



🏠 / 设计模式 / 模板方法模式 / Java



Java 模版方法模式讲解和代码示例

模版方法是一种行为设计模式，它在基类中定义了一个算法的框架，允许子类在不修改结构的情况下重写算法的特定步骤。

[👉 进一步了解模版方法 →](#)

在 Java 中使用模式

复杂度：☆☆☆

流行度：★★☆

使用示例：模版方法模式在 Java 框架中很常见。开发者通常使用它来向框架用户提供通过继承实现的、对标准功能进行扩展的简单方式。

这里是一些核心 Java 程序库中模版方法的示例：

- `java.io.InputStream`、`java.io.OutputStream`、`java.io.Reader` 和 `java.io.Writer` 的所有非抽象方法。
- `java.util.AbstractList`、`java.util.AbstractSet` 和 `java.util.AbstractMap` 的所有非抽象方法。

- `javax.servlet.http.HttpServlet`，所有默认发送 HTTP 405 “方法不允许” 错误响应的 `doXXX()` 方法。你可随时对其进行重写。

识别方法：模版方法可以通过行为方法来识别，该方法已有一个在基类中定义的“默认”行为。

导航

📖 简介

📖 重写算法的标准步骤

📁 networks

📄 Network

📄 Facebook

📄 Twitter

📄 Demo

📄 OutputDemo

重写算法的标准步骤

在本例中，模版方法模式定义了一个可与社交网络协作的算法。与特定社交网络相匹配的子类将根据社交网络所提供的 API 来实现这些步骤。

📁 networks

📄 networks/Network.java: 基础社交网络类

```
package refactoring_guru.template_method.example.networks;

/**
 * Base class of social network.
 */
public abstract class Network {
    String userName;
    String password;

    Network() {}

    /**
     * Publish the data to whatever network.
     */
}
```

```

public boolean post(String message) {
    // Authenticate before posting. Every network uses a different
    // authentication method.
    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();
}

```

networks/Facebook.java: 具体社交网络

```

package refactoring_guru.template_method.example.networks;

/**
 * Class of social network
 */
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: ");
        for (int i = 0; i < this.password.length(); i++) {
            System.out.print("*");
        }
        simulateNetworkLatency();
        System.out.println("\n\nLogIn success on Facebook");
        return true;
    }

    public boolean sendData(byte[] data) {
        boolean messagePosted = true;
        if (messagePosted) {
            System.out.println("Message: '" + new String(data) + "' was posted on Facebook");
            return true;
        } else {
            return false;
        }
    }
}

```

```

public void logOut() {
    System.out.println("User: '" + userName + "' was logged out from Facebook");
}

private void simulateNetworkLatency() {
    try {
        int i = 0;
        System.out.println();
        while (i < 10) {
            System.out.print(".");
            Thread.sleep(500);
            i++;
        }
    } catch (InterruptedException ex) {
        ex.printStackTrace();
    }
}
}

```

networks/Twitter.java: 另一个社交网络

```

package refactoring_guru.template_method.example.networks;

/**
 * Class of social network
 */
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");
        System.out.println("Name: " + this.userName);
        System.out.print("Password: ");
        for (int i = 0; i < this.password.length(); i++) {
            System.out.print("*");
        }
        simulateNetworkLatency();
        System.out.println("\n\nLogIn success on Twitter");
        return true;
    }

    public boolean sendData(byte[] data) {
        boolean messagePosted = true;
        if (messagePosted) {
            System.out.println("Message: '" + new String(data) + "' was posted on Twitter");
        }
    }
}

```

```

        return true;
    } else {
        return false;
    }
}

public void logOut() {
    System.out.println("User: " + userName + " was logged out from Twitter");
}

private void simulateNetworkLatency() {
    try {
        int i = 0;
        System.out.println();
        while (i < 10) {
            System.out.print(".");
            Thread.sleep(500);
            i++;
        }
    } catch (InterruptedException ex) {
        ex.printStackTrace();
    }
}
}

```

Demo.java: 客户端代码

```

package refactoring_guru.template_method.example;

import refactoring_guru.template_method.example.networks.Facebook;
import refactoring_guru.template_method.example.networks.Network;
import refactoring_guru.template_method.example.networks.Twitter;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/**
 * Demo class. Everything comes together here.
 */
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();

        // Enter the message.
        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);
}
```

OutputDemo.txt: 执行结果

```
Input user name: Jhonatan
Input password: qswe
Input message: Hello, World!

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

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

LogIn success on Twitter
Message: 'Hello, World!' was posted on Twitter
User: 'Jhonatan' was logged out from Twitter
```

[继续阅读](#)[Java 访问者模式讲解和代码示例 →](#)[返回](#)[← Java 策略模式讲解和代码示例](#)

模板方法在其他编程语言中的实现

主页



重构



设计模式

会员专属内容



论坛

联系我们

© 2014-2020 Refactoring.Guru. 版权所有

📍 Khmelnytske shosse 19 / 27, Kamianets-Podilskyi, 乌克兰, 32305

✉ Email: support@refactoring.guru

📷 图片作者: Dmitry Zhart

条款与政策

隐私政策

内容使用政策