# Theory

Main lessons

* The Java Tutorial. [Strings](https://docs.oracle.com/javase/tutorial/java/data/strings.html).
* The Java Tutorial. [Regular Expressions](https://docs.oracle.com/javase/tutorial/essential/regex/index.html).

Advanced lessons

* И. Блинов. Глава 7. Строки.
* skipy.ru. Ах, эти строки...

### Дедлайн этапа 2

**Принята задача collections1\_2.**

Следствие. Задача **collections2** не будет обрабатываться скриптом.

# Questions

Вопрос 1.

Что появится в консоли в результате работы фрагмента программы?

String a = "java";

a.toUpperCase();

System.out.println(a);

Ответ. "JAVA";

Источник.

Вопрос 2.

Что появится в консоли в результате работы фрагмента программы?

String s1 = "Java";

String s2 = "Java";

String s3 = new String("Java");

System.out.println("s1 == s2 : " + (s1 == s2));

System.out.println("s1 == s3 : " + (s1 == s3));

System.out.println(s1.equals(s3));

Поясните ответ.

Ответ.

s1 == s2 : true

s1 == s3 : false

true

Источник.

Вопрос 3.

Можно ли выполнить наследование от класса String?

Почему?

Ответ. Класс **String** в Java — это **final** класс, который не может иметь потомков

Источник. <https://javarush.ru/groups/posts/760-java-string-voprosih-k-sobesedovaniju-i-otvetih-na-nikh-ch2>

Вопрос 4.

Назовите основные, на ваш взгляд, методы класса String.

Ответ.

* concat(): объединяет строки
* valueOf(): преобразует объект в строковый вид
* join(): соединяет строки с учетом разделителя
* сompare(): сравнивает две строки
* charAt(): возвращает символ строки по индексу
* getChars(): возвращает группу символов
* equals(): сравнивает строки с учетом регистра
* equalsIgnoreCase(): сравнивает строки без учета регистра
* regionMatches(): сравнивает подстроки в строках
* indexOf(): находит индекс первого вхождения подстроки в строку
* lastIndexOf(): находит индекс последнего вхождения подстроки в строку
* startsWith(): определяет, начинается ли строка с подстроки
* endsWith(): определяет, заканчивается ли строка на определенную подстроку
* replace(): заменяет в строке одну подстроку на другую
* trim(): удаляет начальные и конечные пробелы
* substring(): возвращает подстроку, начиная с определенного индекса до конца или до определенного индекса
* toLowerCase(): переводит все символы строки в нижний регистр
* toUpperCase(): переводит все символы строки в верхний регистр

Источник. <https://metanit.com/java/tutorial/7.1.php>

Вопрос 5.

Какие разновидности конструкторов использует класс String?

Ответ.

public String()

Инициализирует вновь созданный объект String так, чтобы он представлял пустую последовательность символов. Обратите внимание, что в использовании этого конструктора нет необходимости, поскольку строки неизменяемы.

public String(String original)

Инициализирует вновь созданный объект String, чтобы он представлял ту же последовательность символов, что и аргумент; другими словами, вновь созданная строка является копией строки аргумента. Если явная копия оригинала не требуется, в использовании этого конструктора нет необходимости, поскольку строки неизменяемы.

public String(char[] value)

Выделяет новую строку таким образом, чтобы она представляла последовательность символов, которая в настоящее время содержится в аргументе массива символов. Копируется содержимое символьного массива; последующая модификация массива символов не влияет на вновь созданную строку.

public String(byte[] bytes,

Charset charset)

Создает новую строку путем декодирования указанного массива байтов с использованием указанной кодировки.

public String(StringBuilder builder)

Этот конструктор предназначен для упрощения перехода на StringBuilder.

public String(StringBuffer buffer)

Выделяет новую строку, содержащую последовательность символов, которая в данный момент содержится в аргументе строкового буфера

Источник. [https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#String--](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html%23String--)

Вопрос 6.

Какие классы в стандартной библиотеке Java работают со строками?

Ответ. Реализация строк на Java представлена тремя основными классами: String, StringBuffer, StringBuilder.

Источник. <https://habr.com/ru/post/260767/>

Вопрос 7.

Почему экземпляры класса String в Java неизменные и финализированные?

Ответ.

Есть несколько преимуществ в неизменности строк:

1. Строковый пул возможен только потому, что строка неизменна в Java, таким образом виртуальная машина сохраняет много места в памяти(heap space), поскольку разные строковые переменные указывают на одну переменную в пуле. Если бы строка не была неизмененяемой, тогда бы интернирование строк не было бы возможным, потому что если какая-либо переменная изменит значение, это отразится также и на остальных переменных, ссылающихся на эту строку.
2. Если строка будет изменяемой, тогда это станет серьезной угрозой безопасности приложения. Например, имя пользователя базы данных и пароль передаются строкой для получения соединения с базой данных и в программировании сокетов реквизиты хоста и порта передаются строкой. Так как строка неизменяемая, её значение не может быть изменено, в противном случае любой хакер может изменить значение ссылки и вызвать проблемы в безопасности приложения.
3. Так как строка неизменная, она безопасна для многопоточности и один экземпляр строки может быть совместно использован различными потоками. Это позволяет избежать синхронизации для потокобезопасности, строки полностью потокобезопасны.
4. Строки используются в Java classloader и неизменность обеспечивает правильность загрузки класса при помощи Classloader. К примеру, задумайтесь об экземпляре класса, когда вы пытаетесь загрузить java.sql.Connection класс, но значение ссылки изменено на myhacked.Connection класс, который может осуществить нежелательные вещи с вашей базой данных.
5. Поскольку строка неизменная, её hashcode кэшируется в момент создания и нет необходимости рассчитывать его снова. Это делает строку отличным кандидатом для ключа в Map и его обработка будет быстрее, чем других ключей HashMap. Это причина, почему строка наиболее часто используемый объект в качестве ключа HashMap.

Источник. <https://javastudy.ru/interview/strings/>

Вопрос 8.

Заполните ячейки таблицы (Да/Нет).

Characteristic String StringBuilder StringBuffer

Неизменяемый (Immutable)? Да Нет Нет

Имеет пул (Pooled)? Да Нет Нет

Потокобезопасный (Thread-safe)? Да Нет Да

Может изменять размер? Нет Да Да

Источник. <https://javarush.ru/groups/posts/760-java-string-voprosih-k-sobesedovaniju-i-otvetih-na-nikh-ch2>

Вопрос 9.

В чем разница и что общего между StringBuffer и StringBuilder?

Вопрос 10.

Когда лучше использовать StringBuffer, а когда StringBuilder?

Ответ. Операции со StringBuffer, в отличие от StringBuilder, потокобезопасны и синхронизированы. Поэтому в случаях, когда несколько потоков должны работать с одной строкой, то необходимо использовать StringBuffer. Однако в однопоточных окружениях желательно использовать StringBuilder, поскольку преимущество StringBuilder перед StringBuffer заключается в производительности из-за отсутствия затрат для поддержки синхронизации.

Источник. <https://divancoder.ru/2017/06/string-stringbuffer-stringbuilder/>

Вопрос 11.

Какие методы имеются в классах StringBuffer и StringBuilder, которые отсутствуют в классе String?

Ответ.

- Метод append() добавляет подстроку в конец StringBuffer

- Метод insert() добавляет строку или символ по определенному индексу в StringBuffer

- Метод delete() удаляет все символы с определенного индекса о определенной позиции

- Для замены подстроки между определенными позициями в StringBuffer на другую подстроку применяется метод replace()

Источник. <https://metanit.com/java/tutorial/7.3.php>

Вопрос 12.

Какие методы сравнения строк имеются в строковых классах?

Ответ.

сompare(): сравнивает две строки

equals(): сравнивает строки с учетом регистра

equalsIgnoreCase(): сравнивает строки без учета регистра

regionMatches(): сравнивает подстроки в строках

Источник. <https://metanit.com/java/tutorial/7.1.php>

Вопрос 13.

Используя функции строковых классов, написать фрагмент программы, которая будет определять, является ли строка палиндромом.

Ответ.

String variable = ""; #write a string name

StringBuffer rev = new StringBuffer(variable).reverse();

String strRev = rev.toString();

if(variable.equalsIgnoreCase(strRev)) # Check the condition

Источник. <https://overcoder.net/q/524188/java-%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1-%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%B8%D1%82%D1%8C-%D1%8F%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%D1%81%D1%8F-%D0%BB%D0%B8-%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B0-%D0%BF%D0%B0%D0%BB%D0%B8%D0%BD%D0%B4%D1%80%D0%BE%D0%BC%D0%BE%D0%BC>

Вопрос 14.

Что появится в консоли в результате работы фрагмента программы?

final String ZA = " за ";

String value = "ОТЧЕТ о перевозках пассажиров за январь 2019 г.";

String[] monthYear = value.split(ZA)[1].split(" ",3);

System.out.println(Arrays.toString(monthYear));

Поясните ответ.

Ответ. [январь, 2019, г.]

В строке предложение перед ZA вырезалось со значением 1, но в массиве первый номер это ноль, поэтому при выводе вывелись 3 элемента так, как отсчет шел со слова «январь», разделяя при этом три последних слова в предложении по пробелу

Источник. <https://javarush.ru/groups/posts/2907-metod-split-v-java-delim-stroku-na-chasti>

Вопрос 15.

Что появится в консоли в результате работы фрагмента программы?

String s4 = "1" + 2 + 3;

String s5 = 1 + 2 + "3";

System.out.println(s4);

System.out.println(s5);

Поясните ответ.

Ответ. После вывода строки арифметические действия не производятся, поэтому в первом случае вывело цифры подряд, а во втором случае числа сложились.

123

33

Источник.

Вопрос 16.

Что появится в консоли в результате работы фрагмента программы?

String s = "abcde ";

System.out.println(s.trim().length());

System.out.println(s.charAt(4));

System.out.println(s.indexOf('e'));

System.out.println(s.indexOf("de"));

System.out.println(s.substring(2, 4).toUpperCase());

System.out.println(s.replace('a', '1'));

System.out.println(s.contains("DE"));

System.out.println(s.startsWith("a"));

Поясните ответ.

Ответ.

5 – вывод числа элементов строки с удалением конечного пробела

E - возврат символа строки по индексу

4 - индекс первого вхождения подстроки в строку

3 - индекс первого вхождения подстроки в строку две буквы считались за 1 элемент

CD – возврат элементов по индексу, а так же перевод их в верхний регистр

1bcde – замена элемента на заданный

False – проверка содержит ли строка заданный элемент

True – проверка начинается ли элемент с заданного

Источник. <https://metanit.com/java/tutorial/7.1.php>

Вопрос 17.

Что появится в консоли в результате работы фрагмента программы?

StringBuilder b = new StringBuilder();

b.append(12345).append('-');

System.out.println(b.length());

System.out.println(b.indexOf("-"));

System.out.println(b.charAt(2));

StringBuilder b2 = b.reverse();

System.out.println(b.toString());

System.out.println(b == b2);

Поясните ответ.

Ответ.

6 – вывод числа элементов

5 – индекс первого вхождения подстроки в строку

3 – возврат символа строки по индексу

-54321 – возврат элементов в обратном порядке

True – проверка равенства двух значений

Источник.

Вопрос 18.

Что появится в консоли в результате работы фрагмента программы?

StringBuilder s = new StringBuilder("abcde"); a-bcde

s.insert(1, '-').delete(3, 4);

System.out.println(s);

System.out.println(s.substring(2, 4));

Поясните ответ.

Ответ.

a-bde – вставка на место первого элемента «1» но при удалении отсчет сдвинулся и 3 элемент удалился, а 4 нет.

bd – возврат элементов по индексу

Источник.

Вопрос 19.

StringBuffer sb = new StringBuffer("abcde");

sb.insert(2,"123");

sb.append("456");

sb.reverse();

Поясните ответ.

Ответ. Нет строки вывода результата

Источник.

Вопрос 20.

Каким образом можно сцепить строки Java?

Назовите не менее 3 способов.

Ответ.

+

.concat():

.join()

Источник.

Вопрос 21.

Чем отличаются пустая и нулевая строки?

Ответ.

null по существу "ничего" - это значение по умолчанию (использовать термин "свободно" ), которое Java присваивает любой переменной Object, которая не была инициализирована.

null не является действительно значением - и, как таковой, не имеет свойств. Таким образом, вызов чего-либо, предназначенного для возврата значения - например, .length(), всегда будет возвращать ошибку, потому что "ничего" не может иметь свойств.

Чтобы углубиться, создав s1 = "";, вы инициализируете объект, который может иметь свойства, и занимает соответствующее пространство в памяти. Используя s2;, вы указываете, что имя переменной является строкой, но на самом деле не назначаете никакого значения в этой точке.

Источник. <https://overcoder.net/q/201550/%D1%80%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-%D0%BD%D1%83%D0%BB%D0%B5%D0%B2%D0%BE%D0%B9-%D0%B8-%D0%BF%D1%83%D1%81%D1%82%D0%BE%D0%B9-%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%BE%D0%B9>

Вопрос 22.

В какой кодировке хранятся символы в строке?

Ответ. Первоначально Java использовал UCS-2 внутри; теперь он использует UTF-16.

Источник. <https://overcoder.net/q/161407/%D1%87%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-%D0%BA%D0%BE%D0%B4%D0%B8%D1%80%D0%BE%D0%B2%D0%BA%D0%B0-%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%BE%D0%B2-string-%D0%B2-java>

Вопрос 23.

Какие интерфейсы реализуют классы String, StringBuffer и StringBuilder?

Ответ. Классы String, StringBuffer, StringBuilder определены в пакете java.lang и доступны автоматически без объявления импорта. Все три класса реализуют интерфейс CharSequence.

Источник. <http://developer.alexanderklimov.ru/android/java/string.php>

Вопрос 24.

Что такое кодовые точки и кодовые единицы?

Ответ. Единица кода - это 16-битный двоичный код, а кодовая точка - это один или два 16-битных двоичного кода. То есть одна кодовая точка может быть выражена как одна кодовая единица или две кодовые единицы.

Источник. <https://russianblogs.com/article/56331560288/>

Вопрос 25.

Объясните назначение метода intern().

Что появится в консоли в результате работы фрагмента программы?

class GFG {

public static void main(String[] args) {

String s1 = new String("GFG");

String s2 = s1.intern();

System.out.println(s1 == s2);

System.out.println(s1.equals(s2));

String s3 = "GFG";

System.out.println(s2 == s3);

}

}

Поясните ответ.

Ответ.

Когда метод intern() вызван, если пул строк уже содержит строку, эквивалентную к нашему объекту, что подтверждается методом equals(Object), тогда возвращается ссылка на строку из пула. В противном случае объект строки добавляется в пул и ссылка на этот объект возвращается. Этот метод всегда возвращает строку, которая имеет то же значение, что что и текущая строка, но гарантирует что это будет строка из пула уникальных строк.

false

true

true

Источник. <https://javarush.ru/groups/posts/760-java-string-voprosih-k-sobesedovaniju-i-otvetih-na-nikh-ch2>

Вопрос 26.

Как преобразовать строку в число?

Ответ. int i = Integer.parseInt(stringElement);

Источник.

Вопрос 27.

Какой метод вызывается для преобразования переменной в строку?

Ответ. Метод toString() — используется в Java для получения строкового объекта, представляющего значение числового объекта, другими словами — преобразует число в строку.

Если метод принимает в качестве аргумента простой тип данных, то возвращается строковый объект, представляющий значение простого типа данных.

Источник. <http://proglang.su/java/numbers-tostring>

Вопрос 28.

Каким методом в классе String можно проверить строку на соответствие регулярному выражению?

Ответ. Метод класса String - matches() принимает регулярное выражение и возвращает true, если строка соответствует этому выражению. Иначе возвращает false.

Источник. <https://metanit.com/java/tutorial/7.4.php>

Вопрос 29.

Создайте класс с полями: int, long, float и double, String.

Сделайте для этого класса метод toString(), использующий String.format(), и выведите содержимое полей на экран в произвольном порядке, используя спецификаторы.

Ответ.

Источник.

Вопрос 30.

Опишите:

* назначение класса Formatter,
* методы format(),
* спецификаторы формата.

Ответ.

Источник.

Вопрос 31.

Что представляет собой регулярное выражение?

На каких классах базируются regex-возможности языка Java?

В каком пакете эти классы расположены?

Ответ.

Регулярное выражение (RegEx) – это шаблон для поиска строки в тексте. В Java исходным представлением этого шаблона всегда является строка, то есть объект класса String. Однако не любая строка может быть скомпилирована в регулярное выражение, а только та, которая соответствует правилам написания регулярного выражения – синтаксису, определенному в спецификации языка.

Для написания регулярного выражения используются буквенные и цифровые символы, а также метасимволы – символы, имеющие специальное значение в синтаксисе регулярных выражений. Большая часть функциональности по работе с регулярными выражениями в Java сосредоточена в пакете java.util.regex.

Источник. <https://javarush.ru/groups/posts/regulyarnye-vyrazheniya-v-java>

Вопрос 32.

Опишите назначение классов Pattern и Matcher.

Каким образом они связываются?

Предоставить простейший код их взаимодействия.

Ответ. Само регулярное выражение представляет шаблон для поиска совпадений в строке. Для задания подобного шаблона и поиска подстрок в строке, которые удовлетворяют данному шаблону, в Java определены классы Pattern и Matcher.

Для простого поиска соответствий в классе Pattern определен статический метод boolean matches(String pattern, CharSequence input). Данный метод возвращает true, если последовательность символов input полностью соответствует шаблону строки pattern:

Но, как правило, для поиска соответствий применяется другой способ - использование класса Matcher.

Используем класс Matcher. Для этого вначале надо создать объект Pattern с помощью статического метода compile(), который позволяет установить шаблон:

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class StringsApp {

public static void main(String[] args) {

String input = "Hello Java! Hello JavaScript! JavaSE 8.";

Pattern pattern = Pattern.compile("Java(\\w\*)");

Matcher matcher = pattern.matcher(input);

while(matcher.find())

System.out.println(matcher.group());

}

}

Источник. <https://metanit.com/java/tutorial/7.4.php>

Вопрос 33.

При помощи регулярных выражений найти все ссылки внутри веб-страницы, то есть адреса, указанные в атрибуте href.

Ответ.

Источник.

Вопрос 34.

Какой из способов сравнения строк предпочтительнее?

str.equals("abc");

или

"abc".equals(str);

Поясните ответ.

Ответ. 1, потому что правильнее сравнивать соответствует ли переменная значению в скобках, а не наоборот.

Источник.

Вопрос 35.

Как сравнить объекты StringBuilder и StringBuffer?

Ответ.

Источник.

Вопрос 36.

Что появится в консоли в результате работы фрагмента программы?

System.out.printf(

"%d студентов пришли сдавать зачет по %8.10s, \n"

+ "из них не сдали %d, %s",

10, "J2SE", 5, "а остальные сдали на отлично"

);

Поясните ответ.

Ответ.

10 студентов пришли сдавать зачет по J2SE,

из них не сдали 5, а остальные сдали на отлично

В конце строки есть элементы, которые при выводе занимают места соответствующие формату, указанному в ковычках.

Источник.

Вопрос 37.

Сформировать регулярное выражение, при помощи которого можно выбрать все страницы, начинающиеся с "http" или "https" и заканчивающиеся на "by".

Ответ.

Источник.

Вопрос 38.

Сформировать регулярное выражение, при помощи которого можно выбрать все значения URL-запроса с ресурса https://bsut.by.

Например, запрос https://bsut.by?value1=5&value2=10&iscan=true.

Вывод

value1=5

value2=10

iscan=true

Ответ.

Источник.

Вопрос 39.

Базовые конструкции регулярных выражений, синтаксис (очень коротко, суть). Специальные символы, основные логические конструкции, предопределенные классы символов, логические операции, квантификаторы, группы.

Ответ.

Источник.

Вопрос 40.

Назовите основные, на ваш взгляд, методы класса Pattern.

Ответ.

* String pattern() – возвращает исходное строковое представление регулярного выражения, из которого был создан объект
* static boolean matches(String regex, CharSequence input) – позволяет проверить регулярное выражение, переданное в параметре regex на соответствие тексту, переданному в параметре input
* int flags() – возвращает значения параметра flags шаблона, которые были установлены при его создании, или 0, если этот параметр не был установлен

Источник. <https://javarush.ru/groups/posts/regulyarnye-vyrazheniya-v-java>

Вопрос 41.

Назовите основные, на ваш взгляд, методы класса Matcher.

Ответ.

* boolean matches(): возвращает true, если вся строка совпадает с шаблоном
* boolean find(): возвращает true, если в строке есть подстрока, которая совпадает с шаблоном, и переходит к этой подстроке
* String group(): возвращает подстроку, которая совпала с шаблоном в результате вызова метода find. Если совпадение отсутствует, то метод генерирует исключение IllegalStateException.
* int start(): возвращает индекс текущего совпадения
* int end(): возвращает индекс следующего совпадения после текущего
* String replaceAll(String str): заменяет все найденные совпадения подстрокой str и возвращает измененную строку с учетом замен

Источник. <https://metanit.com/java/tutorial/7.4.php>

Вопрос 42.

Что появится в консоли в результате работы фрагмента программы?

System.out.println{"-1234".matches("-?\\d+"));

System. out. println ("5678". matches (" - ? \ \d+"));

System.out.println("+9ll".matches("-?\\d+"));

System.out.println("+911".matches("(-|\\+)?\\d+"));

Поясните ответ.

Ответ.

true

false

false

true

Программа проверила соответствие значения в ковычках с шаблоном с помощью метода .matches().

Источник.

Вопрос 43.

Что появится в консоли в результате работы фрагмента программы?

System.out.println(

Arrays

.toString("Then, when you have found the gold."

.split("n\\W+")

)

);

Поясните ответ.

Ответ. [The, whe, you have found the gold.]

Программа в качестве разделителя слов с помощью метода .split() выбрала "n\\W+".

Источник.

Вопрос 44.

Сформируйте и протестируйте регулярное выражение, которое проверяет, что предложение начинается с прописной буквы и завершается точкой.

Ответ.

Источник.

Вопрос 45.

Сформируйте и протестируйте регулярное выражение, которое заменяет все гласные подчеркиваниями в предложении

"Then, when you have found the gold".

Ответ.

Источник.

Вопрос 46.

Определите, будет ли найдено в строке

"Java now has regular expressions"

совпадение для следующих выражений:

^Java

\Breg.\*

n.w\s+h(a|i)s

s?

s\*

s+

S{4}

S{1}

S{0,3}

Поясните ответ.

Ответ.

Источник.

Вопрос 47.

Примените регулярное выражение

(?i)((^[aeiou])|(\s+[aeiou]))\w+?[aeiou]\b

к строке

"Arline ate eight apples and one orange while Anita hadn't any"

Поясните ответ.

Ответ.

Источник.

Вопрос 48.

Какие подстроки извлекают следующие регулярные выражения?

[0-6]

[^n-p]

[A-Za-z0-9\_]

\w

[A-C][n-p][a-c]

waz{3,5}up

[abc]+

.\*

Ответ.

Источник.

Вопрос 49.

Сформируйте регулярное выражение, которое находит предложения, начинающиеся с "Input:" и заканчивающиеся на "successful".

Ответ.

Источник.

Вопрос 50

Сформируйте регулярное выражение, которое выделяет в обрабатываемом тексте названия файлов (имя + расширение) графических форматов gif, png, jpg.

Ответ.

Источник.