Template Method Pattern

Data Abstraction

Simple and Extensible Reuse of Code

Think about how data abstraction makes the code reuse simple and extensible.

Data Abstraction

- the process of identifying only the required features of an object
- ignore the irrelevant details
- the non-essentials parts are not displayed in the class definition

Template Method Pattern

- The template method is a method in a superclass, usually an **abstract** superclass, and defines the **skeleton** of an operation in terms of a number of high-level steps.
- template method **reused** **WO modification**
- use **abstract** keyword for empty implementation
- The **subclasses provides implementation of the abstract part** of the template class
- The intent of the template method is **to define the overall structure of the operation**, while allowing **subclasses to refine, or redefine certain steps**.

Template Method defines a skeleton of an algorithm in a base class, and let subclasses override the steps without changing the overall algorithm's structure.

```
abstract class Game {
    protected int playersCount;
    abstract void initializeGame();
    abstract void makePlay(int player);
    abstract boolean endOfGame();
    abstract void printWinner();
/* A template method : */
    final void playOneGame(int playersCount) {
        this.playersCount = playersCount;
        initializeGame();
        int j = 0;
        while (!endOfGame()) {
            makePlay(j);
            j = (j + 1) % playersCount;
        }
        printWinner();
    }
}
```

Think about the <u>**logic flow**</u> and sequenced use of template methods.
Look at **fixed parts and changing parts**

```
class Monopoly extends Game {
    /* Implementation of necessary concrete methods */
      void initializeGame() {
      // Initialize money
    }
    void makePlay(int player) {
      // Process one turn of player
    }
    boolean endOfGame() {
 // Return true if game is over according to Monopoly rules
    }
    void printWinner() {
      // Display who won
    }
   /* Specific declarations for the Monopoly game. */
   // ...
     playOneGame(int playersCount)
 }
```

Method implementation to be changed, method names and logic flow of playOneGame Fixed

```
class Chess extends Game {
   /* Implementation of necessary concrete methods */
     void initializeGame() {
     // Put the pieces on the board
    }
    void makePlay(int player) {
      // Process a turn for the player
    }
    boolean endOfGame() {
     // Return true if in Checkmate or Stalemate has been reached
    }
     void printWinner() {
      // Display the winning player
    }
    /* Specific declarations for the chess game. */
     playOneGame(int playersCount)
 }
```

Example https://refactoring.guru/design-patterns/template-method/java/example

In this example, the Template Method pattern defines an algorithm of working with a social network. Subclasses that match a particular social network, implement these steps according to the API provided by the social network.

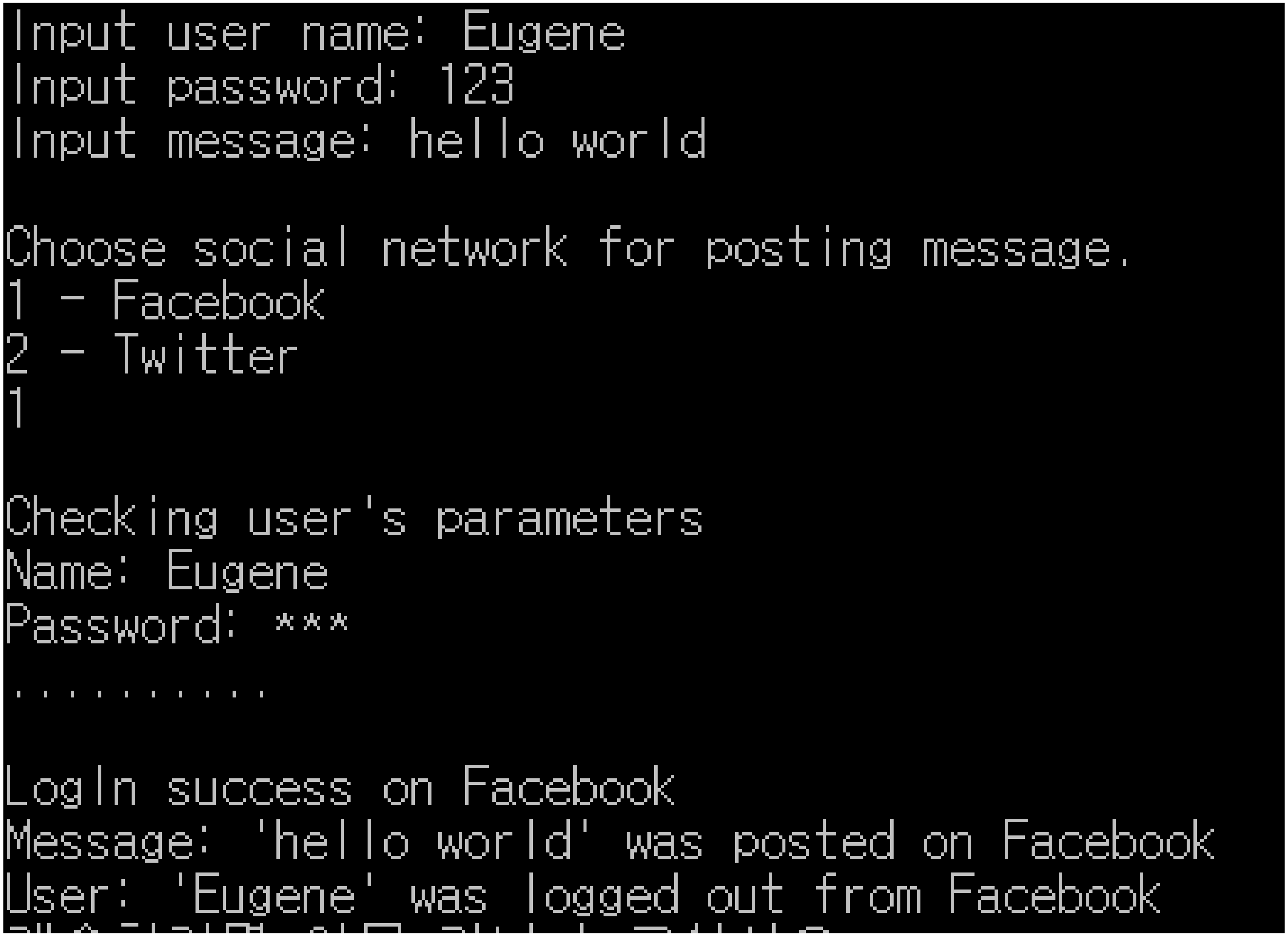
```
public abstract class Network {
    String userName;
    String password;
    Network() {}
   public boolean post(String message) {
        if (logIn(this.userName, this.password)) {
            // Send the post data.
            boolean result =  sendData(message.getBytes());
            logOut();
            return result;
        }
        return false;
    }
    abstract boolean logIn(String userName, String password);
    abstract boolean sendData(byte[] data);
    abstract void logOut();
}
```

```
public class Facebook extends Network {
    public Facebook(String userName, String password) {
        this.userName = userName;
        this.password = password;
    }
 public boolean logIn(String userName, String password) {
        System.out.println("₩nChecking user's parameters");
        System.out.println("Name: " + this.userName);
        System.out.print("Password: ");
…
            simulateNetworkLatency();…
}
 public boolean sendData(byte[] data) {
        boolean messagePosted = true;
        if (messagePosted) {…
}
 public void logOut() {
        System.out.println("User: '" + userName + "' was logged out
from Facebook");
    }
 private void simulateNetworkLatency() {
        try {…
```

```
public class Twitter extends Network {
    public Twitter(String userName, String password) {
        this.userName = userName;
        this.password = password;
    }
 public boolean logIn(String userName, String password) {
        System.out.println("₩nChecking user's parameters");
...
  simulateNetworkLatency();
...
}
 public boolean sendData(byte[] data) {
        boolean messagePosted = true;
   ....
}
 public void logOut() {
        System.out.println("User: '" + userName + "' was logged out from Twitter");
    }
 private void simulateNetworkLatency() {
        try {
 ...
}
```

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Demo {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new
                InputStreamReader(System.in));
        Network network = null;
        System.out.print("Input user name: ");
        String userName = reader.readLine();
        System.out.print("Input password: ");
        String password = reader.readLine();
        System.out.print("Input message: ");
        String message = reader.readLine();
```

```
  System.out.println("₩nChoose social network for posting message.₩n" +
                "1 - Facebook₩n" +
                "2 - Twitter");
        int choice = Integer.parseInt(reader.readLine());
        // Create proper network object and send the message.
        if (choice == 1) {
            network = new Facebook(userName, password);
        } else if (choice == 2) {
            network = new Twitter(userName, password);
        }
        network.post(message);
    }
}
```



```
Input user name: Eugene
Input password: 123
Input message: hello world

Choose social network for posting message.
1 - Facebook
2 - Twitter
1

Checking user's parameters
Name: Eugene
Password: ***
..........

LogIn success on Facebook
Message: 'hello world' was posted on Facebook
User: 'Eugene' was logged out from Facebook
```