

На курсе “Разработчик Spring Framework” используется следующее окружение:

- Git
- Java 11
- Maven 3+
- IntelliJ IDEA Community Edition (и выше) с установленным плагином Lombok

Далее приведены основные шаги по подготовке и настройке окружения для ОС Windows

Установка Git

Для установки git необходимо выполнить следующие действия:

1. Скачать дистрибутив по ссылке <https://git-scm.com/download>
2. Запустить установку и следовать инструкциям инсталлятора
3. Проверить, что все хорошо можно набрав в командной строке “git --version”. Должна быть выведена информация о версии git

Установка Java

Для установки java нужно выполнить следующие действия:

1. Скачать архив JDK (можно выбрать одну из указанных по ссылке: <https://jdk.dev/download/>)
2. Распаковать в выбранную директорию
3. Создать переменную окружения JAVA_HOME, указывающую на папку программы. Например “C:\Programs\JAVA\jdk11\”. Если на компьютере уже есть другая версия Java, переменная окружения настроена на нее и хотелось бы сохранить существующий порядок вещей, то данный этап можно пропустить. В этом случае потребуется дополнительная настройка maven
4. Проверить что переменная окружения настроена, верно, можно набрав в командной строке “ECHO %JAVA_HOME%”. Должен быть выведен путь к папке с нужной версией JDK
5. Проверить, что все хорошо можно набрав в командной строке “java --version”. Должна быть выведена информация о версии Java

Установка Maven

Для установки maven нужно выполнить следующие действия:

1. Скачать архив по ссылке <https://maven.apache.org/download.cgi>
2. Распаковать в выбранную директорию
3. Создать переменную окружения M2_HOME, указывающую на папку программы. Например “C:\Programs\JAVA\apache-maven-3.6.3\”
4. Проверить, что переменная окружения настроена, верно, можно набрав в командной строке “ECHO %M2_HOME%”. Должен быть выведен путь к папке с нужной версией Maven
5. Проверить, что все хорошо можно набрав в командной строке “mvn --version”. Должна быть выведена информация о расположении программы, версии Java и ОС
6. Maven работает с JDK указанной в переменной окружения JAVA_HOME. Если нужно, чтобы он работал с другой версией Java можно задать ее принудительно в файле “M2_HOME\bin\mvn.cmd”, указав там строку “set "JAVA_HOME=ПУТЬ_К_ПАПКЕ_С_JAVA””, где “ПУТЬ_К_ПАПКЕ_С_JAVA” это путь к папке с нужной версией JDK. Например, “C:\Programs\JAVA\jdk11\”. При необходимости эту строку можно отключить, поставив в ее начале “@REM”

Установка IntelliJ IDEA Community Edition

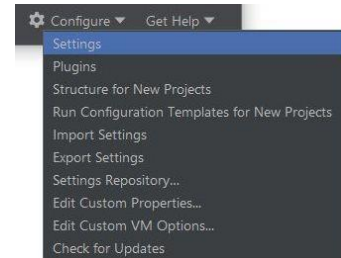
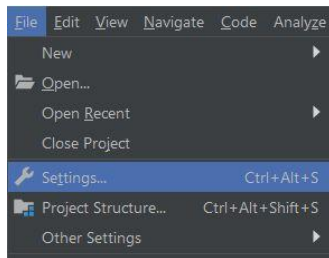
Для установки IntelliJ IDEA Community Edition необходимо выполнить следующие действия:

1. Скачать дистрибутив (не забыв выбрать именно вариант Community если нет лицензии на Ultimate) по ссылке <https://www.jetbrains.com/idea/download/>
2. Запустить установку и следовать инструкциям инсталлятора

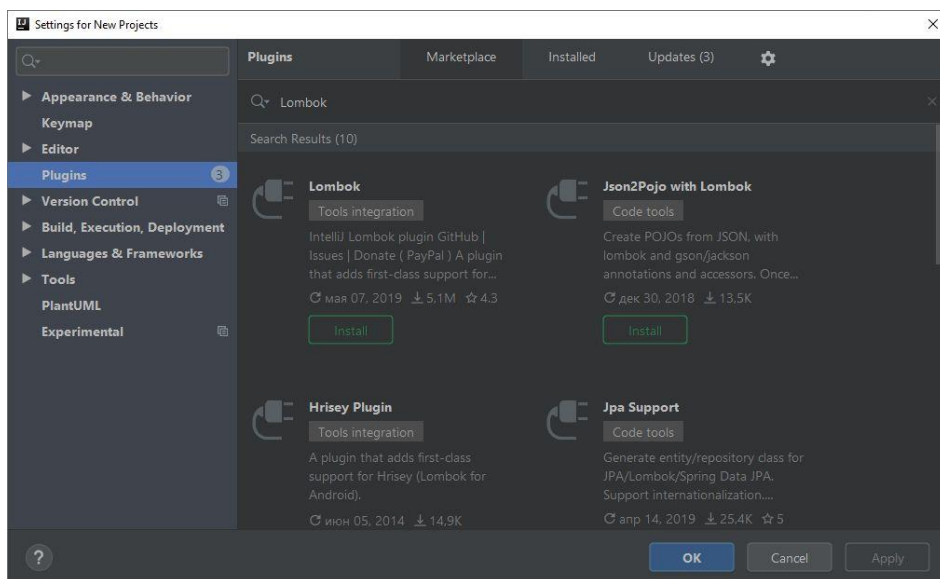
Установка плагина для библиотеки Lombok

Для установки плагина для библиотеки Lombok необходимо выполнить следующие действия:

1. Запустить среду разработки IntelliJ IDEA
2. Открыть настройки “Главное меню -> File -> Settings” или “Configure -> Settings” если не открыто ни одного проекта. В этом режиме можно попасть в нужные настройки, минуя пункт 3 с помощью “Configure -> Plugins”

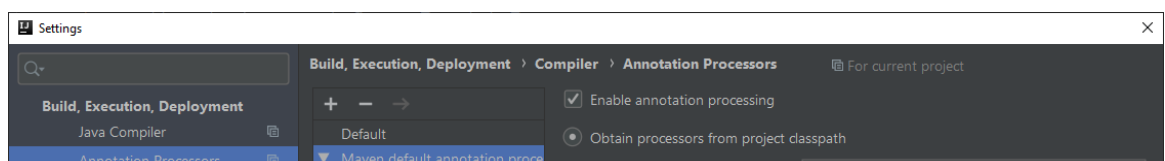


3. В дереве настроек выбрать узел “Plugins”
4. Выбрать вкладку “Marketplace”
5. В строке поиска, в разделе plugins ввести “Lombok” и нажать клавишу Enter
6. В появившемся списке найти одноименный плагин и нажать кнопку “Install”



7. Перезагрузить среду разработки

Стоит обратить внимание, что обработка аннотаций Lombok будет работать, только если в настройках установлен флаг Enable annotation processing, находящаяся в разделе настроек “Build, Execution, Deployment -> Compiler -> Annotation Processors”



На курсе “Разработчик Spring Framework” домашние работы сдаются в формате pull request-ов (далее ПР). Ниже приведены рекомендации к оформлению репозитория и краткая инструкция к формированию ПР.

Рекомендации к оформлению репозитория

Для того чтобы было легче, проверять домашние работы и ориентироваться внутри репозитория, в т.ч. потенциальным работодателям, желательно, чтобы он удовлетворял следующим требованиям:

1. Название репозитория должно быть в формате “YYYY-MM-otus-spring-XXX”, где YYYY-MM год и месяц начала группы длиной 4 и 2 цифры соответственно, а XXX фамилия слушателя
2. Для проверки будет лучше, если каждая новая работа будет сдаваться в новой папке внутри репозитория
3. В репозиторий не должны попадать файл и папки, не относящиеся напрямую к разрабатываемому приложению и git. В данном случае имеются в виду файлы и папки IDE (.idea, target, *.iml), файлы maven-wrapper (.mvn) и т.д.
4. Желательно, чтобы каждая папка с работой содержала файл Readme.md с описанием задания
5. Такой же файл будет не лишним разместить в корне репозитория, указав в нем ФИО слушателя, название курса, группу и список, описывающий в какой папке лежит та или иная выполненная работа

Краткая инструкция к формированию ПР (github)

Для того, чтобы сформировать ПР, в который бы попали коммиты только текущей работы необходимо выполнить следующие действия:

1. До выполнения работы (!), находясь в ветке master (main на GitHub) создать новую ветку под текущую работу. Это можно сделать с помощью команды “git checkout -b XXX”, где XXX название новой ветки
2. Выполнить в этой ветке работу
3. Закоммитить (команда “git commit”), запустить (команда “git push”)
4. Создать pull request на сайте странице github (Pull requests -> New pull request)
5. Послать ссылку на созданный ПР чат работы, в ЛК студента на сайте Отус
6. Когда работа будет принята сдвинуть ветку с master (main на GitHub) с помощью соответствующей кнопки в ПР на github (“Merge pull request”)

Первый пункт самый важный. Если создать ветку, когда уже часть работы выполнено, то в ПР. эти коммиты не попадут и на них нельзя будет написать комментарии

Основные команды git:

- **git checkout "branch_name"** – переключиться на существующую ветку (branch_name - имя ветки на которую нужно переключиться)
- **git checkout -b "branch_name"** – создать ветку
- **git add .** – добавить все (точка в конце команды) новые файлы под управление git
- **git commit -m "comment"** – коммит изменений (comment - комментарий к коммиту)
- Отправка изменений в удаленный репозиторий (branch_name - имя текущей ветки)
 - **git push --set-upstream origin "branch_name"** – первый раз после создания ветки
 - **git push** – во всех остальных случаях