

## Запуск программ без среды разработки

Собирать проекты из больше чем одного файла вручную — занятие довольно болезненное. Поэтому воспользуемся средой разработки Maven

### Установка

Предполагается, что вы уже установили java необходимой версии (17+), а также Git. Проверьте, что у вас переменная окружения JAVA\_HOME существует и указывает на папку установки Java (Например, у меня это C:\Program Files\Java\jdk-17). Если нет — добавьте её. Вот довольно подробный гайд, если вы не знаете, что такое переменные окружения и с чем их едят.

Скачайте Maven (вам интересуют binary zip archive): ссылка. Распакуйте, куда считаете нужным. Установите переменную MAVEN\_HOME — место, куда вы его распаковали (например, у меня это C:\Program

Files\Maven\apache-maven-3.9.5). Добавьте вхождение %MAVEN\_HOME%\bin в переменную Path. Перезагрузите компьютер, чтобы изменения вступили в действие.

### Взаимодействие с репозиторием из консоли

Git Bash — эмулятор терминала для Windows. Изначально он откроется в “домашнем” каталоге — C:\Users\Ваш-пользователь. Для перемещения между каталогами используйте команду cd. Можно посмотреть справку по ней, набрав cd --help. Используйте cd .., чтобы перейти в родительский каталог, cd имя, чтобы перейти в каталог имя (Если в имени присутствуют пробелы или кириллица, нужно выделить имя в кавычки: cd "Имя каталога"). При нажатии Tab терминал дополнит название, если сможет (если у вас единственный файл/каталог с набранным префиксом)

Чтобы клонировать репозиторий, используйте команду

```
git clone <ссылка-на-репозиторий>
```

Будьте внимательны, Ctrl+C и Ctrl+V в консоли работают не ожидаемым образом. Вместо них для копирования и вставки используйте Ctrl+Insert и Shift+Insert. Репозиторий клонируется в каталог, в котором вы находитесь, создав подкаталог с именем репозитория.

Чтобы отправить на сервер своё решение, из каталога репозитория (после предыдущего шага, например, нужно было бы сделать cd kotlin-basics-LDemetrios) выполните команды

```
git add .
git commit 'Ваше сообщение'
git push
```

При клонировании и при пуше у вас могут потребовать ввести passphrase, которую вы указывали при создании ssh-ключа.

### Редакторы

Писать можно хоть в блокноте, но более приятно, всё же, с подсветкой и хоть каким-то автоформатированием. Из очень легковесных и бесплатных редакторов могу посоветовать Sublime Text.

### Собственно, сборка проекта.

Каждый репозиторий с заданием уже объявлен как Maven-проект (для этого в корне должен лежать файл pom.xml с описанием структуры проекта).

В задании по KotlinBasics не учтено, что может возникнуть необходимость работать без IDE, поэтому чтобы запускаться из консоли, поменяйте pom.xml в проекте на тот, что лежит здесь.

Из консоли (не из Git Bash, а именно из Windows'овой консоли), из каталога проекта (пользоваться можно той же командой `cd` для перемещения), выполните

```
mvn package
```

При первом запуске выполнение команды займёт довольно много времени — ей нужно скачать все зависимости, используемые в проекте (в нашем случае это `kotlin-stdlib`).

После этого можно запустить вашу программу:

```
java -jar target\KotlinBasics-1.0-SNAPSHOT-jar-with-dependencies.jar
```

(имя архива перед этим вам выведет команда `mvn package`:

```
[INFO] --- jar:3.3.0:jar (default-jar) @ KotlinBasics ---
[INFO] Building jar: C:\Users\Admin\oop-LDemetrios\target\KotlinBasics-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- assembly:3.6.0:single (make-assembly) @ KotlinBasics ---
[INFO] Building jar: C:\Users\Admin\oop-LDemetrios\target\KotlinBasics-1.0-SNAPSHOT-jar-with-
dependencies.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.122 s
[INFO] Finished at: 2023-10-08T17:03:35+03:00
[INFO] -----
```

— жирным выделено искомое имя )

Что происходит? `mvn package` в соответствии с описанием в `pom.xml` собирает программу в “цели” — в случае Java и JVM-языков это `jar`-файл. По сути, `jar` (Java ARchive) — это `zip`, внутри которого скомпилированные файлы и отдельный файл, указывающий, что является в программе главным (в нашем случае он только указывает на местоположение метода `main`). `java -jar <...>` — вы этот архив запускаете как `java`-программу. Дело в том, что Kotlin компилируется в JVM-bytecode, в то же, во что и Java. Поэтому исполнитель у них один и тот же.

Забавный факт: при создании задания по ООП я забыл поменять название проекта, поэтому там тоже архива будет называться `KotlinBasics` :).