

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

Создать строку можно с помощью оператора new, различных конструкторов, которые могут принимать массив байтов, символов и т.д., или просто присвоив переменной строку-значение. Строковый литерал — это набор символов, заключённый в кавычки. Упрощённое создание объекта — присваивание переменной строки в двойных кавычках.

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?**

Объект типа String неизменяемый. Каждый раз при изменении строки создаётся новый объект. Класс String объявлен как final, поэтому от него нельзя наследоваться. Строковые объекты неизменяемые, потому что строки хранятся в пуле литералов. Таким образом строки используются сразу несколькими клиентами. И если один из них захочет изменить строку, то все остальные также столкнутся с этими изменениями.

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

Кодировка — это таблица, задающая взаимно-однозначное соответствие между символами и их кодами. Наиболее распространённые кодировки: UTF-8, ISO-8859-1, Windows-1251, ASCII. Объекты String в Java имеют кодировку UTF-16, которая не изменяется. Только массив byte[] может иметь различные кодировки.

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?**

Пул литералов — это набор литералов, который хранится в памяти Java heap. Строка заносится в пул при создании нового объекта с помощью оператора new. При вызове метода intern() возвращается ссылка на строку из пула, если такая строка уже существует. Пул литералов в Java 7 хранится в куче (heap). В Java 6 пулы литералов хранились в PermGen — Permanent Generation – область памяти в JVM, предназначенная для хранения описания классов Java и некоторых дополнительных данных. Её минус заключался в том, что она имела ограниченный размер.

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?

Класс `String` является неизменяемым, в то время как объекты классов `StringBuilder` и `StringBuffer` можно изменять. `StringBuffer` — потокобезопасный. Метод `equals()` для `StringBuilder` сравнивает на равенство объекты. Для сравнения строк можно использовать `sb1.toString().equals(sb2.toString())` или, что более эффективно, сначала сравнивать длину строк на равенство при помощи метода `length()`, а затем сравнивать строки посимвольно при помощи `charAt(i)`. Однако начиная с Java 11 `StringBuilder` реализует интерфейс `Comparable`, поэтому сравнить строки на равенство можно с помощью `sb1.compareTo(sb2) = 0`.

6. Что такое Unicode?

Unicode — это стандарт кодирования символов. Он состоит из универсального набора символов и семейства кодировок UTF.

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

Кодовые единицы типа `char` представляют кодовые точки Unicode в кодировке UTF-16. Методы класса `String` для работы: `codePointCount(int beginIndex, int endIndex)`, `offsetByCodePoint(0, i)`, `codePointAt(i)`.