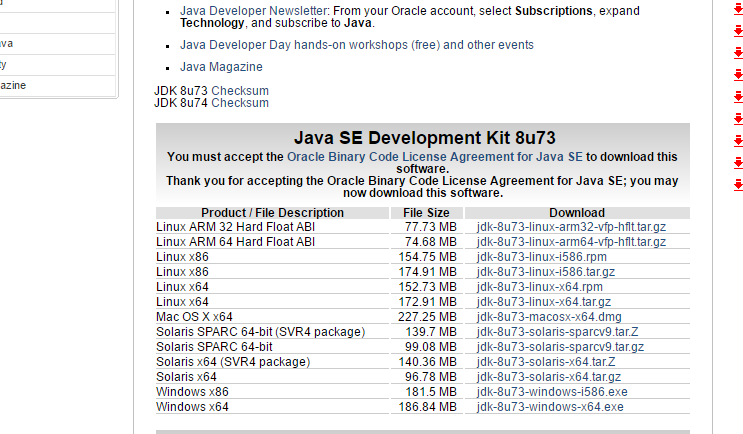
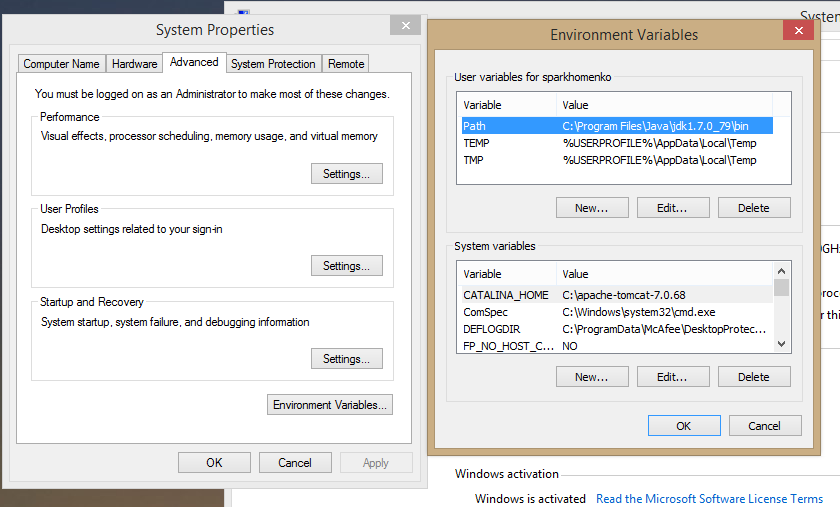
1. **Java Installation**

Перейдите на страницу с дистрибутивами JDK:  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>  
  
Выберите “Choose license agreement” и загрузите нужную версию Java:  
 - для 32 разрядной Windows это Windows x86  
 - для 64 разрядной Windows это Windows x64



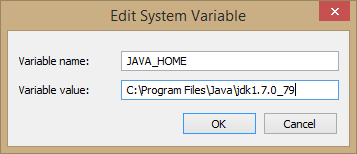
После загрузки запустите программу установки и следуйте инструкциям.

После установки необходимо настроить переменные окружения. Для этого перейдите в дополнительные параметры системы и во вкладке “Advanced” нажмите кнопку “Environment Variables”

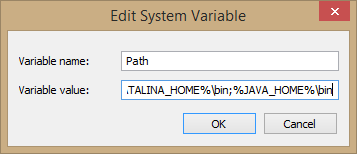


Вы можете добавлять переменные или в секцию переменных для пользователя (верхняя таблица) или в секцию системных переменных (нижняя таблица). В первом случае переменные применятся для текущего пользователя, который вошел в систему, во втором случае для всех пользователей.

Добавьте следующие переменные:  
- JAVA\_HOME, которая будет содержать путь к папке с установленной JDK  
- Path, которая будет содержать путь к файлам java.exe и javac.exe (они находятся в папке “bin” инсталляции JDK)



Обратите внимание на то, что путь к JDK у вас может отличаться от того, который указан на рисунке

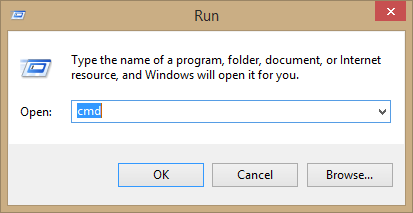


Переменная Path может содержать другие значения, в этом случае вы должны добавить новую переменную через запятую, как показано на рисунке.

Вместо %JAVA\_HOME% мы могли бы указать полный путь к папке “bin” как  
C:\Program Files\Java\jdk1.7.0\_79\bin, но так как часть этого пути мы задали в переменной JAVA\_HOME, то можно использовать эту переменную для более короткой записи.

После установки переменных запустите командную строку. Если она уже была запущена, обязательно закройте ее и запустите снова, так как переменные применятся только после перезапуска.

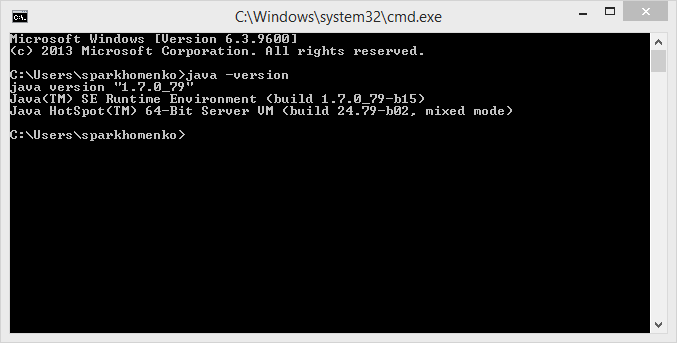
Запустить командную строку (консоль) можно открыв меню «Выполнить» и введя команду “cmd”.



Теперь проверим, что мы всё сделали правильно.

Введите команду:  
java -version

Вы должны увидеть информацию о версии Java:



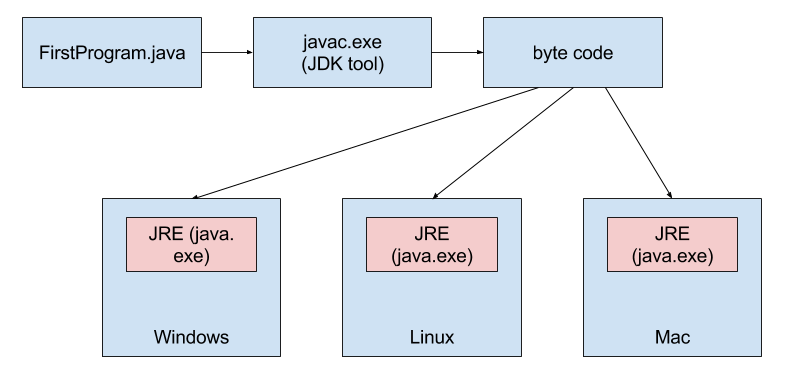
1. **Основная информация о среде исполнения**

Сначала рассмотрим основные термины и жизненный цикл программы (приложения).

JRE - Java Runtime Environment - среда для запуска байт кода, нужно конечному пользователю, который будет использовать нашу программу

JDK - Java Development Kit - содержит средства для разработки, используется разработчиками программ на Java, также содержит в себе JRE (то что мы скачали, установили и будем использовать)

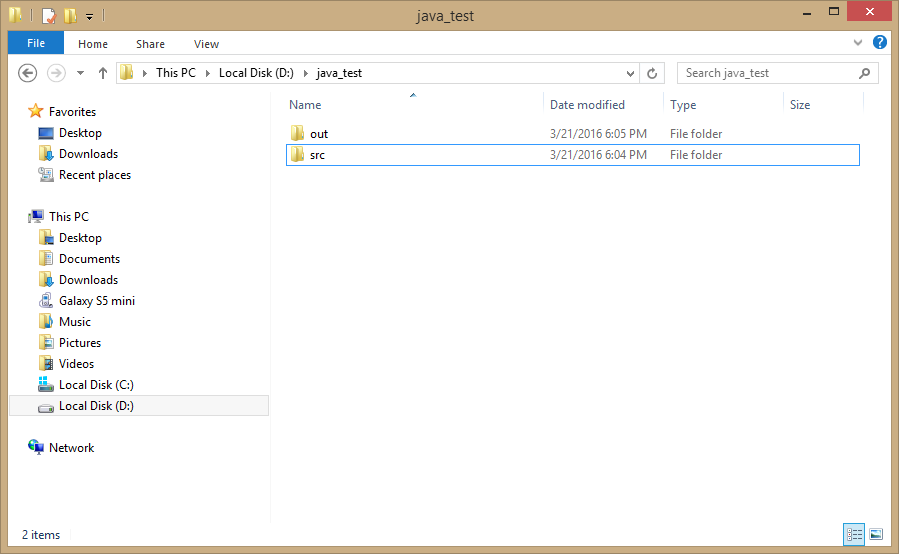
Процесс компиляции и сборки программы изображен на рисунке:



С помощью программы javac.exe происходит компиляция программы (программа написанная на java преобразовывается в байт-код). Затем байт код может быть запущен в любой операционной системе, на которой установлена JRE. Таким образом, нам не нужно писать разный код для каждой операционной системы, разной будет только установка JRE, которую можно скачать с сайта Oracle. По этой причине Java является мультиплатформенным языком программирования.

1. **Создание простой программы**

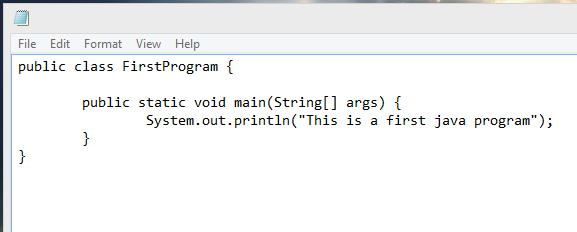
Создайте где-нибудь папку с любым именем, в которой создайте папки “src” и “out”:



В папке “src” cоздайте файл FirstProgram.java

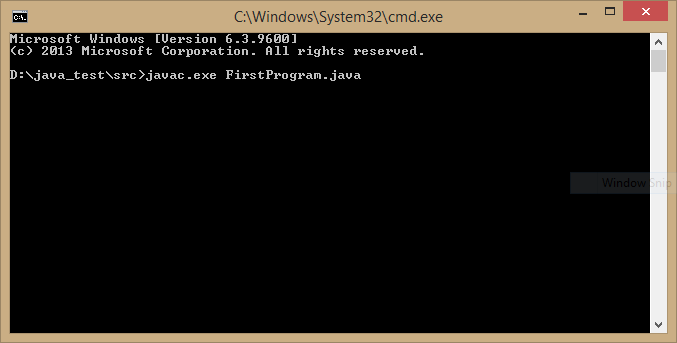
Возможно вам нужно будет включить опцию «Показывать расширения файлов» (в Windows по умолчанию эта опция выключена и несмотря на то, что вы добавили расширения “.java”, вы получите файл с расширением “.txt”)

Откройте файл любым текстовым редактором и добавьте следующий код:



Я не стал вставлять текст - напишите этот код сами, без копирования.

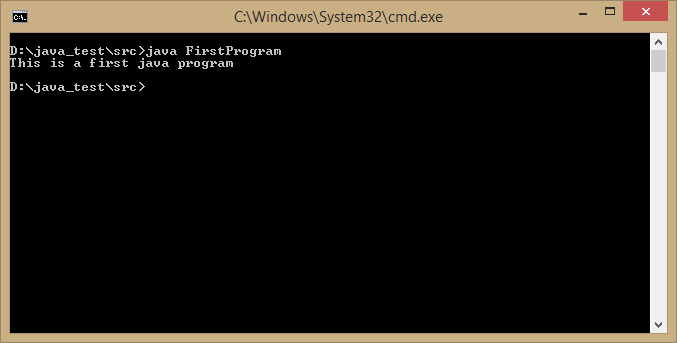
Откройте консоль в пределах папки “src” и запустите следующую команду:



Если вы не увидели сообщения об ошибке, всё прошло успешно и вы увидите, что в этой же папке у вас был создан файл “FirstProgram.class”.

В этом классе хранится байт-код, который мы может выполнить и увидеть результат выполнения нашей программы.

Выполните программу запустив следующую команду:  
java FirstProgram



Выражение “System.out.println(“This is a first java program”); вывело сообщение в консоль. Это единственное, что сейчас делает наша программа.

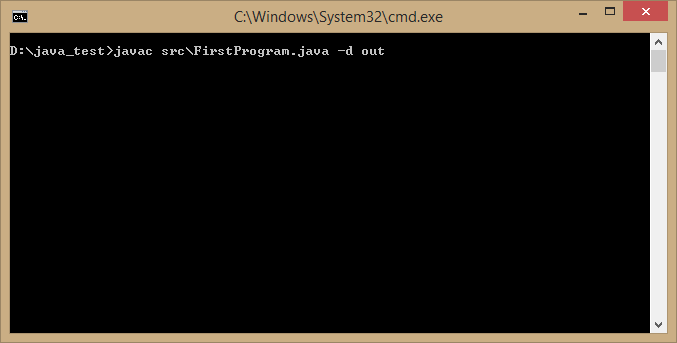
В параметре для программы “java” мы передаем только имя нашего класса “FirstProgram” (все файлы будут автоматически просмотрены этой программой и будет запущен тот класс, который мы указали) в отличие от “javac” (компиляция), где мы передавали имя файла.

Теперь сделаем так, чтобы “FirstProgram.class” оказался в папке out, для более логичного расположения файлов, мы не хотим смешивать код и результат компиляции (байт код). У конечного пользователя будет находится только байт-код, код на Java ему ни к чему.

Удалите файл “FirstProgram.class” из папки “src”.

Вернитесь на уровень выше и запустите консоль, если она уже запущена, можете напечатать:  
cd ..

Скомпилируйте программу:

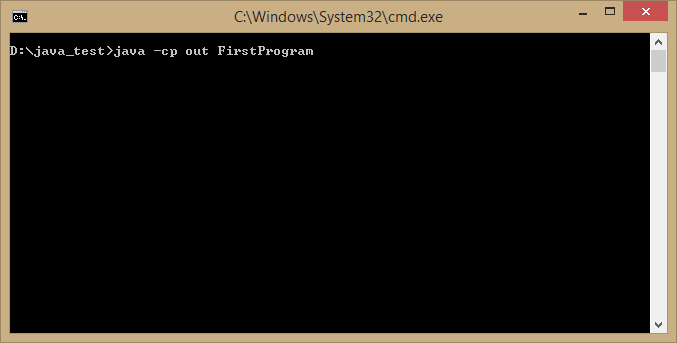


Здесь мы указали путь к файлу “FirstProgram.java” начиная с текущего уровня, с тем же результатом мы могли бы указать полный путь начиная от диска “D” (D:\java\_test\src\FirstProgram.java).

Обратите внимание на дополнительный параметр “-d”, после которого мы можем указать папку, в которую поместится результат компиляции (“FirstProgram.class” файл).

Если компиляция прошла успешно, можно приступить к запуску программы.

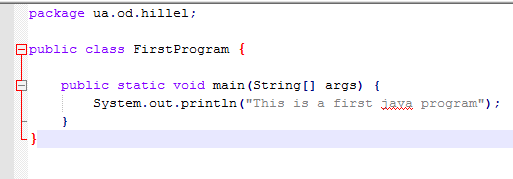
Выполните ту же команду как и в прошлый раз:  
java -cp out FirstProgram



Вспомните, что для команды “java.exe” мы указываем имя класса, но в текущей на данный момент папке (здесь “D:\java\_test”) нет нашего “FirstProgram.class” файла и мы должны как-то показать, где он находится. Для этого используется опция “-cp”, которая означает CLASSPATH. В нее мы передаем папку с классами.

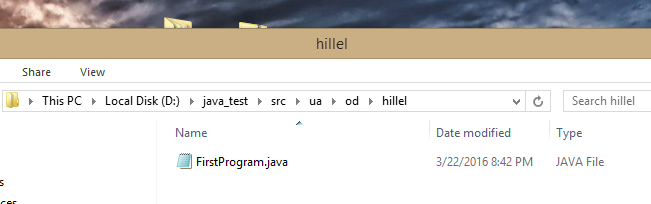
Для структурирования классов могут быть использованы пакеты - префикс перед именем класса.

Давайте добавим его в наш код:



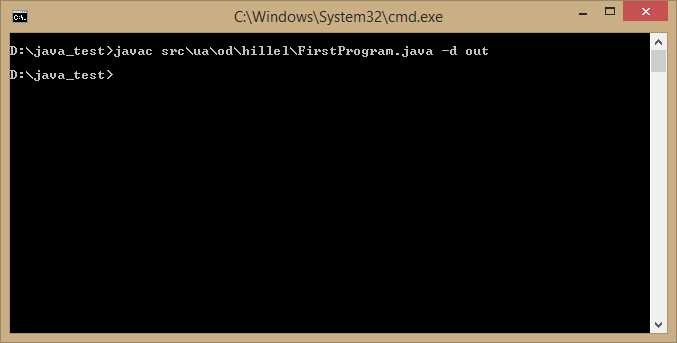
Пакет представляет собой доменное имя наоборот, и добавляет уникальности нашему классу.

Сохраните изменения и поместите ваш файл “FirstProgram.java” в папку соответствующую названию пакета:



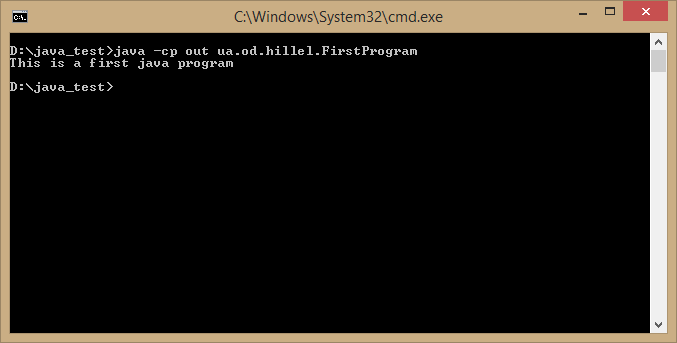
После этого вернитесь на уровень корневой папки (у меня “java\_test”).

Скомпилируйте программу:  
javac src\ua\od\hillel\FirstProgram.java -d out



В папке “out” у вас должна создасться такая же структура каталогов, какую вы создали в папке “src”. Если у вас остался файл “FirstProgram.class” в корне “out” удалите его.

Запустите программу:



Теперь наш класс называется “ua.od.hillel.FirstProgram”.

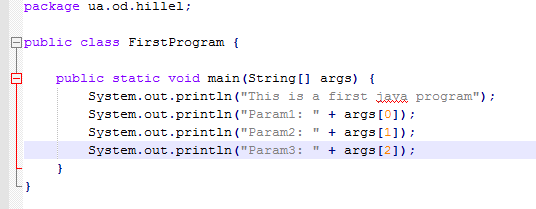
1. **Передача параметров из командной строки в программу**

Мы можем передавать параметры из консоли в нашу программу следующим образом:

java -cp out ua.od.hillel.FirstProgram str1 str2 str3

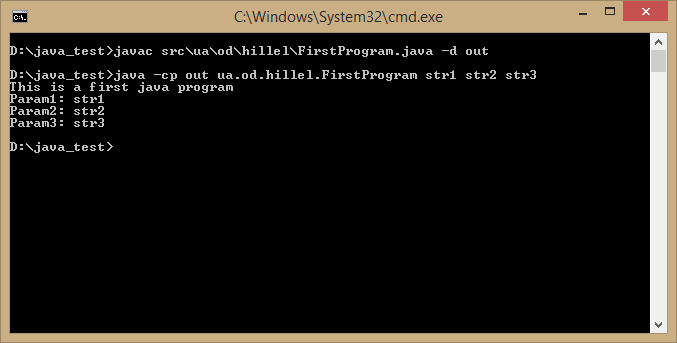
Как видите параметры передаются через пробел после строки запуска программы.

В коде мы можем их взять из переменной “args”:



В программе нумерация параметров начинается с нуля.

Скомпилируйте и запустите программу:



1. **Создание JAR-файла для классов**

Java JAR файлы являются архивом формата ZIP и вы можете открыть такой файл любым архиватором, который поддерживает этот формат.

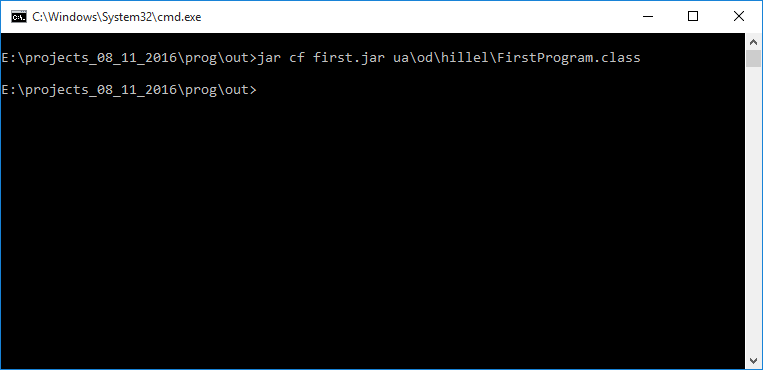
Для этих целей вы можете скачать и установить бесплатный архиватор 7-Zip, который легко позволяет менять файлы внутри самого архива, если понадобится такая необходимость:  
<http://www.7-zip.org/>

Давайте теперь запакуем класс FirstProgram в архив. Это выполняется с помощью утилиты “jar.exe”, которая находится в папке “bin” с установленной JDK, обязательно проверьте, что она там есть.

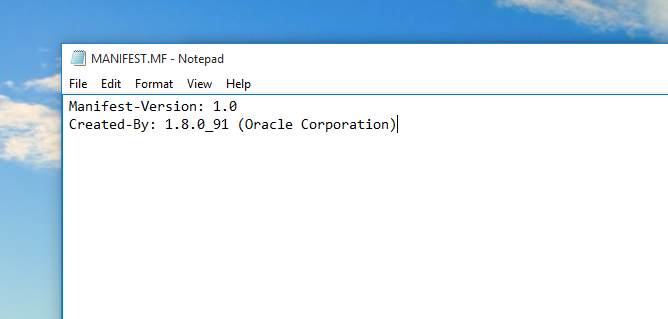
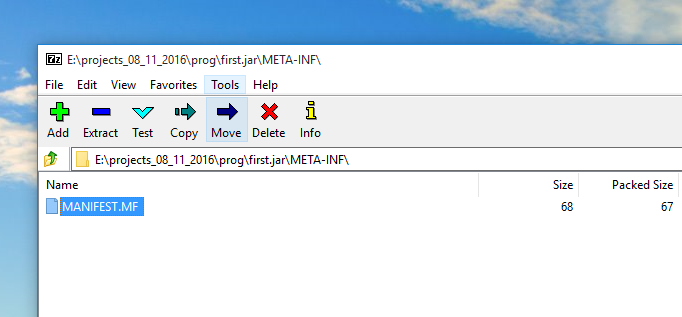
Дальше запустите команду, находясь в директории с папкой “out”:  
jar cf first.jar ua\od\hillel\FirstProgram.class

cf – укажет на то, что мы будет создавать файл .jar  
first.jar – имя архива (может быть любым, в зависимости от предпочтений)  
ua\od\hilllel\FirstProgram.class – путь к файлу с байткодом

Вот как это выглядит в консоли:



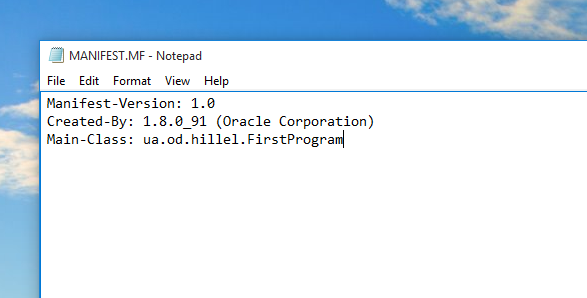
Кроме файлов .class, которые помещаются в архив, здесь еще есть файл манифеста, в котором хранится информация о версии файла-архива и версии jdk, в котором был создан этот файл:



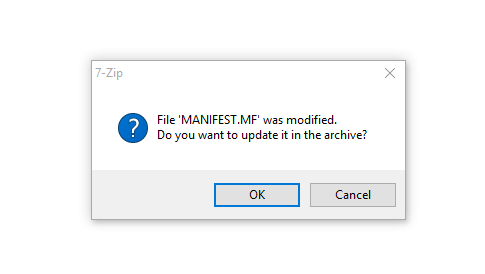
Для запуска нашей программы из консоли нужно отредактировать файл манифеста, указав в нем класс, в котором находится main-метод программы.

Если вы используете 7-Zip архиватор, вы можете щелкнуть правой кнопкой на файле MANIFEST.MF и выбрать опцию «Редактировать».

Допишите следующую строчку в MANIFEST.MF:  
Main-Class: ua.od.hillel.FirstProgram



Сохраните файл, 7-Zip предложит заменить его в архиве:



Нажмите «ОК» и вы автоматически получите обновленный архив.

Теперь откройте консоль в папке с вашим jar-файлом и выполните следующую команду:

java -jar first.jar str1 str2 str3

В вашей консоли вы должны увидеть следующий результат:

