

Dev tools (IDE, системы сборки, системы контроля версий):

- Безопасный рефакторинг – применить изменения во всем проекте можно за пару кликов.
- Функция Live Edit позволяет мгновенно посмотреть все изменения в браузере
- Поддержка удаленной разработки по SSH
- Инструменты для работы с базами данных и SQL файлами
- Инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования
- Удаленная отладка
- Создание прототипов без написания кода

IDE:

- Microsoft Visual Studio
- NetBeans
- PyCharm
- IntelliJ IDEA
- Eclipse
- RubyMine
- Xcode
- WebStorm
- Visual Studio Code



Microsoft Visual Studio:

- Платформа Windows и MacOS
- Бесплатная и платная (от 1200\$ в год) версия
- Создана для разработки на C/C++, C#, .Net, TypeScript, Node.JS
- Интеграция с Azure



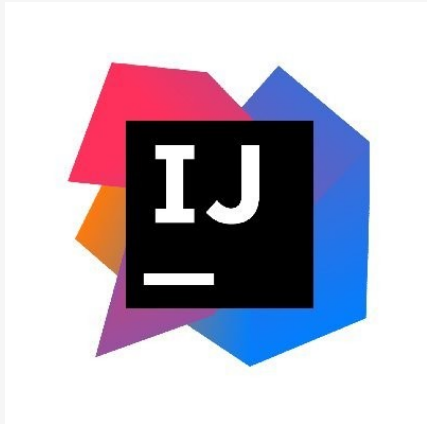
NetBeans:

- Платформа Windows, Linux, Mac OSX and BSD
- Бесплатная версия
- Создана для разработки на Java, PHP, Javascript, HTML5, CSS



PyCharm:

- Платформа Windows, macOS и Linux
- Бесплатная и платная(90\$ в год) версия
- Создана для разработки на Python, JavaScript, CoffeeScript, TypeScript, CSS



IntelliJ IDEA:

- Платформа Windows, macOS и Linux
- Бесплатная и платная(150\$ в год) версия
- Создана для разработки на Java, Kotlin, Groovy, Scala (JavaScript, TypeScript в платной версии)



Eclipse:

- Платформа Windows, macOS и Linux
- Бесплатная версия
- Создана для разработки на Java, C/C++, JavaScript, TypeScript, PHP, CSS, HTML
- Возможность расширения и установки любых плагинов



RubyMine:

- Платформа Windows, macOS и Linux
- Только платная версия (90\$ в год)
- Создана для разработки на Ruby on Rails, JavaScript, CoffeeScript, ERB and HAML, CSS



Xcode:

- Платформа только macOS
- Бесплатная версия в составе MacOS
- Создана для разработки на Objective-c, Swift
- Имеет интеграцию со всеми мобильными сервисами Apple и Appstore



WebStorm:

- Платформа Windows, macOS и Linux
- Только платная (60\$ в год) версия
- Создана для разработки на Javascript и TypeScript
- Поддерживает все популярные фреймворки JavaScript



Visual Studio Code:

- Платформа Windows, macOS и Linux
- Бесплатная версия
- Создана для разработки на любых языках за счет расширяемости
- Поддерживает огромное количество расширений и плагинов

Интерпретируемые и компилируемые языки:

- Компилируемые языки
- Интерпретируемые языки
- Jit-компилируемые языки

Компилируемые языки:

- C/C++
- Haskell
- Rust
- Go
- C#/F#
- Swift

Интерпретируемые языки:

- Basic
- JavaScript
- Perl
- PHP
- Python

Jit-компилируемые языки:

- Java
- Ruby
- Python

Веб разработка:

- Статические сайты (HTML, CSS и JavaScript)
- Динамические сайты (JS, PHP, ASP, TypeScript)

Пакеты и пакетные менеджеры:

- NPM
- Maven
- Graddle
- Ant
- Nugget



NPM:

Package.json

```
{
  "name": "package.json-mastery",
  "version": "1.0.0",
  "description": "Mastery of the package.json file",
  "private": false,
  "main": "index.js",
  "scripts": {
    "start": "node index",
    "dev": "nodemon index",
    "test": "jest"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/Easybuoy/package.json-mastery.git"
  },
  "keywords": [
    "node",
    "javascript",
    "npm",
    "yarn"
  ],
  "author": "Ezekiel Ekunola",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/Easybuoy/package.json-mastery/issues"
  }
}
```

Maven:

Pom.xml

```
52 <dependencyManagement>
53   <dependencies>
54     <dependency>
55       <groupId>org.springframework.cloud</groupId>
56       <artifactId>spring-cloud-dependencies</artifactId>
57       <version>${spring-cloud.version}</version>
58       <type>pom</type>
59       <scope>import</scope>
60     </dependency>
61   </dependencies>
62 </dependencyManagement>
63
64 <build>
65   <plugins>
66     <plugin>
67       <groupId>org.springframework.boot</groupId>
68       <artifactId>spring-boot-maven-plugin</artifactId>
69     </plugin>
70   </plugins>
71 </build>
72
73 <repositories>
74   <repository>
75     <id>spring-milestones</id>
76     <name>Spring Milestones</name>
77     <url>https://repo.spring.io/milestone</url>
78   </repository>
79 </repositories>
80
81 </project>
```

Gradle:

Build.gradle

```
project.afterEvaluate {

    android.applicationVariants.each {
        ApplicationVariant variant ->
            String variantName = variant.name.capitalize()
            def mergeManifestTask = project.tasks.getByTaskName("process${variantName}Manifest")
            mergeManifestTask.doLast {
                mm ->
                    def manifestPath = mm.manifestOutputFile
                    if(project.hasProperty("channel")) {
                        addChannel(manifestPath)
                    }
            }
    }
}

def addChannel(File manifest) {
    def channelNo = project.property("channel")
    def xml = new XmlParser().parse(manifest)
    xml.application[0].appendNode("meta-data", ['android:name': 'channel', 'android:value': channelNo])
    manifest.withPrintWriter("UTF-8") {
        XmlUtil.serialize(xml, it)
    }
}
```

Ant:

Build.xml

```
1 <project name="SalesforceTest"
2     default="test"
3     basedir="."
4     xmlns:sf="antlib:com.salesforce">
5
6     <taskdef resource="com/salesforce/antlib.xml" uri="antlib:com.salesforce">
7         <classpath>
8             <pathelement location="../../ant-salesforce.jar" />
9         </classpath>
10    </taskdef>
11
12    <property file="build.properties"/>
13    <property environment="env"/>
14
15    <condition property="sf.username" value=""><not><isset property="sf.username"/></not></condition>
16    <condition property="sf.password" value=""><not><isset property="sf.password"/></not></condition>
17
18    <target name="retrieve">
19        <mkdir dir="retrieve"/>
20        <sf:retrieve
21            username="{sf.username}"
22            password="{sf.password}"
23            serverurl="{sf.serverurl}"
24            retrieveTarget="retrieve"
25            unpackaged="package.xml"/>
26    </target>
27
28 </project>
```

NuGet:

Csproj(часть)

```
1 <Project Sdk="Microsoft.NET.Sdk" ToolsVersion="15.0">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>netcoreapp1.0</TargetFramework>
6   </PropertyGroup>
7
8   <ItemGroup>
9     <Compile Include="**\*.cs" />
10    <EmbeddedResource Include="**\*.resx" />
11  </ItemGroup>
12
13  <ItemGroup>
14    <PackageReference Include="Microsoft.NETCore.App" Version="1.0.1" />
15  </ItemGroup>
16
17 </Project>
```

Домашнее задание:

Deadline: **1.10.2021**

Написать функционирующего телеграмм бота на языке GO имеющего минимум 3 команды:

- Git возвращает адрес вашего репозитория.
- Tasks возвращает нумерованный список ваших выполненных заданий.
- Task# где # номер задания, возвращает ссылку на папку в вашем репозитории с выполненной задачей.