Вопросы по теме Класс String:

1. Какая разница между String, StringBuilder, StringBuffer?
2. Что такое пул стрингов (строк)?
3. Как отсортировать список строк в алфавитном порядке?
4. В какой кодировке хранятся строки в Java?
5. Почему char[] лучше для хранение пароля чем String?
6. Можно ли использовать строки в switch?
7. Что такое immutable объекты?
8. Как создать immutable класс?
9. Почему String immutable и final?
10. Почему String хороший ключ для HashMap?

Класс String.

Класс String предназначен для работы со строками. Все строковые литералы в программах на Java, такие как "abc", реализованы как экземпляры этого класса. Особенности класса String:

1. все операции по изменении строки возвращают её новый экземпляр, в том числе и конкатенация строк ("abc" + "abc");
2. объявлен как final (не может иметь потомков);
3. поддерживает интерфейсы Serializable, Comparable<String>, CharSequence;
4. является immutable (неизменяемым) классом;
5. каждый объект в Java может быть преобразован в строку через метод toString();
6. строки хранятся в кодировке UTF-16;
7. кэширует свой hashcode (поэтому тип данных String считается хорошим ключем для HashMap, т.к. позволяет не тратить время на вычисление хэш-кода);
8. метод equals() и методы поиска (например indexOf()) оптимизируются JIT компилятором на нативном уровне;
9. компилятор умеет оптимизировать конкатенацию строк (до Java 9 это был синтаксический сахар над созданием объекта класса StringBuilder, несколькими вызовами append() и затем вызовом toString(), после Java 9 вместо StringBuilder генерируется bytecode для вызова класса StringConcatFactory);
10. начиная с Java 7 объекты класса String можно использовать в операторе switch.

Пул строк (String Pool) – набор объектов String, хранящихся в динамической памяти (Memory Heap) с целью экономия этой памяти. При создании нового объекта String сначала идет проверка, если в Memory Heap уже есть объект с идентичным строковым значением, то новый объект не создается, а переменной присваивается ссылка на существующий объект. Исключение – создание объекта с помощью оператора new.

Все строковые значения после создания хранятся в пуле строк, пока сборщик мусора не очистит объекты, на которые отсутствуют ссылки. Для хранения пароля данный механизм является не безопасным, т.к. получив доступ к пулу строк можно найти сохраненный там пароль. Для избегания подобного рекомендуется пароль хранить в массиве char[] (который после использования можно сразу же очистит), либо можно использовать специализированные библиотеки (например, JPasswordField).

Класс StringBuilder и StringBuffer – изменяемые классы для работы со строками (позволяют изменять содержание сроки, а не создают новый объект, как это делает класс String). StringBuffer потокобезопасен, а StringBuilder нет. Оба эти классы являются реализацией абстрактного класса AbstractStringBuilder. Внутри хранится массив char[], который расширяется по правилу: value.length * 2 + 2. По умолчанию размер (capacity) у StringBuilder равен 16.

Immutable объект.

Объект считается неизменяемым (immutable), если его состояние не может измениться после его создания. Максимальное использование неизменяемых объектов широко признано надежной стратегией создания простого и надежного кода. Неизменяемые объекты особенно полезны в параллельных приложениях. Поскольку они не могут изменить состояние, они не могут быть повреждены вмешательством потока или обнаружены в несогласованном состоянии.

Создание неизменяемого класса:

1. отсутствие методов установки состояния (нету сеттеров);
2. все переменные private final;
3. сам класс final;
4. если переменные изменяются внутри класса, это изменение должно быть не видно и не иметь никакого эффекта за пределами класса (включая влияние на такие вещи, как equals()и hashcode());
5. никогда не возвращать какое-либо изменяемое поле, всегда возвращать либо копию, либо неизменяемую версию поля.

Класс String является неизменяемым классом. Причины:

1. для обеспечения безопасности приложения (например, если строка используется для передачи параметров (параметры подключения к базе данных или параметры сетевого соединения) и если она будет изменяема, то данные можно будет легко повредить);
2. для поддержки механизма загрузки классов, в которых объекты String используются как аргументы (изменяемость может привести к загрузке неправильного класса);
3. для безопасности работы в многопоточной среде;
4. неизменяемость позволяет использовать механизм String Pool (пул строк).