

9.6

String Pool

00:00–00:20

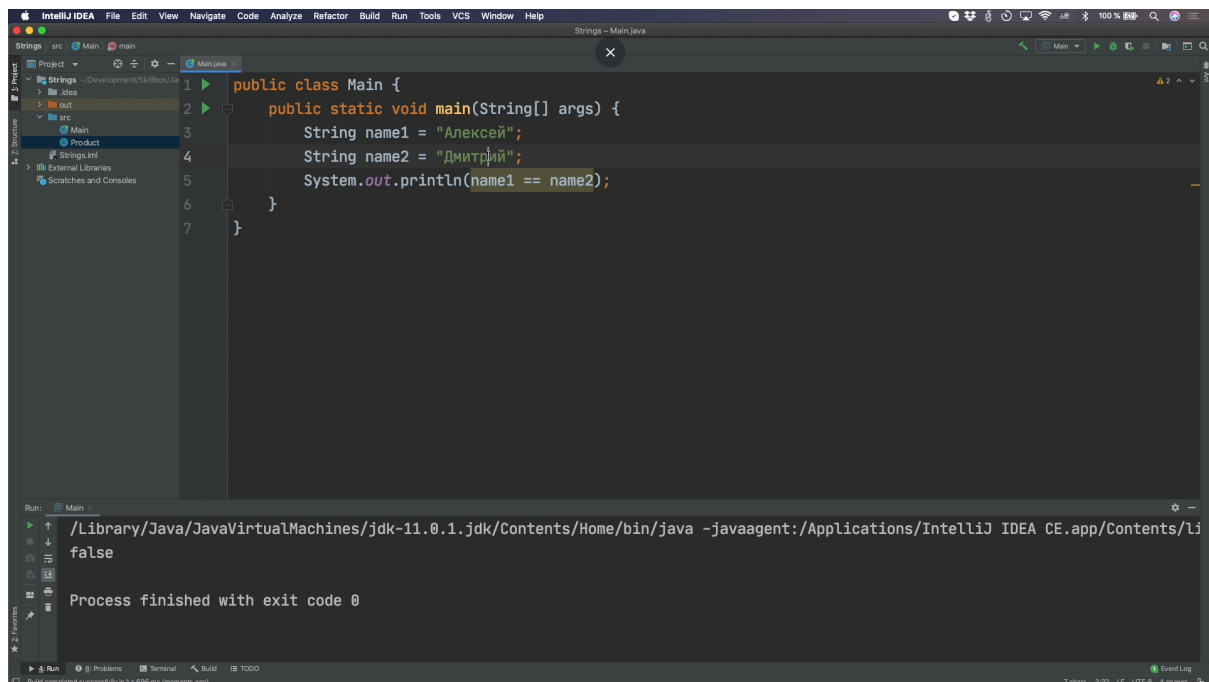
Интро

Привет! В этом материале мы поговорим о так называемом пуле строк, который оптимизирует расход памяти на одинаковые строки и в который строки могут попадать, могут не попадать и могут быть явным образом отправлены.

00:20–03:28

String Pool

Вы уже помните, что при сравнении объектов с помощью двойного равенства происходит сравнение ссылок на эти объекты, и при выполнении этого кода мы получим результат `false`, что вполне логично:



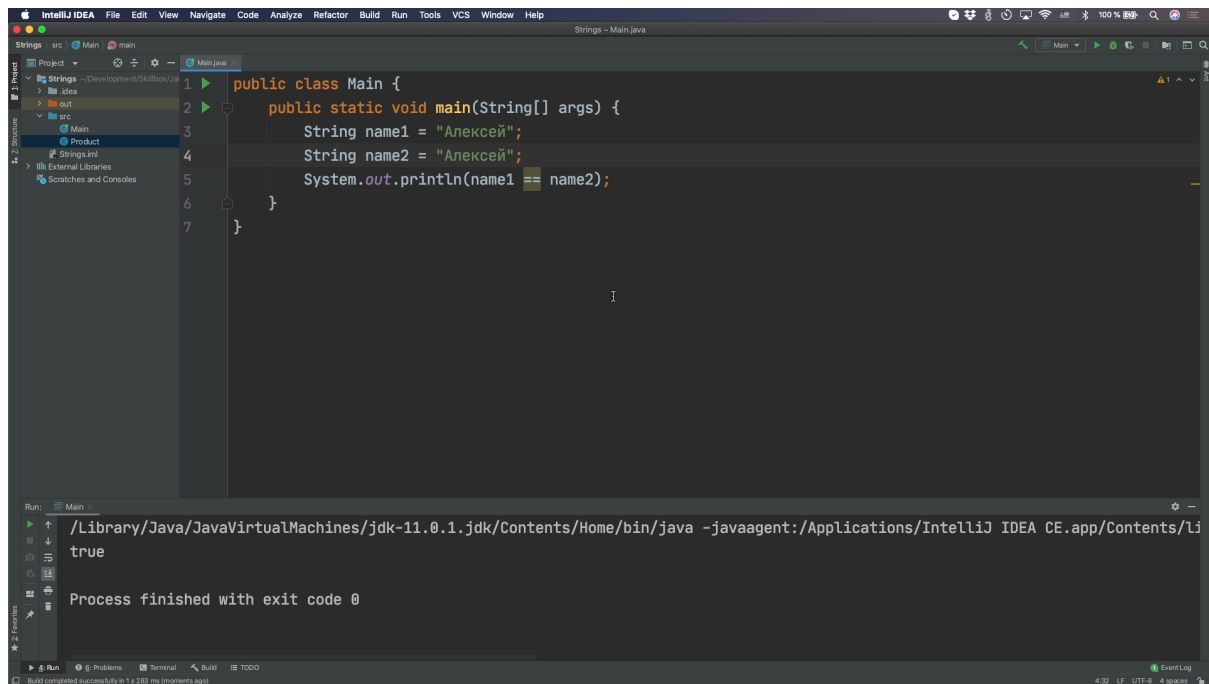
```
1 public class Main {
2     public static void main(String[] args) {
3         String name1 = "Алексей";
4         String name2 = "Дмитрий";
5         System.out.println(name1 == name2);
6     }
7 }
```

Run: Main

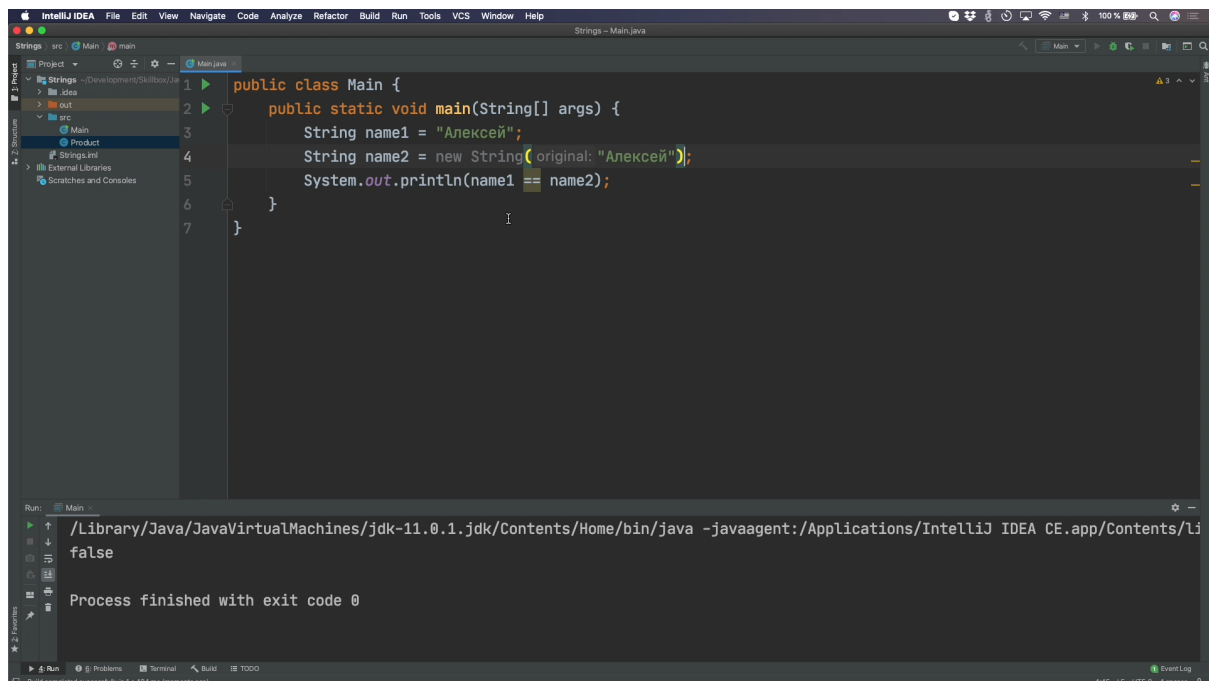
```
/Library/Java/JavaVirtualMachines/jdk-11.0.1.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/Li
false

Process finished with exit code 0
```

Но если здесь строки будут одинаковые, то при таком сравнении мы получим `true`, хотя, казалось бы, мы изначально создавали два разных объекта:



Если же мы здесь напишем следующим образом и это будут по-прежнему две одинаковые строки, то при выполнении кода у нас уже будет результат `false`, и это как раз логично — два разных объекта в памяти, две разные ссылки, естественно, они не равны:



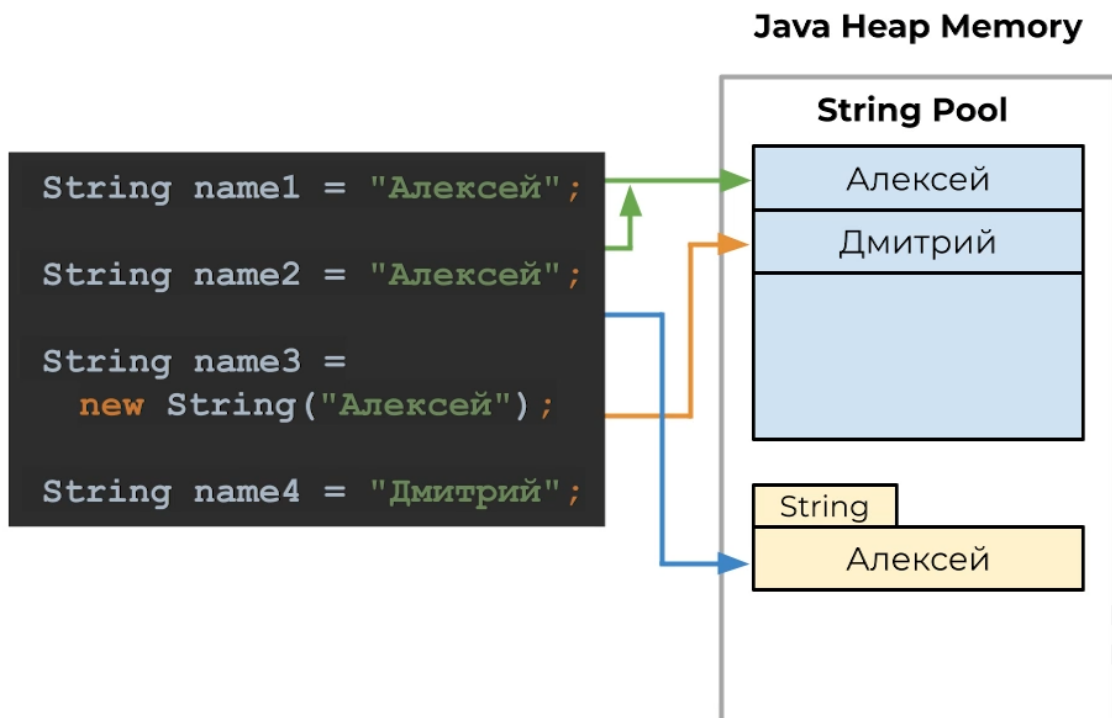
Так почему же в ситуации, когда у нас нет никакого `new String` и в коде два разных объекта, результат равен `true`?

Потому что в Java существует string pool, или пул строк, — это специальный пул объектов класса string, который позволяет хранить одинаковые строки в одних и тех же объектах для экономии памяти.

Обозначим этот string pool прямоугольником и посмотрим, как он работает. Он находится в памяти Java-машины, которую мы обозначим серым прямоугольником. При приравнивании переменной к строке происходит добавление этой строки в пул. Создавая вторую такую же строку, мы будем получать ссылку на тот же самый объект в пуле. Такое уникальное хранение строк в памяти называется **интернирование** (по-английски string interning).

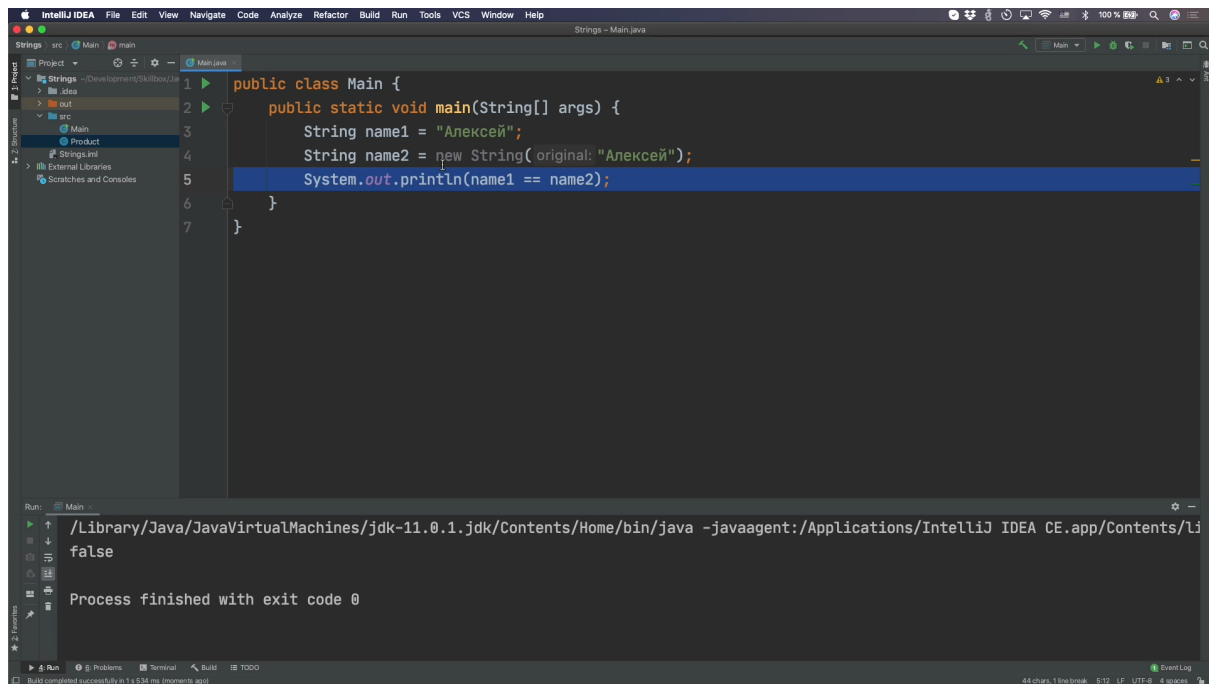
Если же мы создадим строку с помощью конструктора класса string, такая строка не попадёт в пул — она будет храниться отдельно, в виде отдельного объекта. Если мы создадим ещё какую-то строку, то она попадёт в пул.

String Pool

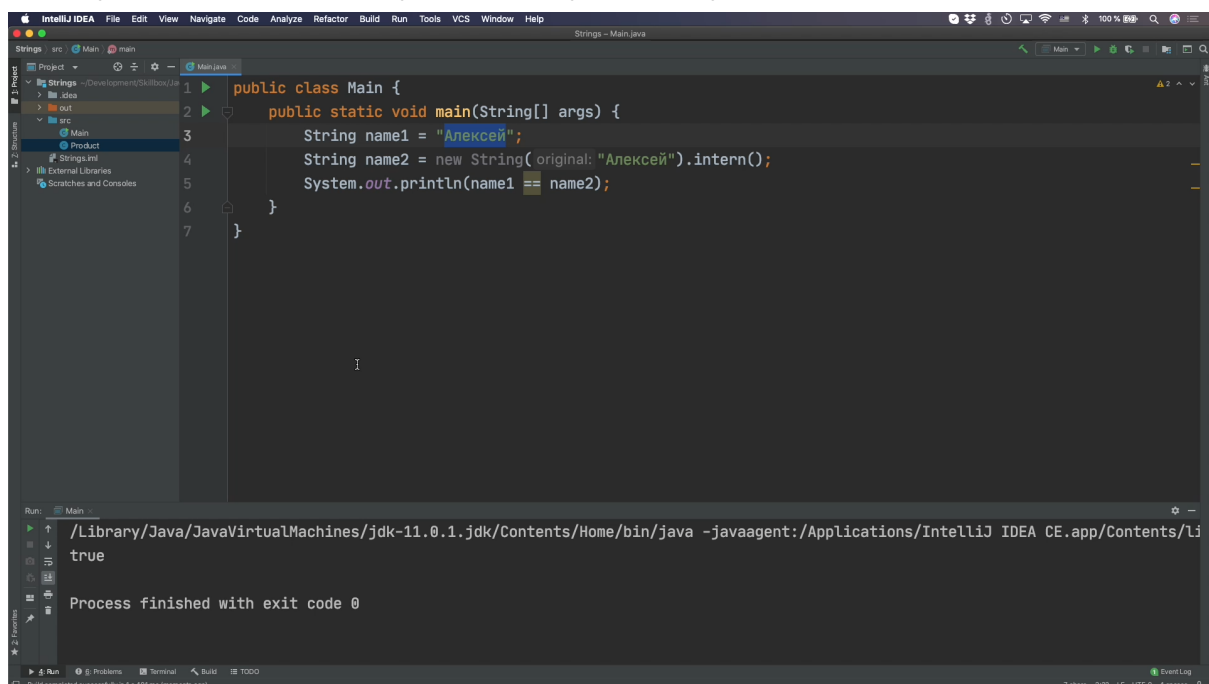


В Java есть возможность интерпретировать строку, которая была создана как отдельный объект, вызвав у неё метод `intern` — этот метод будет возвращать ссылку на эту строку в пуле строк.

Давайте посмотрим, как это работает:



Результат выполнения такого кода — false, поскольку строка string name1 входит в пул, а строка string name2 уже не входит. Но если использовать метод intern, то он будет возвращать ссылку строки вне пула на объект в пуле, и результат будет равен true.



Ещё раз: если мы у строки вызовем метод intern, он вернёт ссылку на такую же строку в пуле, если она там есть. Если её там нет — она сначала будет туда добавлена, а потом на неё будет возвращена ссылка.

03:38–03:58

Итоги

Итак, вы познакомились с так называемым string pool и узнали, в каких случаях строки туда не попадают. Разобрались, что такое интернирование строк, и познакомились с методом intern, который возвращает ссылку на такую же строку в пуле.