Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

Отчёт

по лабораторной работе № 5
по дисциплине «Технологии и методы программирования»

Тема: «Паттерны «шаблонный метод» и «стратегия»»

Вариант 4

Выполнил:
А. В. Куликова,
студент группы ИУ8-21М
Проверил: А. Ю. Быков

Москва, 2024

1. Постановка задачи

Реализовать с помощью двух паттернов шаблонный метод и стратегия пример получения хешей некоторых данных с использование не менее двух разных хеш-функций.

При использовании шаблонного метода создается абстрактный класс (интерфейс), например, с именем Hash с абстрактным методом (чистой виртуальной функцией в языке Си++), например,

abstract String getHash(String data);

Создаются отдельные классы для получения разных хеш-функций, в которых абстрактный метод переопределяется для реализации конкретной хеш-функции. Для получения разных хеш-функций создаются объекты разных классов. Для каждого объекта вызывается метод для получения хеш-функции.

В паттерне стратегия для получения хеша в главном классе приложения можно использовать метод класса, например, вида

String getHash(String data, Hash hash)

где Hash hash ссылка, имеющая тип абстрактного класса (интерфейса), используется синтаксис Java. Внутри метода вызывается метод String getHash(String data) объявленный в Hash.

Методу String getHash(String data, Hash hash) вторым параметром передается объекта класса, производного от Hash, реализующий конкретную хеш-функцию.

В задание продемонстрировать получение разных хешей с использование разных стратегий.

2. Ход работы

Листинг 1 – Код программы user.java

```
// Абстрактный класс Hash с методом getHash
abstract class Hash {
    public abstract String getHash(String data);
}

// Класс для хеширования данных с использованием MD5
class MD5Hash extends Hash {
    @Override
    public String getHash(String data) {
        return "MD5 : " + data + "\n";
    }
}

// Класс для хеширования данных с использованием SHA-256
class SHA256Hash extends Hash {
    @Override
    public String getHash(String data) {
        return "SHA-256 : " + data + "\n";
    }
}

// Класс для хеширования данных с использованием SHA-1
class SHA1Hash extends Hash {
    @Override
    public String getHash(String data) {
        return "SHA-1 : " + data + "\n";
    }
}

// Класс для хеширования данных с использованием SHA-512
class SHA512Hash extends Hash {
    @Override
    public String getHash(String data) {
        return "SHA-512 : " + data + "\n";
    }
}

// Класс, использующий шаблонный метод и стратегию для получения хешей
class HashGenerator {
    public String getHash(String data, Hash hash) {
        System.out.println(hash);
        return hash.getHash(data);
    }
}

class Main {
```

```
    public static void main(String[] args) {
        HashGenerator generator = new HashGenerator();

        String data = "Hello";

        // Использование MD5 для хеширования
        Hash md5Hash = new MD5Hash();
        String md5Result = generator.getHash(data, md5Hash);
        System.out.println(md5Result);

        // Использование SHA-256 для хеширования
        Hash sha256Hash = new SHA256Hash();
        String sha256Result = generator.getHash(data, sha256Hash);
        System.out.println(sha256Result);

        // Использование SHA-1 для хеширования
        Hash sha1Hash = new SHA1Hash();
        String sha1Result = generator.getHash(data, sha1Hash);
        System.out.println(sha1Result);

        // Использование SHA-512 для хеширования
        Hash sha512Hash = new SHA512Hash();
        String sha512Result = generator.getHash(data, sha512Hash);
        System.out.println(sha512Result);
    }
}
```

Результат:

```
PS C:\javasing\lab5>  & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-agentlib:jdwp=transport=dt_s
ocket,server=n,suspend=y,address=localhost:53426' '--enable-preview' '-XX:+ShowCodeDetailsInExcept
ionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\7a46523ef5537550baaa2
cd7a4e6e651\redhat.java\jdt_ws\lab5_5dfc882e\bin' 'Main'
MD5Hash@6842775d
MD5 : Hello

SHA256Hash@61a52fbd
SHA-256 : Hello

SHA1Hash@63d4e2ba
SHA-1 : Hello

SHA512Hash@33a10788
SHA-512 : Hello
```