**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 3**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Работа со строками»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: ассистент

Михалевич В.Г.

Гомель 2021

**Цель работы:** изучить классы для работы со строками в *Java*.

**Задание:** Вариант 8. Условия задания на рисунке 1.

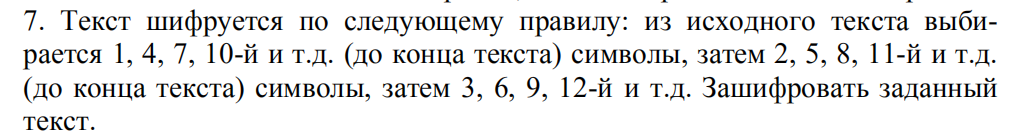




Рисунок 1 – Вариант задания

Структура созданных классов на рисунке 2.

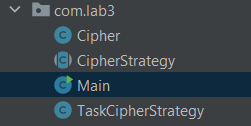


Рисунок 2 – Структура созданных классов

В работе был реализован паттерн программирования стратегия.

Сделан класс шифр, который в конструкторе принимает алгоритм шифрования. Также был создан абстрактный класс алгоритма шифрования, который содержит в себе два метода – зашифровать и расшифровать. От него был унаследован шифр, который было необходимо сделать по заданию.

В методе *main* класса *Main* был создан экземпляр класса шифр. После этого два раза был вызван метод шифрования этого экземпляра. Строки для шифрования на рисунке 3.





Рисунок 3 – Строки для шифрования

Результат выполнения задания представлен на рисунке 4.

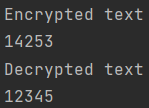
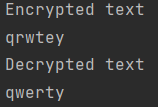


Рисунок 4 – Результат выполнения задания

**Вывод:** Были изучены методы работы со строками в *Java*.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.lab3;  
  
public abstract class CipherStrategy {  
 public abstract String encrypt(String text);  
 public abstract String decrypt(String text);  
}

package com.Lab\_1;  
  
public class Cipher {  
 CipherStrategy strategy;  
  
 public Cipher(CipherStrategy strategy) {  
 this.strategy = strategy;  
 }  
  
 public String encrypt(String text) {  
 return strategy.encrypt(text);  
 }  
  
 public String decrypt(String text) {  
 return strategy.decrypt(text);  
 }  
}

package com.Lab\_1;  
public class TaskCipherStrategy extends CipherStrategy {  
 @Override  
 public String encrypt(String text) {  
 char[] charText = text.toCharArray();  
 StringBuilder builder1 = new StringBuilder();  
 StringBuilder builder2 = new StringBuilder();  
 StringBuilder builder3 = new StringBuilder();  
 int flag = 1;  
 for(int i = 0; i < text.length(); i++) {  
 switch (flag) {  
 case 1 -> builder1.append(charText[i]);  
 case 2 -> builder2.append(charText[i]);  
 case 3 -> builder3.append(charText[i]);  
 }  
 if(flag == 3)  
 flag = 1;  
 else  
 flag += 1;  
 }  
 return builder1.toString() + builder2.toString() + builder3.toString();  
 }  
  
 @Override  
 public String decrypt(String text) {  
 char[] charText = text.toCharArray();  
 StringBuilder builder1 = new StringBuilder();  
 StringBuilder builder2 = new StringBuilder();  
 StringBuilder builder3 = new StringBuilder();  
 int flag = 1;  
 for(int i = 0; i < text.length(); i++) {  
 switch (flag) {  
 case 1 -> builder1.append(charText[i]);  
 case 2 -> builder3.append(charText[i]);  
 case 3 -> builder2.append(charText[i]);  
 }  
 if(flag == 2)  
 flag = 1;  
 else  
 flag += 1;  
 }  
 return builder1.toString() + builder2.toString() + builder3.toString();  
 }  
}

package com.Lab\_1;  
  
public class Main {  
 public static void main(String[] args) {  
 Cipher cipher = new Cipher(new TaskCipherStrategy());  
 String text = "qwerty";  
 System.out.println("Initial text:" + text);  
 System.out.println("Encrypted text");  
 text = cipher.encrypt(text);  
 System.out.println(text);  
 System.out.println("Decrypted text");  
 System.out.println(cipher.decrypt( text));  
  
  
 String text2 = "12345";  
 System.out.println("Initial text:" + text2);  
 System.out.println("Encrypted text");  
 text2 = cipher.encrypt(text2);  
 System.out.println(text2);  
 System.out.println("Decrypted text");  
 System.out.println(cipher.decrypt( text2));  
 }  
}