**Java Strings**

1. В чем разница между созданием строки как new String() и литералом (при помощи двойных кавычек)?

Когда мы создаем строку используя new(), она создается в хипе и также добавляется в пул строк, в то же время строка, созданная при помощи литерала, создается только в пуле строк.



1. Как реализуется класс String, какие поля там есть?
2. Класс реализует интерфейсы Serializable и CharSequence. Поскольку он входит в пакет java.lang, его не нужно импортировать.
3. Класс **String** в Java — это **final** класс, который не может иметь потомков.
4. Класс String — immutable класс, то есть его объекты не могут быть изменены после создания. Любые операции над объектом String, результатом которых должен быть объект класса String, приведут к созданию нового объекта.
5. Благодаря своей неизменности, объекты класса String являются потокобезопасными и могут быть использованы в многопоточной среде.
6. Каждый объект в Java может быть преобразован в строку через метод toString, унаследованный всеми Java-классами от класса Object.

Класс java.lang.String состоит из трех final полей: массив символов, длина строки и сдвиг в массиве, с помощью которого реализуется метод substring. Так же в этом классе есть поле, в котором хранится посчитанный хэшкод, после первого вызова метода hashCode.

1. Как работает метод substring() класса String?

Метод substring класса String используется для получения части какой-то строки. Результатом «вырезания» строки методом substring будет новая строка. Сам метод перегружен и представлен в двух вариантах.

**substring(int beginIndex)** — возвращает новую строку, которая является подстрокой данной строки. Подстрока начинается с символа в указанной позиции индекса и продолжается до конца этой строки.

**substring(int beginIndex, int endIndex)** — возвращает новую строку, которая является подстрокой данной строки. Подстрока начинается с указанного **beginIndex** и включает символ в индексе **endIndex — 1**. Таким образом, длина подстроки **endIndex — beginIndex**.

String siteName = "prologistic.com.ua";

        System.out.println("Последние 7 символов строки siteName: " + siteName.substring(siteName.length() - 7));

        System.out.println("Первые 3 символа строки siteName: " + siteName.substring(0, 3));

        System.out.println("Site name: " + siteName.substring(0, 11));

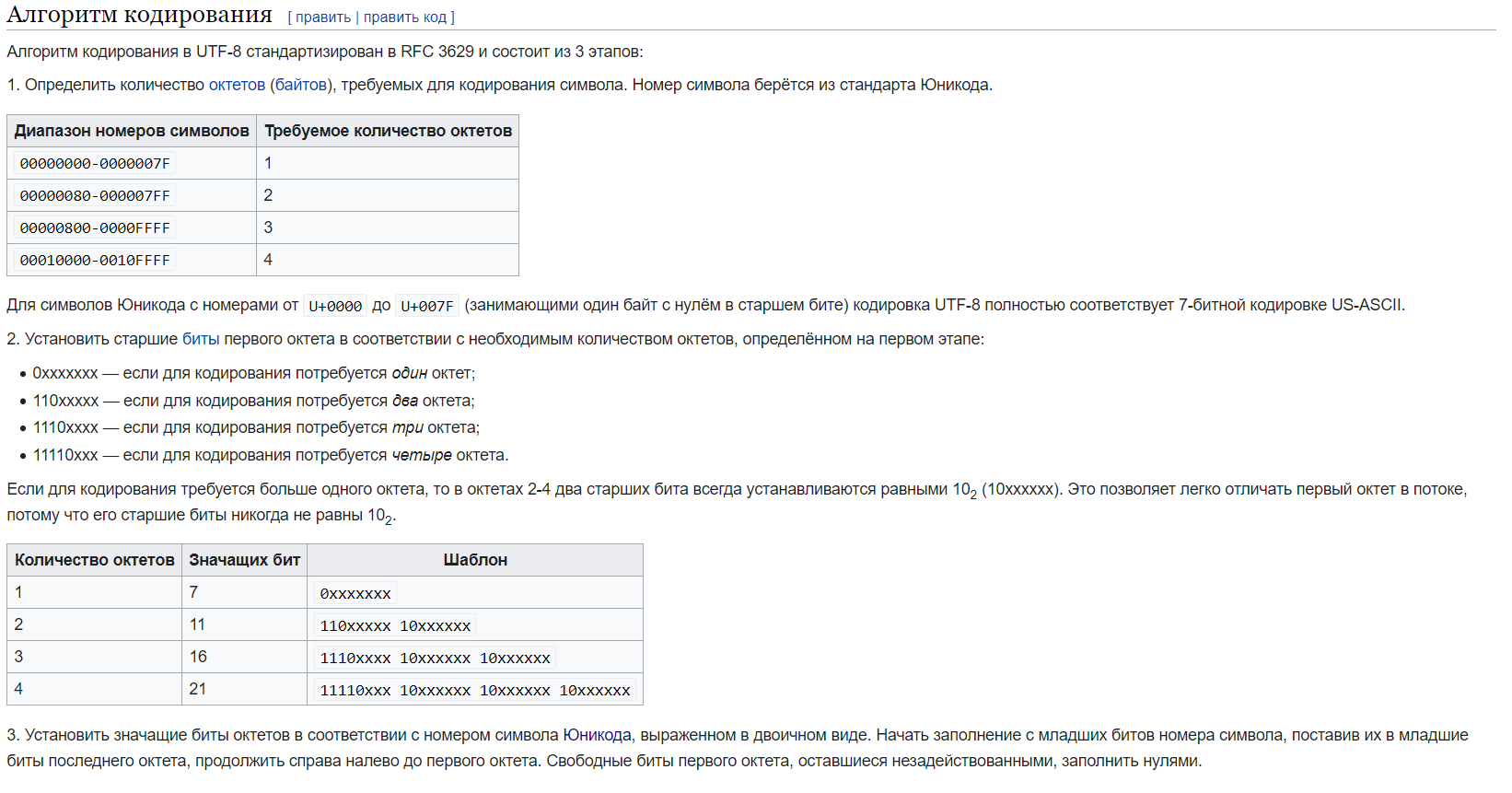
1. Понятие Юникод. UTF-8, описание кодировки. Отличие от UTF-16.

**Юникод**— стандарт кодирования символов, включающий в себя знаки почти всех письменных языков мира.

**UTF**-**8** – стандарт кодирования, преобразующий номера ячеек таблицы Юникод в бинарные коды с использованием переменного количества бит: **8**, **16**, 24 или 32.

**UTF**-**16** – стандарт кодирования, преобразующий номера ячеек таблицы Юникод в бинарные коды с использованием переменного количества бит:**16** или 32.

В зависимости от языка алфавита может быть выбрана как кодировка UTF-8, так и кодировка UTF-16. Для английского алфавита однозначно более выгодно использование кодировки UTF-8, для русского алфавита буквы представляются одинаковым количеством бит при использовании как одной, так и другой кодировки.



1. String, StringBuilder, StringBuffer. Отличия.

