**Strings**

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

* Самый простой способ создания строки (присвоив ему значение строкового литерала) выглядит так:

String greeting = "ввв ввв ввввв!";

В данном случае «ввв ввв ввввв» — это строковый литерал (т.е. константа) состоящая из последовательности символов, заключенных в кавычки. Всякий раз, когда компилятор сталкивается со строковым литералом, он создает объект типа String со значением. Как и любой другой объект, строку можно создать, используя ключевое слово new.

Можно создать массив строк:

String[] cats = {"Васька", "Барсик", "Мурзик"};

Можно создать пустой объект класса String:

String str = new String();

Можно создать строку через массив символов:

сhar[] chars = { 'c', 'a', 't' };

String str = new String(chars);

**упрощенное создание объекта String?????**

1. **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??**

* 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.

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

Благодаря своей неизменности, объекты класса String являются потокобезопасными и могут быть использованы в многопоточной среде. Если строка будет изменяемой, тогда это станет серьезной угрозой безопасности приложения.

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

* Кодировкой называют способ представления в памяти компьютера цифр, буков и всех остальных знаков. Каждой букве или каждому символу соответствует какое-то число. Кодировкой называется набор символов и соответствующий им набор кодов. Именно поэтому тип char считается не только символьным, но и числовым типом.

Существуют кодировки UTF-8, UTF-16, Windows-1251, KOI8-R,… и UNIcode.

На основе строки можно создать массив байт в любой известной Java кодировке:

String s = "Good news everyone!";

byte[] buffer = s.getBytes("Windows-1251");

или

String s = "Good news everyone!";

Charset koi8 = Charset.forName("KOI8-R");

byte[] buffer = s.getBytes(koi8);

Преобразовать набор байт в строку:

byte[] buffer = new byte[1000];

inputStream.read(buffer);

String s = new String(buffer, "Windows-1251");

1. **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?**

* Пул литералов – это набор строк, который хранится в памяти heap (Java1.7) и perm (Java 1.6). Пул литералов\строк возможен исключительно благодаря неизменяемости строк в Java. Когда мы используем двойные кавычки для создания строки, сначала ищется строка в пуле с таким же значением, если находится, то просто возвращается ссылка, иначе создается новая строка в пуле, а затем возвращается ссылка. Пул строк помогает экономить большой объем памяти, но с другой стороны создание строки занимает больше времени.

1. **В чем отличие объектов классов StringBuilder и StringBuffer от объектов класса String? Какой из этих классов потокобезопасный? Как необходимо сравнивать на равенство бъекты классов StringBuilder и StringBuffer и почему?**

* StringBuilder и StringBuffer – это аналог класса String, только изменяемый. Разница в том, что StringBuffer потокобезопасен, и все его методы синхронизированы, а StringBuilder — нет. Это единственная особенность. StringBuilder в Java работает быстрее StringBuffer’а благодаря несинхронизированности методов.

String str = new String("карамба");  
StringBuilder str1 = new StringBuilder(str);  
StringBuilder str2 = new StringBuilder(str);  
String t1 = str1.toString();  
String t2 = str2.toString();  
  
System.*out*.println(t1.equals(t2));

1. **Что такое Unicode?**

* Это стандарт, а не кодировка — сам по себе Юникод не определяет, как символы будут сохранятся на жестком диске или передаваться по сети. Он лишь определяет связь между символом и некоторым числом, а формат, согласно с которым эти числа будут превращаться в байты, определяется Юникод-кодировками (например, UTF-8 или UTF-16).

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

* codePointCount - возвращает количество кодовых единиц для данной строки в кодировке UTF-16.

1. int cpCount = str.codePointCount(0, str.length());

* codePointAt() - возвращает кодовую единицу в позиции n, где n находится в интервале от 0 до s.length() – 1,
* int cp = str.codePointAt(3);

**Regular Expressions**

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 (regular expression/regex/regexp) — это строка, которая является шаблоном (pattern), описывающим некий набор строк. Шаблон определяет, какие именно строки относятся к набору. Шаблон состоит из литералов и метасимволов — символов со специальным, а не буквальным значением. Поиск по шаблону –— это поиск по тексту для нахождения совпадений, то есть строк, соответствующих шаблону регулярного выражения.