

Plan

2023-10-11

- 1. String
- 2. Character
- 3. Practice

- 1. String
- 2. Character
- 3. Практика

Theory

► English

▼ На русском

Задача 1.

Напишите программу, которая запрашивает строку от пользователя и затем:

- сообщает сколько слов в строке
- сообщает сколько символов в строке
- печатает строку задом наперед.
- сообщает сколько букв верхнего регистра.
- сообщает сколько букв нижнего регистра.
- сообщает сколько цифр в строке.

Задача 2

Дан текст:

Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.);It is one of the most popular programmin language in the world; It has a large demand in the current job market; It is easy to learn and simple to use; It is open-source and free; It is secure, fast and powerful; It has a huge community support (tens of millions of develope Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs; As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa;



Необходимо вывести этот текст в консоль (примерно) в таком виде вместе с заголовком Why Use Java?:

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It has a large demand in the current job market
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development c
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

Задача 3

Дан текст (текст взят с [https://en.wikipedia.org/wiki/Java\(programminglanguage\)](https://en.wikipedia.org/wiki/Java(programminglanguage))):

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementati dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to

recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

As of September 2023, Java 21 is the latest version, while Java 17, 11 and 8 are the current long-term support (LTS) versions.

Quelle: wikipedia

- Вырезать из текста полное предложение, которое начинается с `It` (до `.`)
- Проверить, начинается ли текст с `Java is a high-level` - новый метод - `startsWith()`.
- Проверить, заканчивается ли текст подстрокой `wikipedia` - новый метод - `endsWith()`.

Задание 4: программы проверки пароля

Техническое задание для программы проверки пароля

Описание:

Создайте класс `PasswordValidator` на Java для проверки пароля на соответствие требованиям, которые будут устанавливаться через конструктор класса.

Требования:

1. Пароль должен содержать минимум заданное количество букв нижнего регистра.
2. Пароль должен содержать минимум заданное количество букв верхнего регистра.
3. Пароль должен содержать минимум заданное количество цифр.
4. Пароль должен иметь заданную длину.
5. Пароль должен содержать хотя бы один из символов, указанных в списке символов.
6. Количество символов из списка должно быть не менее определенного значения.

Интерфейс:

1. Создайте класс `PasswordValidator` с полями, определенными в конструкторе:
 - `minLowerCase` (int): Минимальное количество букв нижнего регистра.
 - `minUpperCase` (int): Минимальное количество букв верхнего регистра.
 - `minDigits` (int): Минимальное количество цифр.
 - `minLength` (int): Минимальная длина пароля.
 - `symbolList` (String): Список символов, которые должны быть в пароле.
 - `minSymbolCount` (int): Минимальное количество символов из списка.
2. В классе `PasswordValidator` создайте метод `isValid`, который принимает строку (пароль) для проверки и возвращает `true`, если пароль соответствует всем требованиям, и `false` в противном случае.
3. Использование методов `Character` класса:
 - Для проверки, является ли символ буквой нижнего регистра, используйте метод `Character.isLowerCase(char c)`.
 - Для проверки, является ли символ буквой верхнего регистра, используйте метод `Character.isUpperCase(char c)`.
 - Для проверки, является ли символ цифрой, используйте метод `Character.isDigit(char c)`.

Пример использования:

```
public class Main {  
    public static void main(String[] args) {  
        int minLowerCase = 2;  
        int minUpperCase = 2;
```

```

    int minDigits = 1;
    int minLength = 12;
    String symbolList = "!@#$$%^";
    int minSymbolCount = 2;

    PasswordValidator validator = new PasswordValidator(minLowerCase, minUpperCase, minDigits, minLength, symbolList);

    String password = "MyP@ssword123";
    boolean isValid = validator.isValid(password);

    if (isValid) {
        System.out.println("Пароль верный.");
    } else {
        System.out.println("Пароль не соответствует требованиям.");
    }
}
}

```

Скелет класса PasswordValidator:

```

/**
 * @author Andrej Reutow
 * created on 09.10.2023
 * <p>
 * Класс для проверки пароля на соответствие заданным требованиям.
 */
public class PasswordValidator {

    private final int minLowerCase;
    private final int minUpperCase;
    private final int minDigits;
    private final int minLength;
    private final String symbolList;
    private final int minSymbolCount;

    /**
     * Конструктор класса PasswordValidator для инициализации параметров проверки пароля.
     *
     * @param minLowerCase Минимальное количество букв нижнего регистра.
     * @param minUpperCase Минимальное количество букв верхнего регистра.
     * @param minDigits Минимальное количество цифр.
     * @param minLength Минимальная длина пароля.
     * @param symbolList Список символов, которые должны быть в пароле.
     * @param minSymbolCount Минимальное количество символов из списка.
     */
    public PasswordValidator(int minLowerCase,
                             int minUpperCase,
                             int minDigits,
                             int minLength,
                             String symbolList,
                             int minSymbolCount) {

    }

    /**
     * Проверяет, соответствует ли заданный пароль требованиям.
     *
     * @param password Пароль для проверки.
     * @return true, если пароль соответствует требованиям, и false в противном случае.
     */
    public boolean isValid(String password) {

        return false;
    }
}

```

```
/**
 * Проверяет, содержит ли пароль заданное количество цифр.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество цифр, и false в противном случае.
 */
private boolean isDigitsContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданные символы из списка.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов из списка, и false в противном случае.
 */
private boolean isSymbolsContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданное количество символов верхнего регистра.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов верхнего регистра, и false в противном случае.
 */
private boolean isUpperCaseContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданное количество символов нижнего регистра.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов нижнего регистра, и false в противном случае.
 */
private boolean isLowerCaseContains(String password) {

    return false;
}

/**
 * Проверяет, является ли длина пароля достаточной.
 *
 * @param password Пароль для проверки.
 * @return true, если длина пароля больше или равна минимальной длине, и false в противном случае.
 */
private boolean isLengthValid(String password) {

    return -1;
}
}
```

Homework

► English

▼ На русском

Задание 1

Дано предложение:

Что такое технология Java и каково ее применение?

Java представляет собой язык программирования и платформу вычислений, которая была впервые выпущена компанией Sun Microsystems в 1995 г. Технология эволюционировала из скромной разработки до инструмента, который играет серьезную роль в современном цифровом мире, предоставляя надежную платформу для множества сервисов и приложений. Инновационные продукты и цифровые услуги, разрабатываемые для будущего, также будут создаваться на основе Java.

Несмотря на то что большинство современных приложений Java объединяет в себе среду и приложение Java, все еще существует множество приложений и даже некоторые веб-сайты, которые не будут работать, если на компьютере не установлена настольная версия Java. Java.com — это веб-сайт, предназначенный для пользователей, которым, возможно, Java по-прежнему требуется для настольных приложений, в частности для приложений на Java 8. Более подробную информацию для разработчиков и пользователей, которые хотели бы научиться программированию на Java, см. на веб-сайте dev.java, для бизнес-пользователей — на веб-странице oracle.com/java.

- Напишите метод, который отвечает на вопрос сколько символов в этом предложении.
- Подсчитайте, сколько букв (не символов и не цифр и не знаков!) в этом предложении.
- Подсчитайте, сколько букв латинского алфавита в этом предложении.
- Подсчитайте, сколько букв кириллического алфавита в этом предложении.

Задание 2* Программа кодирования и декодирования пароля

Техническое задание

Цель:

Разработать программу на Java для кодирования и декодирования пароля с использованием следующих условий:

- Каждый четный индекс символа пароля сдвигается на 1000 символов вправо в Unicode.
- Каждый нечетный индекс символа пароля сдвигается на 10 символов влево в Unicode.
- В закодированный пароль добавляются 6 случайных символов из таблицы Unicode.

Функциональные требования:

1. Программа должна иметь два основных метода: `encodePassword` для кодирования пароля и `decodePassword` для декодирования пароля.
2. Метод `encodePassword` должен принимать строку с паролем в качестве входных данных и возвращать закодированный пароль в виде строки.
3. Метод `decodePassword` должен принимать закодированный пароль в виде строки и возвращать исходный пароль в виде строки.
4. Программа должна добавлять 6 случайных символов из Unicode в закодированный пароль при каждой кодировке пароля.

Нефункциональные требования:

1. Программа должна быть простой в использовании через текстовый интерфейс в консоли.
2. Программа должна быть легко читаемой и поддерживаемой.
3. Программа должна быть протестирована (smoke test - https://ru.wikipedia.org/wiki/Smoke_test).

Code

code/PasswordValidator_25/src/PasswordValidatorApp.java

```
/**
 * @author Andrej Reutow
 * created on 09.10.2023
 */
public class PasswordValidatorApp {

    public static void main(String[] args) {
        PasswordValidator passwordValidator = new PasswordValidator(
            2, 2, 2, 12, "#!/*-", 2
        );

        String pass1 = "!IAmValidPass007!";
        String pass2 = "notvalid";
        String pass3 = "notvalid007";
```

```

String pass4 = "notValid007";
String pass5 = "007";
String pass6 = "007!";
String pass7 = "n";
String pass8 = "NOTVALID";
String pass9 = "NOTVALID007";
String pass10 = "NOTVALID007!";
String pass11 = "!";
String pass12 = "IAmNotValidPass007!";

System.out.println("validate pass: " + pass1 + " result: " + passwordValidator.isValid(pass1));
System.out.println("validate pass: " + pass2 + " result: " + passwordValidator.isValid(pass2));
System.out.println("validate pass: " + pass3 + " result: " + passwordValidator.isValid(pass3));
System.out.println("validate pass: " + pass4 + " result: " + passwordValidator.isValid(pass4));
System.out.println("validate pass: " + pass5 + " result: " + passwordValidator.isValid(pass5));
System.out.println("validate pass: " + pass6 + " result: " + passwordValidator.isValid(pass6));
System.out.println("validate pass: " + pass7 + " result: " + passwordValidator.isValid(pass7));
System.out.println("validate pass: " + pass8 + " result: " + passwordValidator.isValid(pass8));
System.out.println("validate pass: " + pass9 + " result: " + passwordValidator.isValid(pass9));
System.out.println("validate pass: " + pass10 + " result: " + passwordValidator.isValid(pass10));
System.out.println("validate pass: " + pass11 + " result: " + passwordValidator.isValid(pass11));
System.out.println("validate pass: " + pass12 + " result: " + passwordValidator.isValid(pass12));
    }
}

```

code/PasswordValidator_25/src/PasswordValidator.java

```

/**
 * @author Andrej Reutow
 * created on 09.10.2023
 * <p>
 * Класс для проверки пароля на соответствие заданным требованиям.
 */
public class PasswordValidator {

    private final int minLowerCase;
    private final int minUpperCase;
    private final int minDigits;
    private final int minLength;
    private final String symbolList;
    private final int minSymbolCount;

    /**
     * Конструктор класса PasswordValidator для инициализации параметров проверки пароля.
     *
     * @param minLowerCase Минимальное количество букв нижнего регистра.
     * @param minUpperCase Минимальное количество букв верхнего регистра.
     * @param minDigits Минимальное количество цифр.
     * @param minLength Минимальная длина пароля.
     * @param symbolList Список символов, которые должны быть в пароле.
     * @param minSymbolCount Минимальное количество символов из списка.
     */
    public PasswordValidator(int minLowerCase,
                             int minUpperCase,
                             int minDigits,
                             int minLength,
                             String symbolList,
                             int minSymbolCount) {

        this.minDigits = 0;
        this.minLength = 0;
        this.symbolList = null;
        this.minSymbolCount = 0;
        this.minLowerCase = 0;
        this.minUpperCase = 0;
    }
}

```

```
}

/**
 * Проверяет, соответствует ли заданный пароль требованиям.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль соответствует требованиям, и false в противном случае.
 */
public boolean isValid(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданное количество цифр.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество цифр, и false в противном случае.
 */
private boolean isDigitsContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданные символы из списка.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов из списка, и false в противном случае.
 */
private boolean isSymbolsContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданное количество символов верхнего регистра.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов верхнего регистра, и false в противном случае.
 */
private boolean isUpperCaseContains(String password) {

    return false;
}

/**
 * Проверяет, содержит ли пароль заданное количество символов нижнего регистра.
 *
 * @param password Пароль для проверки.
 * @return true, если пароль содержит заданное количество символов нижнего регистра, и false в противном случае.
 */
private boolean isLowerCaseContains(String password) {

    return false;
}

/**
 * Проверяет, является ли длина пароля достаточной.
 *
 * @param password Пароль для проверки.
 * @return true, если длина пароля больше или равна минимальной длине, и false в противном случае.
 */
private boolean isLengthValid(String password) {

    return false;
}
```

```
}  
}
```

code/ClassWork_26/src/HwSolution_25.java

```
/**  
 * @author Andrej Reutow  
 * created on 11.10.2023  
 */  
  
/**  
 * Реализуйте метод, который принимает строку и возвращает новую строку, в которой все слова перевернуты, но порядок  
 */  
public class HwSolution_25 {  
  
    public static String reversedStrings(String value) {  
        // Easy come easy go  
        // ysaE emoc ysae og  
        String[] words = value.split(" ");  
        String result = "";  
        for (int i = 0; i < words.length; i++) {  
            String word = words[i];  
            String reversedWord = reverseString(word);  
            result = result + reversedWord + " ";  
  
            if (i == words.length - 1) {  
                result = reversedWord;  
            } else {  
                result = reversedWord + " ";  
            }  
        }  
  
        return result.trim();  
    }  
  
    private static String reverseString(final String text) {  
        // Easy  
        // ysaE  
        String result = "";  
        for (int i = text.length() - 1; i >= 0; i--) {  
            result += text.charAt(i);  
        }  
  
        return result;  
    }  
}
```

code/ClassWork_26/src/HwSolutionApp.java

```
/**  
 * @author Andrej Reutow  
 * created on 11.10.2023  
 */  
public class HwSolutionApp {  
  
    public static void main(String[] args) {  
  
        String result1 = HwSolution_25.reversedStrings("Easy come easy go");  
        String expectedResult1 = "ysaE emoc ysae og";  
  
        System.out.println(result1); //ysaE emoc ysae og  
        System.out.println(expectedResult1.equals(result1)); // true  
    }  
}
```



```
}  
}
```

code/Lesson_26/src/task1/Task1Runner.java

```
package task1;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
/**  
 * @author Andrej Reutow  
 * created on 11.10.2023  
 */  
public class Task1Runner {  
  
    private static final Scanner SCANNER = new Scanner(System.in);  
  
    public static void main(String[] args) {  
        //      System.out.println("Input some text 1");  
        //      String line1 = readLine();  
        //  
        //      System.out.println("Input some text 1");  
        //      String line2 = readLine();  
        //  
        //      String[] strings = new String[2];  
        //      strings[0] = line1;  
        //      strings[1] = line2;  
        //  
        //      System.out.println(Arrays.toString(strings));  
        //  
        //      for (int i = 0; i < strings.length; i++) {  
        //          final String currentLine = strings[i];  
        //          final int wordCount = TaskUtils.wordCount(currentLine);  
        //          final String reversedLine = TaskUtils.reverseString(currentLine);  
        //          final int digitsCaseLetters = TaskUtils.countDigits(currentLine);  
        //          final int countLowerCaseLetters = TaskUtils.countLowerCaseLetters(currentLine);  
        //          final int countUpperCaseLetters = TaskUtils.countUpperCaseLetters(currentLine);  
        //          int countChars = TaskUtils.countChars(currentLine);  
        //  
        //          System.out.println("Line:\n" + currentLine);  
        //          System.out.println("wordCount: " + wordCount);  
        //          System.out.println("reversedLine: " + reversedLine);  
        //          System.out.println("digitsCaseLetters: " + digitsCaseLetters);  
        //          System.out.println("countLowerCaseLetters: " + countLowerCaseLetters);  
        //          System.out.println("countUpperCaseLetters: " + countUpperCaseLetters);  
        //          System.out.println("countChars: " + countChars);  
        //  
        //      }  
  
        char pointChar = '\u2022';  
        System.out.println(pointChar + " " + "someString1");  
        System.out.println(pointChar + " " + "someString2");  
        System.out.println(pointChar + " " + "someString3");  
    }  
  
    private static String readLine() {  
        return SCANNER.nextLine();  
    }  
}
```

code/Lesson_26/src/task1/TaskUtils.java

```
package task1;

/**
 * @author Andrej Reutow
 * created on 11.10.2023
 */
public class TaskUtils {
    //
    //  Задача 1.
    //  Напишите программу, которая запрашивает строку от пользователя и затем:
    //  сообщает сколько слов в строке
    //  сообщает сколько символов в строке
    //  печатает строку задом наперед.
    //  сообщает сколько букв верхнего регистра.
    //  сообщает сколько букв нижнего регистра.
    //  сообщает сколько цифр в строке.

    /**
     * сообщает сколько слов в строке
     */
    public static int wordCount(final String line) {
        final String[] words = line.split(" ");
        return words.length;
    }

    public static int countChars(final String line) {
        return line.length();
    }

    public static String reverseString(final String line) {
        String result = "";
        for (int i = line.length() - 1; i >= 0; i--) {
            result += line.charAt(i);
        }
        return result;
    }

    public static int countUpperCaseLetters(final String line) {
        int counter = 0;
        for (int i = 0; i < line.length(); i++) {
            if (line.charAt(i) >= 'A' && line.charAt(i) <= 'Z') {
                counter++;
            }
        }
        return counter;
    }

    public static int countLowerCaseLetters(final String line) {
        int counter = 0;
        for (int i = 0; i < line.length(); i++) {
            if (Character.isLowerCase(line.charAt(i))) {
                counter++;
            }
        }
        return counter;
    }

    public static int countDigits(final String line) {
        int counter = 0;
        for (int i = 0; i < line.length(); i++) {
            if (Character.isDigit(line.charAt(i))) {
                counter++;
            }
        }
        if (line.charAt(i) >= '0' && line.charAt(i) <= '9') {

```

```
//          counter++;
//      }
    }
    return counter;
}
}
```

code/Lesson_26/src/task2/TextEditorUtils.java

```
package task2;

/**
 * @author Andrej Reutow
 * created on 10.10.2023
 */
public class TextEditorUtils {

    public static void main(String[] args) {
        //      final String text = "Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.);It is one
        //
        final String text = "Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.);It is one o
            "language in the world; It has a large demand in the current job market; It is easy to learn and sim
            "open-source and free; It is secure, fast and powerful; It has a huge community support (tens of mil
            "Java is an object oriented language which gives a clear structure to programs and allows code to be
            "development costs; As Java is close to C++ and C#, it makes it easy for programmers to switch to Ja

        String result = text.replace("\n", "");
        String[] strings = result.split(";");

        System.out.println("Why Use Java?");
        char pointChar = '\u2022';

        for (int i = 0; i < strings.length; i++) {
            System.out.println(pointChar + " " + strings[i].trim());
        }

    }
}
```

code/Lesson_26/src/task3/TextEditorUtils.java

```
package task3;

/**
 * @author Andrej Reutow
 * created on 10.10.2023
 */
public class TextEditorUtils {

    public static void main(String[] args) {
        String text = "Java is a high-level, class-based, object-oriented programming language that is designed to h
            "dependencies as possible. It is a general-purpose programming language intended to let programmers
            "anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without
            "recompile. Java applications are typically compiled to bytecode that can run on any Java virtual ma
            "regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but
            "facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection
            "modification) that are typically not available in traditional compiled languages. As of 2019, Java
            "popular programming languages in use according to GitHub, particularly for client-server web\n" +
            "applications, with a reported 9 million developers.\n" +
            "\n" +
            "Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as
            "Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual
            "libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance
```

```
"specifications of the Java Community Process, Sun had relicensed most of its Java technologies unde
"license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference impleme
"OpenJDK JVM which is free open-source software and used by most developers and is the default JVM f
"distributions.\n" +
"\n" +
"As of September 2023, Java 21 is the latest version, while Java 17, 11 and 8 are the current long-t
"versions.\n" +
"\n" +
"Quelle: wikipedia";

//      work(text);
//      work2(text);

System.out.println("Find word Java at start:" + startsWithWord(text, "Java"));
System.out.println("Find word Java at end:" + endsWithWord(text, "Quelle: wikipedia"));
}

public static void work(String source) {
    String[] sent = source.split("\\.");
    for (int i = 0; i < sent.length; i++) {
        String current = sent[i].trim();
        if (current.startsWith("It")) {
            System.out.println(current);
        }
    }
}

public static void work2(String source) {
    String[] sent = source.split("\\.");
    for (int i = 0; i < sent.length; i++) {
        String current = sent[i].trim();

        String[] words = current.split(" ");

        if (words[0].equals("It")) {
            System.out.println(current);
        }
    }
}

public static boolean startsWithWord(String source, String startText) {
    return source.startsWith(startText);
}

public static boolean endsWithWord(String source, String endText) {
    return source.endsWith(endText);
}
}
```