("Size {0}\r\n", size);
    System.out.printf("TouchEnabled {0}\r\n", touchEnabled);
}
```

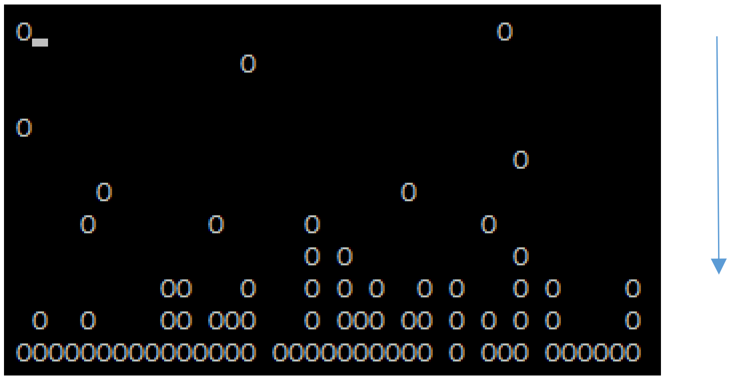
8. What would happen if you made the **id** field **private** or **protected**
    a. Can you access it from the main method?
    b. Can you access it in the print() method?

   Make sure you understand the impact of using private/protected/public and when to use what.

17.2 – Snow! (Advanced)

1. It's soon Christmas time and we need to make a snow application that will simulate snow in the console.

   Like:



```
O                             O
              O

O
                               O
     O                  O
    O       O     O          O
                  O O          O
         OO   O   O O O  O O   O O    O
 O  O    OO OOO   O OOO OO O O O O    O
OOOOOOOOOOOOOOO OOOOOOOOOO O OOO OOOOOO
```

   The snowflakes will randomly fall down the screen.

2. Create a new project and add a reference to **com.googlecode.lanterna** library (V2.1.9).

   Start with a Flake class to represent a single snow flake:

```
public class Flake {
    public int x;
    public int y;
}
```

   Then use:

   - A generic list that contains a list of all the flakes. We add new Flakes to the list (at random X-positions) and then display them on the screen
   - At each press a key, all the snowflakes should drop one step down, so add the necessary logic to implement this. (Change the y position for every flake)
   - Add if you can logic to make sure that a flake can't move if the next position is already occupied by a flake.

Hints

- To wait for a key-press in Lanterna

```
//Wait for a key to be pressed
Key key;
do{
    Thread.sleep(5);
    key =terminal.readInput();
}
while(key == null);
```

- Clearing the screen

```
terminal.clearScreen();
```

- To put a character on the screen at a given position

```
terminal.moveCursor(x,y);
terminal.putCharacter('O');
terminal.moveCursor(0,0);
```

Advanced Stretch task
Implement snow and use your imagination to make it more realistic! Some ideas are:

- Add some structure in the scene that the snow will fall onto and pile up on

- Add logic so that if two flakes are on top of each other, then the top one will randomly fall to the left or right (if it can), like:

Before	After
X X	 XX

Questions and concepts to study further on your own:

- Private vs protected vs public
- What is multiple inheritance?
  https://en.wikipedia.org/wiki/Multiple_inheritance
- Why does Java not support multiple inheritance? And what would the problem with multiple-inheritance?
  https://stackoverflow.com/questions/2515477
- When to use and not use inheritance