



CodingBootcamp

Week 0 Day 0

Аннотация: Этот документ является практическим руководством к Week 0 Day 0 в CodingBootcamp.

Академия Ковалевского



Содержание

1. Что такое Java	2
2. Установка IntelliJ IDEA	3
3. Импорт проекта	4
4. Компиляция кода и Maven	5
5. Запуск тестов из Maven	6
6. Выводим "Hello world"	7





1. Что такое Java?

Справка

Определение

Java — строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle).

Типизация

Языки со строгой типизацией не позволяют смешивать в выражениях различные типы и не будут выполнять автоматически неявные преобразования типов. К примеру, нельзя вычесть из строки число или какой-то другой тип, не являющегося строкой.

Рекомендации

- Найти в поисковике [информацию](#) о типах языков программирования
- Посмотреть видеоурок "[Введение в Java](#)"
- Посмотреть видеоурок "[Виды языков программирования](#)"



Советы

- 👉 Пробуйте разные ключевые фразы в Google, чтобы найти нужную Вам информацию
- 👉 Используйте англоязычные запросы, т.к. полезной информации в англоязычном сегменте намного больше
- 👉 Используйте Google Translate, если возникают проблемы с пониманием информации на английском языке
- 👉 Если информация тяжела для понимания, найдите другой источник
- 👉 Пользуйтесь поиском по YouTube для нахождения тематических видеороликов
- 👉 Концентрируйтесь на конкретной задаче и поиске информации, связанной именно с ней



2. Установка IntelliJ IDEA

Справка

Среда разработки

Интегрированная среда разработки (Integrated development environment — IDE) — комплекс программных средств, используемый программистами для разработки программного обеспечения. Среда разработки включает в себя: текстовый редактор, транслятор (компилятор и/или интерпретатор), средства автоматизации сборки, отладчик.

IntelliJ IDEA

IntelliJ — одна из самых мощных и популярных интегрированных сред разработки (IDE) для Java. Эта многофункциональная IDE обеспечивает быструю разработку и помогает улучшить качество кода.

Задания

- Скачать IntelliJ IDEA Community с официального [сайта](#)
- Найти в поисковике [информацию](#) об установке среды разработки и установить её

3. Импорт проекта

Справка

Контроль версий

Система контроля версий (Version Control System, VCS) представляет собой программное обеспечение, которое позволяет отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления и т.п. Основным понятием VCS является репозиторий (repository) – специальное хранилище файлов и папок проекта, изменения в которых отслеживаются.

Git

Git — распределенная система контроля версий, разработанная Линусом Торвальдсем для работы над ядром операционной системы Linux. Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией.

GitHub

GitHub — сервис онлайн-хостинга репозитория, обладающий всеми функциями распределённого контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

Рекомендации

- Посмотреть видеоурок “[Как использовать систему контроля версий git](#)”

Задания

- Скачать соответствующий данному уроку репозиторий в виде zip архива
- Распаковать архив и импортировать проект в IntelliJ IDEA
- В настройках проекта выбрать “Add SDK” и “Download JDK”
- Добавить в вашей ОС переменную окружения “JAVA_HOME” и указать в ней путь к папке с загруженным JDK
- В переменную окружения “Path” добавить путь к папке “bin”, содержащейся в каталоге с загруженным JDK



4. Компиляция кода и Maven

Справка

Компиляция

Компиляция — процесс преобразования исходного текста программы, написанной на языке программирования высокого уровня, в программу на машинном языке, «понятную» компьютеру. Полученный код, называемый исполняемой программой, можно устанавливать и запускать на нужном компьютере без дополнительных преобразований.

Сборка проекта

Сборка (build) — процесс получения информационного продукта из исходного кода. Чаще всего включает компиляцию и компоновку, выполняется инструментами автоматизации.

Maven

Maven — унифицированный инструмент для автоматизации сборки проектов. Сборка при помощи Maven имеет ряд преимуществ по сравнению с обычной сборкой в IDE, особенно на больших проектах.

Рекомендации

- Найти в поисковике информацию о компиляции и автоматизированной сборке проектов

Задания

- Скачать Maven официального [сайта](#)
- Распаковать архив и добавить путь к папке “bin” в переменную окружения “Path”
- Протестировать правильность установки Maven, запустив в терминале IntelliJ IDEA команду “`mvn -v`”
- Собрать проект, выполнив команду “`mvn compile`”
- Исправить синтаксическую ошибку в коде

5. Запуск тестов из Maven

Справка

Тестирование

Тестирование — это процесс проверки функционала программы с целью подтверждения того, что она работает в соответствии с определёнными требованиями. Это крайне важный этап разработки ПО, который помогает создавать качественный продукт. Тестирование, которое пишется, непосредственно, на уровне разработчика называется Unit-тестирование. Оно делится на две большие группы — ручное и автоматизированное тестирование.

JUnit

JUnit — это фреймворк, разработанный для автоматического тестирования программ, написанных с использованием технологии Java. JUnit тест — это простой класс Java, методы которого помечены аннотацией `@Test`. Как правило, один метод отвечает за тестирование одного кейса. Пример:

```
1. public class MainTest {  
2.     @Test  
3.     public void testGreen() {  
4.         assertTrue(true);  
5.     }  
6. }
```

Рекомендации

- Найти в поисковике информацию о тестировании при помощи JUnit

Задания

- Запустить все тестовые классы выполнив команду `"mvn test"`
- Запустить один тестовый класс `MainTest` выполнив команду `"mvn -Dtest=MainTest test"`
- Запустить один тестовый метод `testGreen()` из тестового класса `MainTest` выполнив команду `"mvn -Dtest=MainTest#testGreen test"`



6. Выводим “Hello world”

Справка

Привет мир

Традиция использования фразы “Hello world” в качестве тестового сообщения берёт начала от примера программы из книги 1978 года под названием “The C Programming Language”. С тех пор миллионы разработчиков начали свои карьеры с этой простой программы.

Задания

- Выведите в консоль терминала “Hello world” при помощи команды `“mvn compile exec:java -Dexec.mainClass=com.kovalevskyi.academy.codingbootcamp.week0.day0.Main”`
- Скомпилируйте и запустите java-программу из командной строки при помощи команд `“javac”` и `“java”`