Programs

Table Of Contents

[1. Immutable Class 3](#_Toc518559313)

[2. Producer consumer 5](#_Toc518559314)

[3. Print Odd / Even 7](#_Toc518559315)

[4. Custom Exception 8](#_Toc518559316)

[5. Future Topics 9](#_Toc518559317)

[6. Future Subject 9](#_Toc518559318)

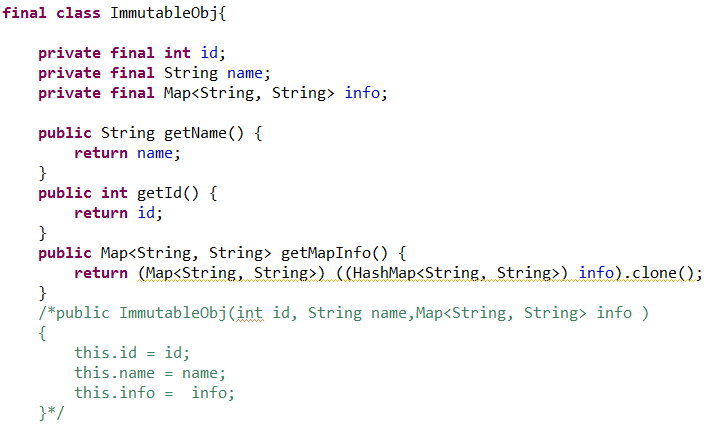
***Programs***

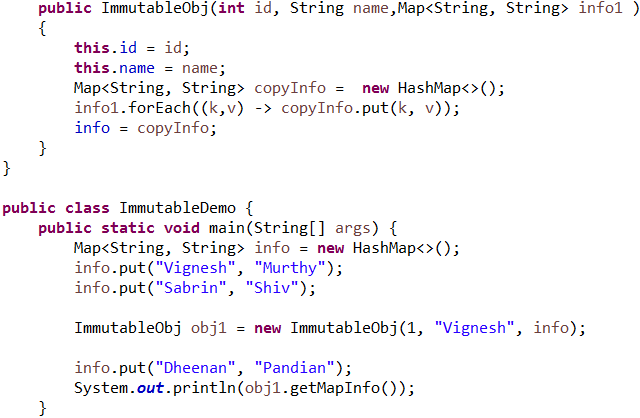
# Immutable Class

The following are the steps involve in the creation of an immutable class.

1. Declare the class as final So that it cannot be extended.
2. Make all the fields as private so that is direct access is not allowed.
3. Do not provide setter method for any of the fields.
4. Make all the mutable fields as final so that it can be initialized only once.
5. Initialize all the fields through a constructor.
6. Performing cloning of objects in getter methods to return the copy of an object rather than returning the actual object.

File: ImmutableDemo.java



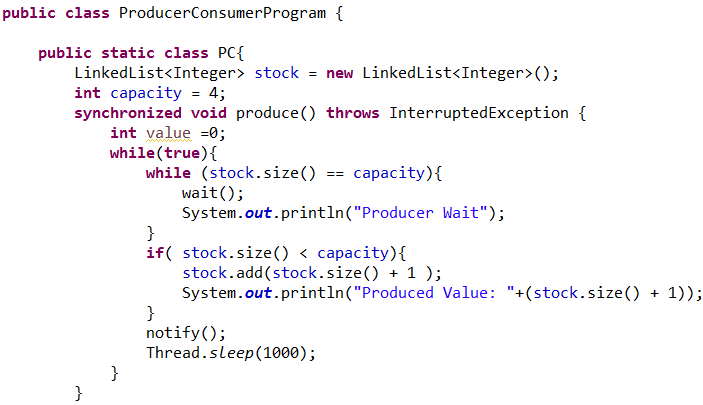


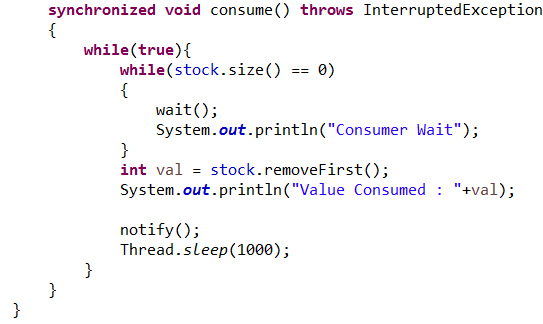
In the above mentioned example the following 2 concepts needs to be noted.

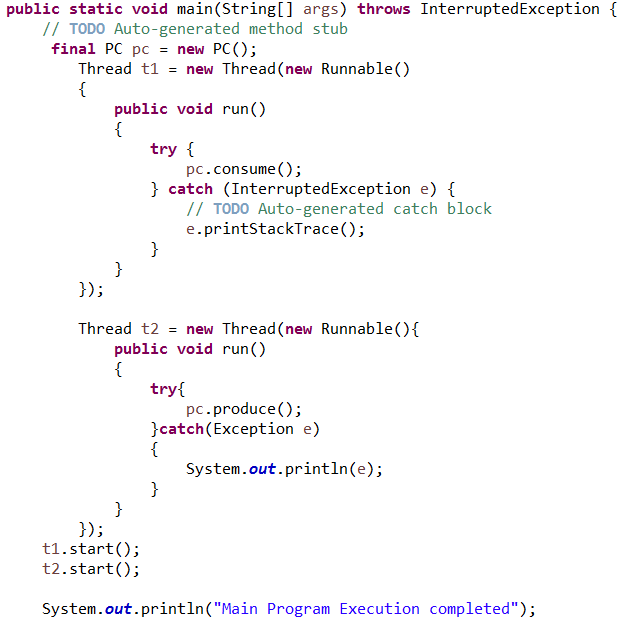
First Concept: There are two versions of the parameterized constructor one to create a shallow copy and the other one to create a deep copy. The reason we do that is if we use a shallow copy by assigning the incoming variable to the existing variable, both the map points to the same memory and any change to the source gets reflected to the immutable object as well.

Second Concept: The getter has to be implemented to return a clone of the collection object rather than returning the actual object to avoid modification.

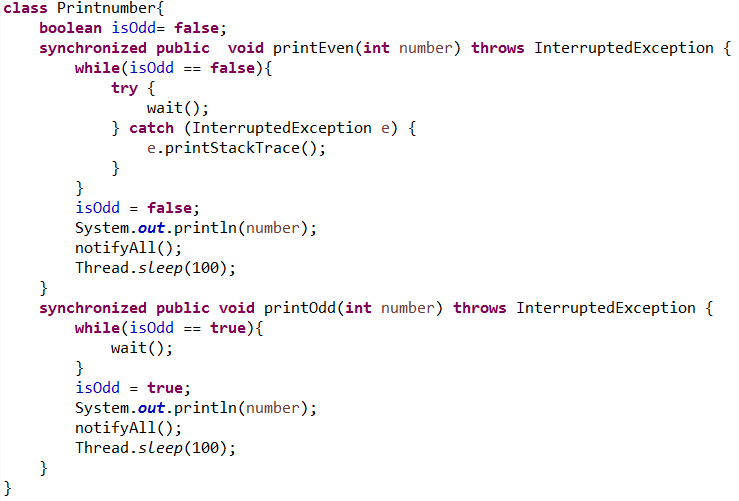
# Producer consumer

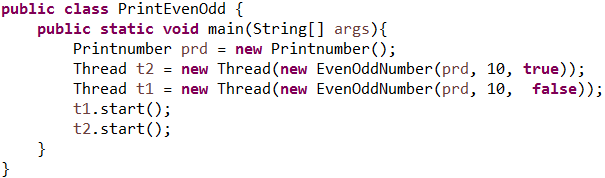




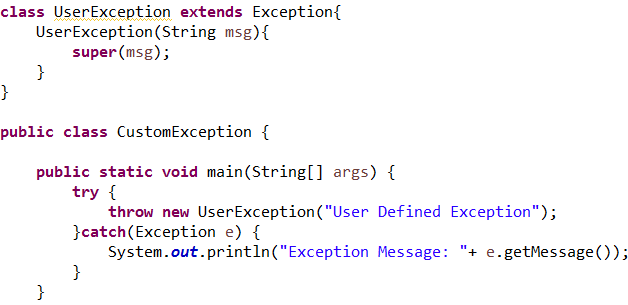


# Print Odd / Even



# Custom Exception



# Future Topics

|  |  |  |
| --- | --- | --- |
| ***SI NO*** | ***Topic*** | ***Status*** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |
|  |  |  |
|  |  |  |

# Future Subject

|  |  |  |
| --- | --- | --- |
| ***SI NO*** | ***Topic*** | ***Status*** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |
|  |  |  |
|  |  |  |