

2



Classes in Kotlin

PROJETO BÁSICO DA AULA - CONCEITO E PRÁTICAS SOBRE CLASSE

#Kotlin



Udinei Silva

20/08/2021 20:29

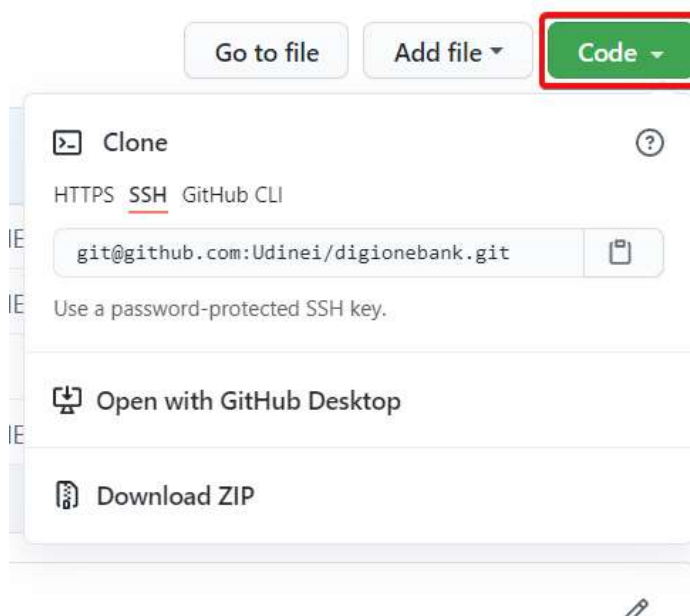
PROJETO BÁSICO DA AULA - CONCEITO E PRÁTICAS SOBRE CLASSE

Pelo fato de não ter achado o projeto zerado no github do professor, e querer acompanhar as aulas codificando e fazendo os exercícios, eu resolvi criar esse projeto zerado, e subir para o github para compartilhar com vocês, pra quem quer acompanhar as aulas sem ter criar o projeto do zero.

E pra quem quiser criar o projeto do zero, compartilho abaixo o passo a passo de criação do projeto no intellij.

Você pode baixar o projeto zerado (básico) do meu github para acompanhar as aulas acessando o link abaixo.

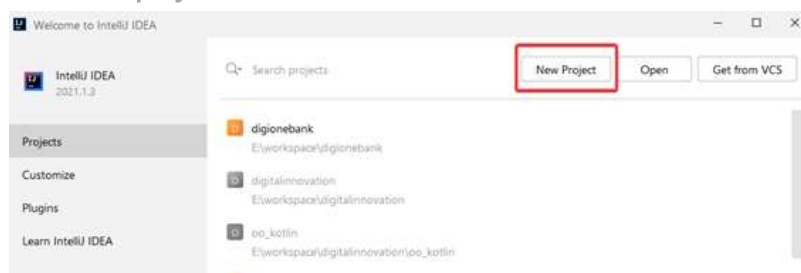
<https://github.com/Udinei/digionebank>



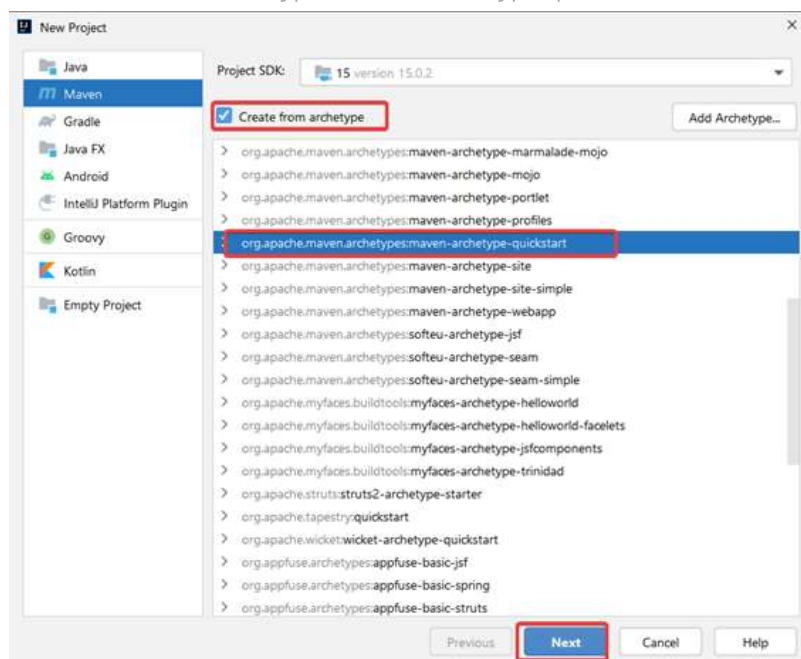
ou você pode seguir os passos abaixo para criar no IntelliJ um projeto zerado, com a estrutura igual ao da aula.

CRIAÇÃO DO PROJETO NO INTELIJ

Criando o projeto



Selecionando ao archetype “maven-archetype.quickstart”



Informando o nome e local do projeto no disco local

New Project

Name: **MyProject**

Location: E:\workspace\MyProject\MyProjec

Artifact Coordinates

GroupId: org.example
The name of the artifact group, usually a company domain

ArtifactId: MyProjec
The name of the artifact within the group, usually a project name

Version: 1.0-SNAPSHOT

Previous Next Cancel Help

Editando e confirmando os dados do projeto

New Project

Maven home path: Bundled (Maven 3)

(Version: 3.6.3)

User settings file: C:\Users\Udinei\m2\settings.xml ☐ Override

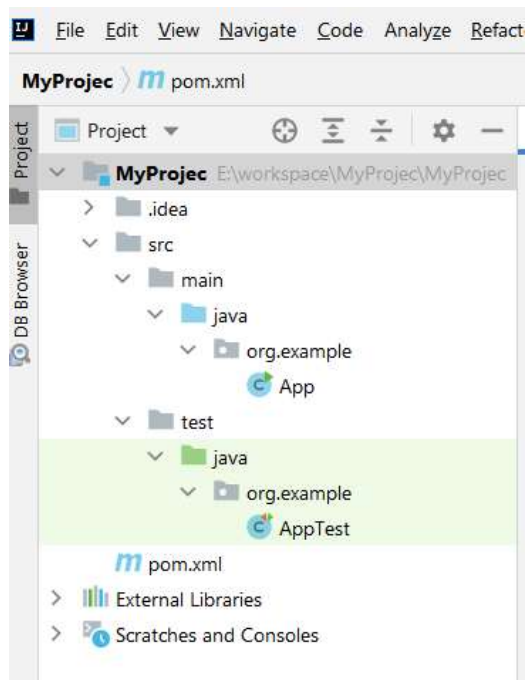
Local repository: C:\Users\Udinei\m2\repository ☐ Override

Properties

groupId	org.example
artifactId	MyProjec
version	1.0-SNAPSHOT
archetypeGroupId	org.apache.maven.archetypes
archetypeArtifactId	maven-archetype-quickstart
archetypeVersion	RELEASE

Previous Finish Cancel Help

O IntelliJ cria o projeto baseado archetype do maven "maven-archetype.quickstart" com a seguinte estrutura



Apague a pasta java das pastas main e test, e crie nessas mesmas pastas(main e test) a pasta kotlin

Copiar todo o conteúdo do arquivo pom.xml do projeto no endereço (pom.xml do projeto do professor)

https://raw.githubusercontent.com/jether2011/digitalinnovation/main/fundamentos_oo/pom.xml

e colar (sobrepôr) todo conteúdo no pom.xml do projeto

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns="http://maven.apache.org/POM/4.0.0"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <artifactId>digionebank</artifactId>
    <groupId>one.digitalinnovation.bank</groupId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>digionebank</name>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <kotlin.code.style>official</kotlin.code.style>
        <kotlin.compiler.jvmTarget>1.8</kotlin.compiler.jvmTarget>
    </properties>

    <repositories>
        <repository>
            <id>mavenCentral</id>
            <url>https://repo1.maven.org/maven2/</url>
        </repository>
    </repositories>

    <build>
        <sourceDirectory>src/main/kotlin</sourceDirectory>
        <testSourceDirectory>src/test/kotlin</testSourceDirectory>
        <plugins>
            <plugin>
                <groupId>org.jetbrains.kotlin</groupId>
                <artifactId>kotlin-maven-plugin</artifactId>
                <version>1.4.21</version>
                <executions>
                    <execution>
                        <id>compile</id>
                        <phase>compile</phase>
                        <goals>
                            <goal>compile</goal>
                        </goals>
                    </execution>
                    <execution>
                        <id>test-compile</id>
                        <phase>test-compile</phase>
                        <goals>
                            <goal>test-compile</goal>
                        </goals>
                    </execution>
                </executions>
            </plugin>
        </plugins>
    </build>

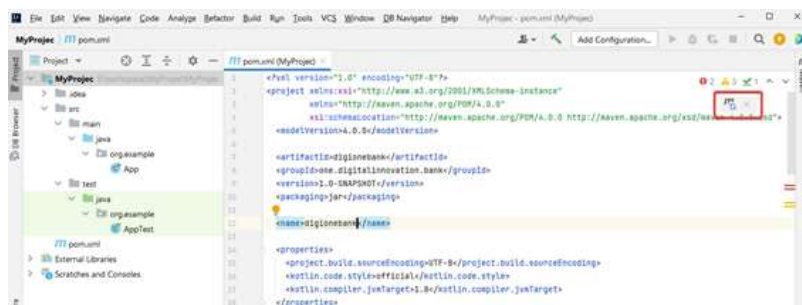
    <dependencies>
        <dependency>
            <groupId>org.jetbrains.kotlin</groupId>
            <artifactId>kotlin-test-junit5</artifactId>
            <version>1.4.21</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-api</artifactId>
```

```

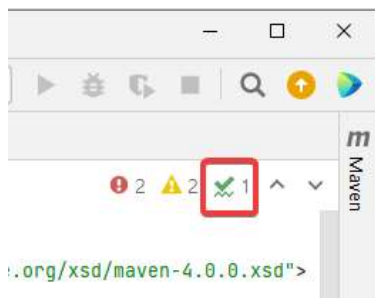
    <version>5.6.0</version>
    <scope>test</scope>
  </dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter-engine</artifactId>
  <version>5.6.0</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.jetbrains.kotlin</groupId>
  <artifactId>kotlin-stdlib-jdk8</artifactId>
  <version>1.4.21</version>
</dependency>
</dependencies>
</project>

```

Atualizando o projeto... clique no ícone do plugin do Maven, assim automaticamente todo o projeto será atualizado com o novo conteúdo do pom.xml



Clique no ícone para confirmar as alterações no projeto



Feche e abra o projeto novamente no IntelliJ, verifique se está confirmada e valide as alterações realizadas no pom.xml do projeto



Projeto configurado!

Vlw!

Comentário

Normal

COMENTAR

Comentários (1)

Mais Antigos



0



Maria Moura
20/08/2021 22:46

Obigada!



Udinei Silva

Trabalho no desenvolvimento de sistemas e na arquitetura utilizando a linguagem Java e outras tecnologias como Spring Boot, HTML, CSS, MySql, Postgres, Oracle, JavaScript, JPA etc..

 SEGUIR A PUBLICAÇÃO

 Brasil