1. Как создать объект класса String, какие конструкторы класса String вы знаете? Что такое строковый литерал? Объясните, что значит “упрощенное создание объекта String”?

String() String(String original) String(char[] value) String(char[] value, int offset, int count)

String(int[] codePoints, int offset, int count) String(byte[] ascii, int hibyte, int offset, int count)

String(byte[] ascii, int hibyte) String(byte[] bytes, int offset, int length, String charsetName)

String(byte[] bytes, int offset, int length, Charset charset) String(byte[] bytes, String charsetName)

String(byte[] bytes, Charset charset) String(byte[] bytes, int offset, int length) String(byte[] bytes)

String(StringBuffer buffer) String(StringBuilder builder) String(char[] value, int off, int len, Void sig) String(AbstractStringBuilder asb, Void sig) String(byte[] value, byte coder)

2. Можно ли изменить состояние объекта типа String? Что происходит при попытке изменения состояния объекта типа String? Можно ли наследоваться от класса String? Как вы думаете, почему строковые объекты immutable?

Да

Immutable = String Pool, Security, Synchronization, Hashcode Caching

3. Объясните, что такое кодировка? Какие кодировки вы знаете? Как создать строки в различной кодировке?

Unicode: UTF-8, UTF-16,UTF-32.

i18n - Интернационализация

L10n - Локализация

Кодирование — это перевод информации в удобную для передачи, обработки или хранения форму с помощью некоторого кода

new String(data, "UTF-8");

4. Что такое пул литералов? Как строки заносятся в пул литералов? Как занести строку в пул литералов и как получить ссылку на строку, хранящуюся в пуле литералов? Где хранится(в каком типе памяти) пул литералов в Java 1.6 и Java 1.7?

Благодаря неизменности строк в Java, JVM может оптимизировать объем выделяемой для них памяти, сохраняя только одну копию каждой литеральной строки в пуле. Этот процесс называется интернированием. Запихивается с помощью String s = “da”; Пул хранится в куче.

Когда мы создаем объект String с помощью оператора new (), он всегда создает новый объект в динамической памяти. С другой стороны, если мы создадим объект, используя синтаксис строкового литерала, он может вернуть существующий объект из пула строк, если он уже существует. В противном случае он создаст новый объект String и поместит его в пул строк для повторного использования в будущем.

Вручную можно запихнуть с помощью .intern();

5. В чем отличие объектов классов StringBuilder и StringBuffer от объектов класса String? Какой из этих классов

потокобезопасный? Как необходимо сравнивать на равенство объекты классов StringBuilder и StringBuffer и почему?

Отличие в том что String неизменяемый, а билдер и баффер изменяемы. Stringbuffer потокобезопасный.

Сравнивать через sb1.toString().equals(sb2.toString()) потому что у них не переопределены equals и по дэфолту сравниваются ссылки на объекты.

6. Что такое Unicode?

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

7. Какие методы класса String используются для работы с кодовыми точками? Как вы думаете, когда следует их

использовать?

codePointAt

codePointBefore

codePointAfter

1. Расскажите, что представляет собой регулярное выражение? Что такое метасимволы регулярного выражения? Какие вы знаете классы символов регулярных выражений? Что такое квантификаторы? Какие логические операторы регулярных выражений вы знаете? Что значит “якорь” для регулярного выражения?

Регулярное выражение (RegEx) – это шаблон для поиска строки в тексте.

Метасимволы – символы, имеющие специальное значение в синтаксисе регулярных выражений.

Метасимволы для поиска совпадений границ строк или текста

^ начало строки

$ конец строки

\b граница слова

\B не граница слова

\A начало ввода

\G конец предыдущего совпадения

\Z конец ввода

\z конец ввода

Метасимволы для поиска символьных классов

\d цифровой символ

\D нецифровой символ

\s символ пробела

\S непробельный символ

\w буквенно-цифровой символ или знак подчёркивания

\W любой символ, кроме буквенного, цифрового или знака подчёркивания

. любой символ

Метасимволы для поиска символов редактирования текста

\t символ табуляции

\n символ новой строки

\r символ возврата каретки

\f переход на новую страницу

\u 0085 символ следующей строки

\u 2028 символ разделения строк

\u 2029 символ разделения абзацев

Метасимволы для группировки символов

[абв] любой из перечисленных (а,б, или в)

[^абв] любой, кроме перечисленных (не а,б, в)

[a-zA-Z] слияние диапазонов (латинские символы от a до z без учета регистра )

[a-d[m-p]] объединение символов (от a до d и от m до p)

[a-z&&[def]] пересечение символов (символы d,e,f)

[a-z&&[^bc]] вычитание символов (символы a, d-z)

Квантификаторы — метасимволы для обозначения количества символов. Квантификатор всегда следует после символа или группы символов.

? один или отсутствует

\* ноль или более раз

+ один или более раз

{n} n раз

{n,} n раз и более

{n,m} не менее n раз и не более m раз

Java Regular expressions supports 3 logical operators they are −

XY: X followed by Y

X|Y: X or Y

(X): capturing group.

Якорь – определяет позицию шаблона в строке текста (^ – начало строки; $ – конец строки - в том случае, если они являются самым первым и самым последним символами шаблона. Иначе эти символы становятся литералами (представляют сами себя).

2. Какие java-классы работают с регулярными выражениями? В каком пакете они расположены? Приведите пример анализа текста с помощью регулярного выражения и поясните код примера.

Пакет java.util.regex исходно состоит из следующих трех классов:

Pattern Class – объект класса Pattern представляет скомпилированное представление регулярного выражения. В классе Pattern публичный конструктор не предусмотрен. Для создания шаблона, вам сперва необходимо вызвать один из представленных публичных статичных методов compile(), который далее произведет возврат объекта класса Pattern. Регулярное выражение в данных методах принимается как первый аргумент.

Matcher Class – объект класса Matcher представляет механизм, который интерпретирует шаблон, а также производит операции сопоставления с вводимой строкой. Аналогично классу Pattern, Matcher не содержит публичных конструкторов. Объект класса Matcher может быть получен путем вызова метода matcher() на объекте класса Pattern.

PatternSyntaxException – объект класса PatternSyntaxException представляет непроверяемое исключение, которое обозначает синтаксическую ошибку в шаблоне регулярного выражения.

3. Что такое группы в регулярных выражениях? Как нумеруются группы? Что представляет собой группа номер 0(ноль)? Приведите пример с использованием групп регулярного выражения.

Группа - это пронумерованная часть регулярного выражения. Например, в следующем выражении (\d+)zzz присутствуют две группы.

Группы сбора нумеруются посредством определения числа открывающих круглых скобок слева направо. Так, в выражении ((A)(B(C))) присутствуют четыре подобные группы:

((A)(B(C)))

(A)

(B(C))

(C)

Также имеется специальная группа, группа 0, которая во всех случаях представляет выражение в полном виде.