

Установка среды разработки

В этом видео мы разберём процесс установки среды разработки, в которой создаётся программное обеспечение на языке Java. Установим **IntelliJ Idea** — самую популярную среду разработки на Java.

Существуют и другие среды разработки: Eclipse или NetBeans. Но они менее удобные и используются гораздо реже.

Скачивание и установка

Переходим на страницу скачивания на официальном сайте JetBrains:

<https://www.jetbrains.com/ru-ru/idea/download/>

Доступны два варианта — *Ultimate Edition* и *Community Edition*. Первая — платная, а вторая — бесплатная, но с меньшим функционалом.

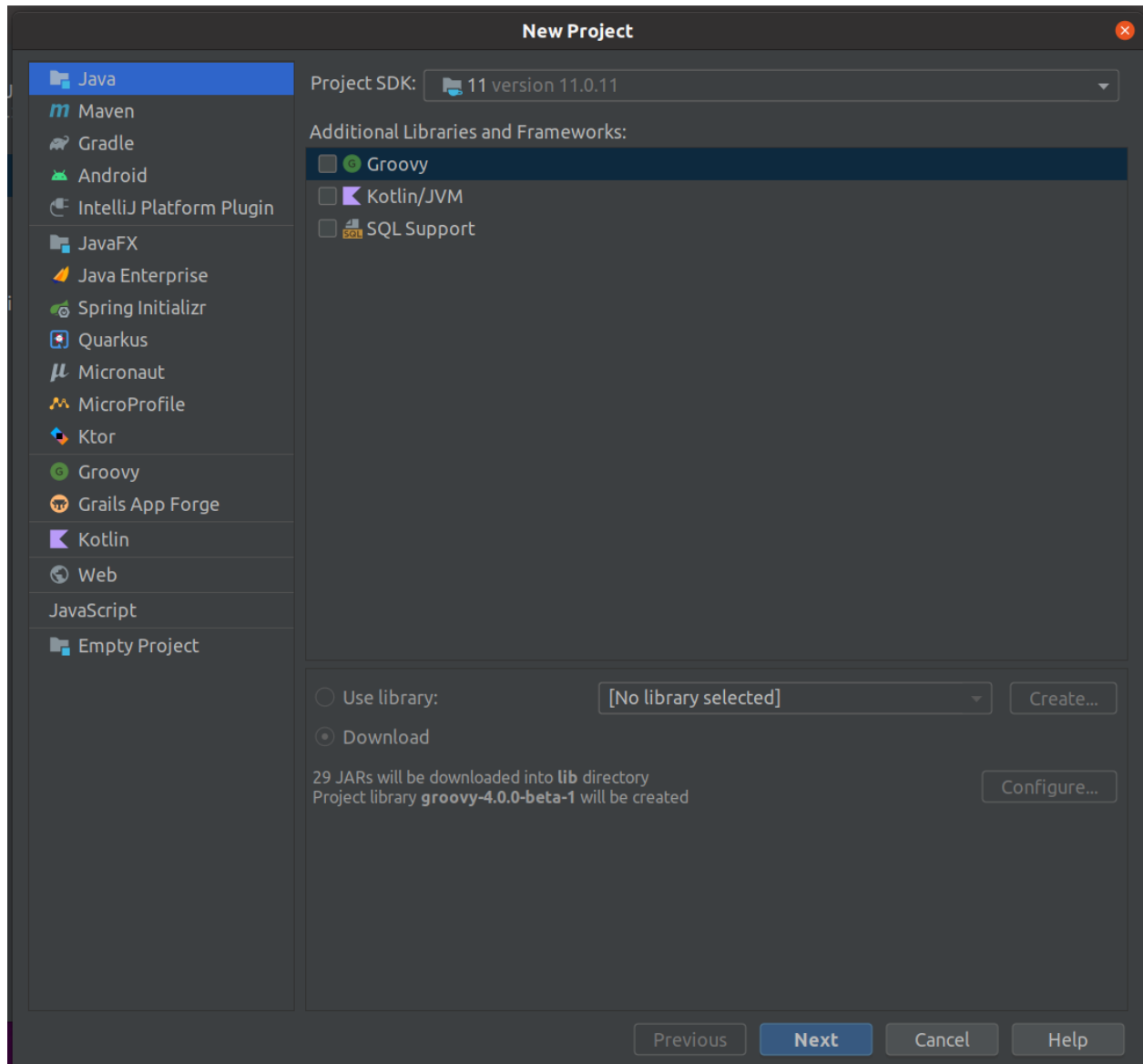
Мы установим бесплатную версию Community. Всё обучение пройдёт в ней, так как её функционала будет достаточно.

Инструкция по установке среды разработки IntelliJ IDEA под все операционные системы:

https://skillbox.ru/media/base/kak_ustanovit_jdk_i_sredu_razrabotki_intellij_idea/

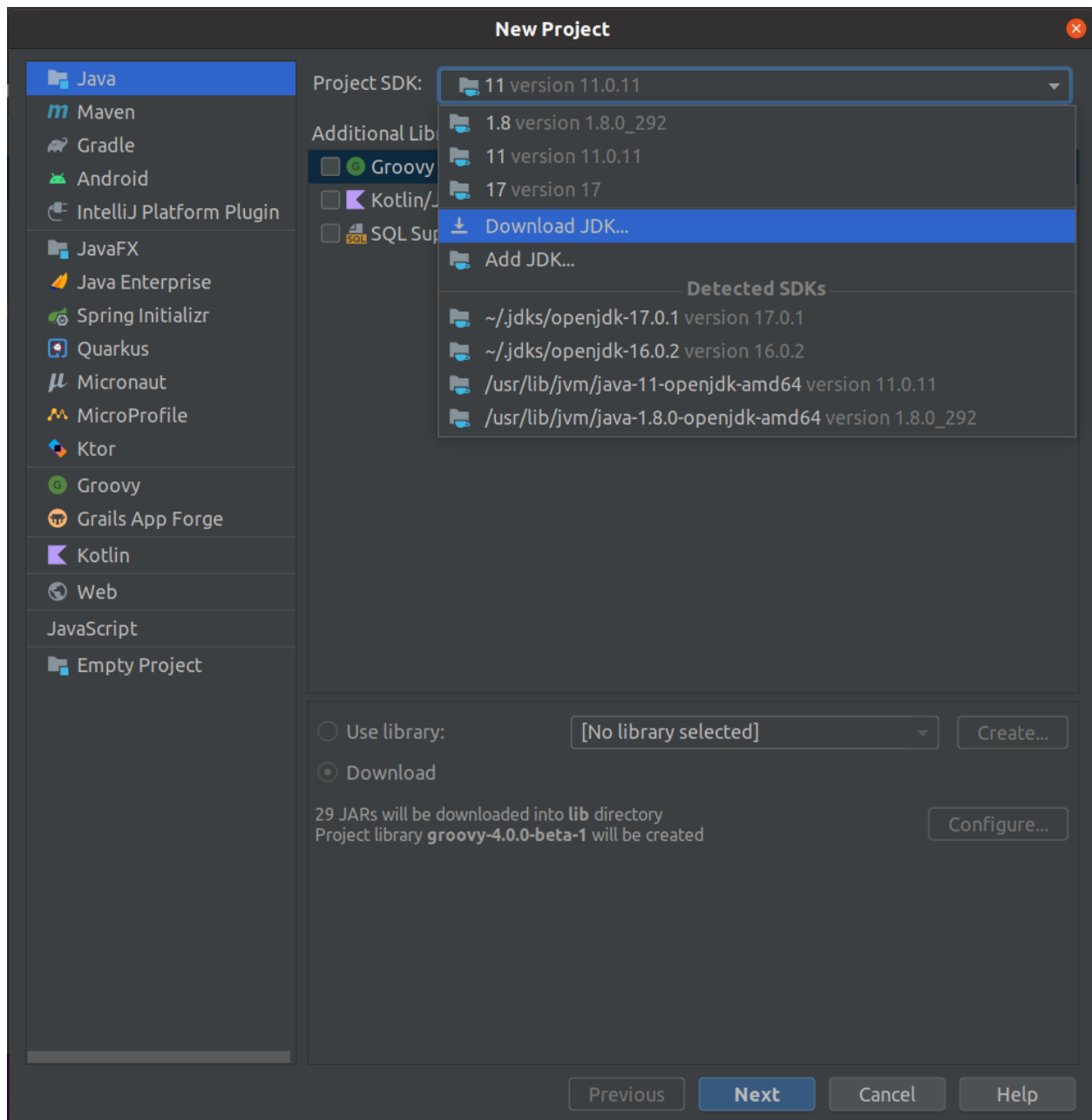
Создание проекта

Проект — это папка, в которой лежат другие папки и файлы с программным кодом.



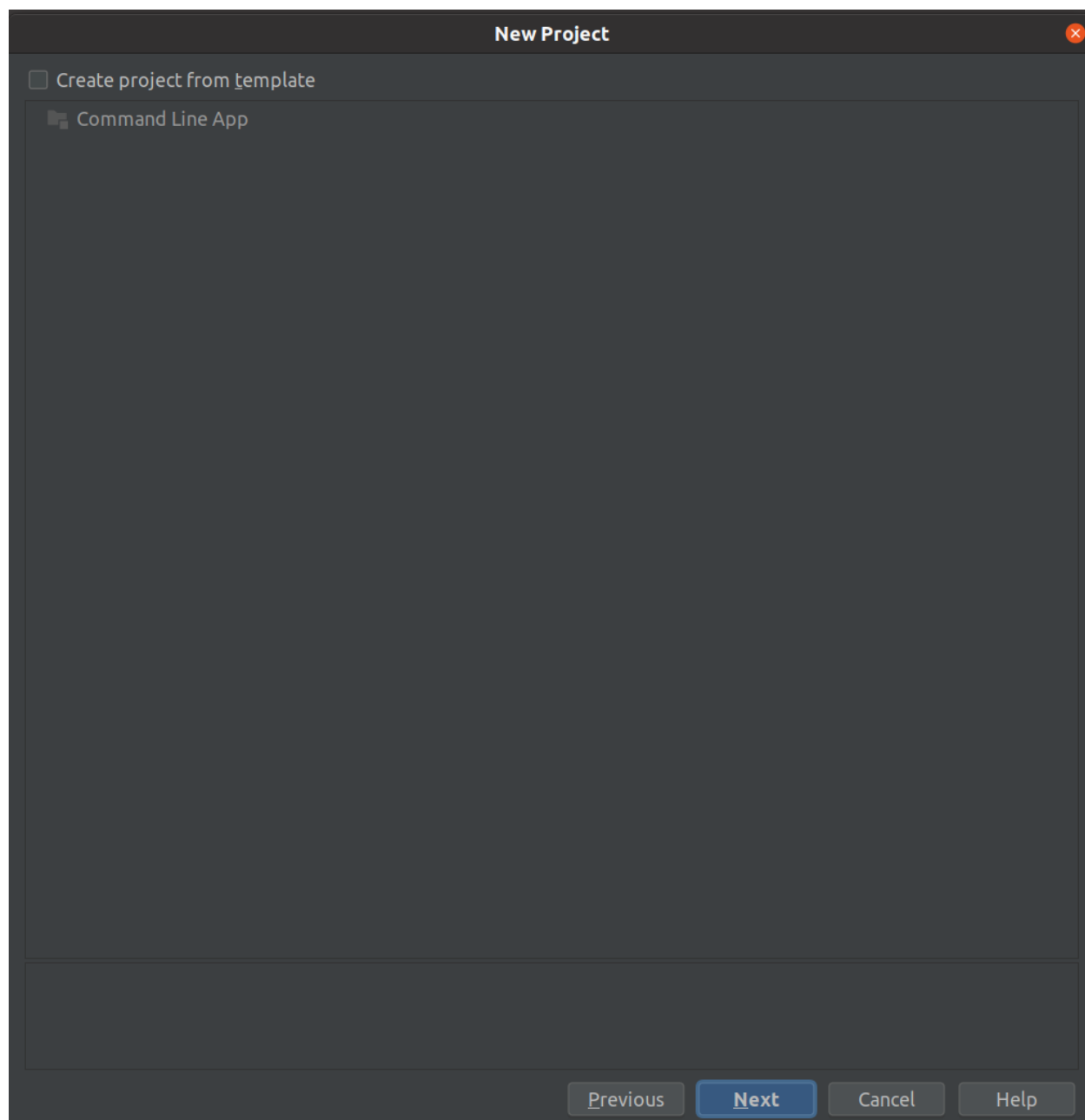
Предлагается выбрать SDK — Software Development Kit. Но SDK — просто более общий термин. Нам нужен Java Development Kit, так как мы будем писать программу на Java. Это специальное программное обеспечение, необходимое для разработки на Java. Если его нет на компьютере, то можно скачать и установить его прямо в этом окне.

Лучше выбрать Oracle OpenJDK 11.

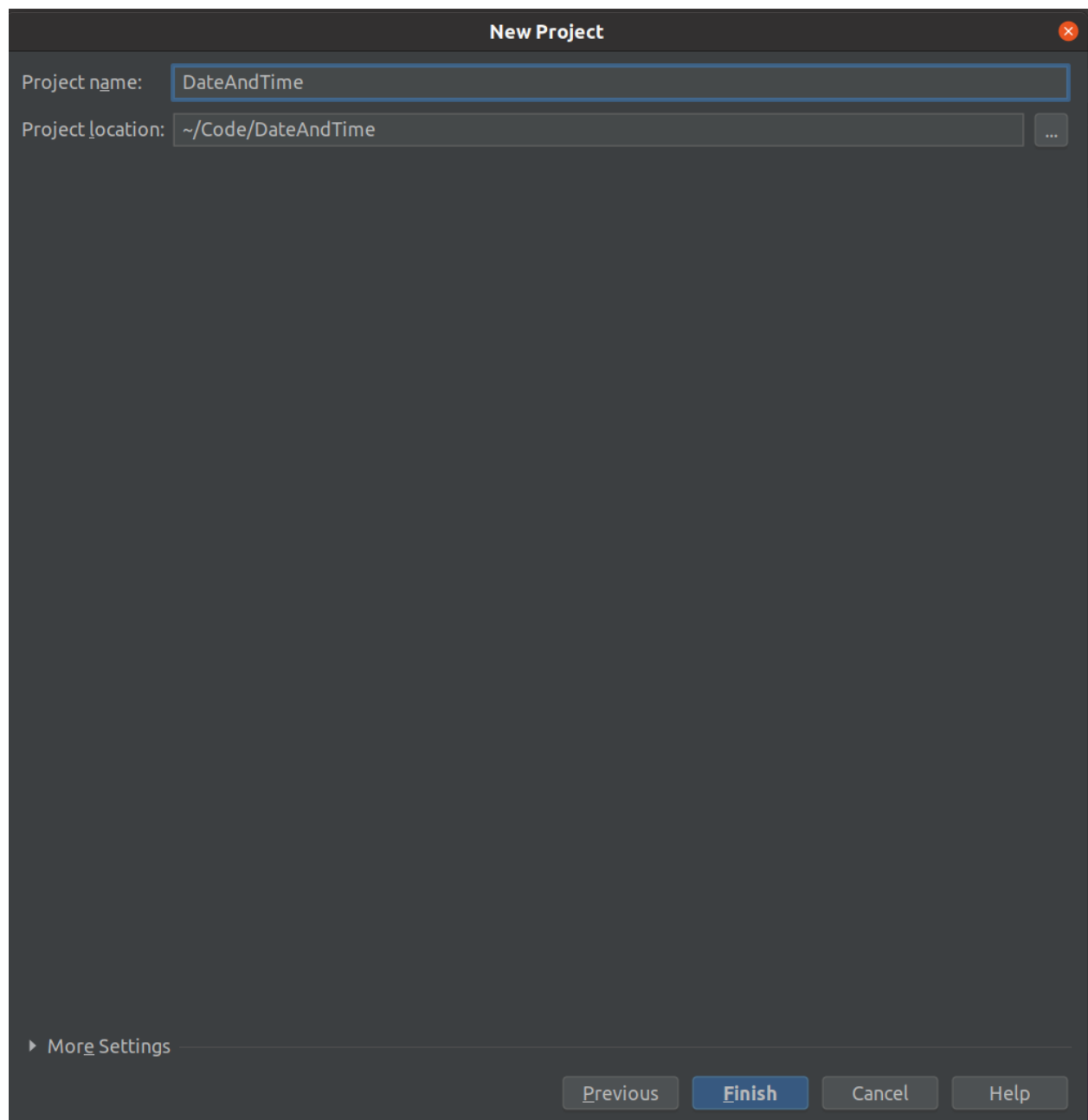


Теперь, когда мы будем создавать новые проекты, установленный нами JDK будет выбираться автоматически. Но можно выбрать и любую другую версию при создании нового проекта или позже в настройках

Следующий шаг проходим, ничего не изменяя.



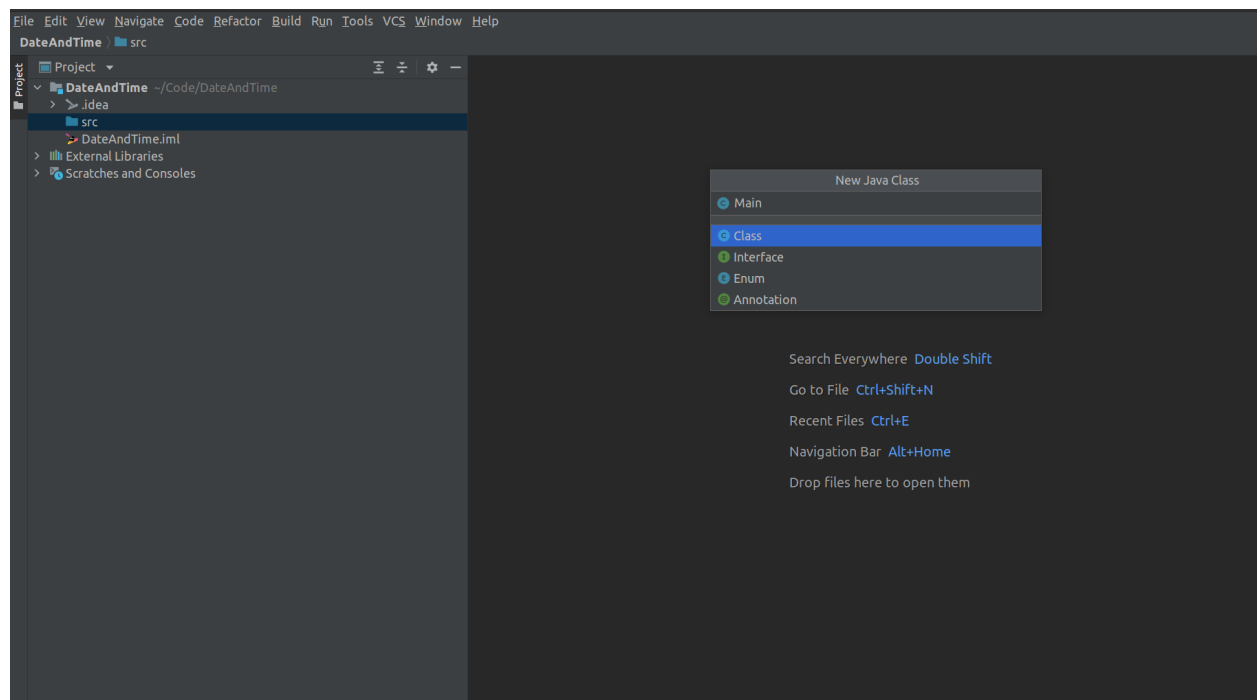
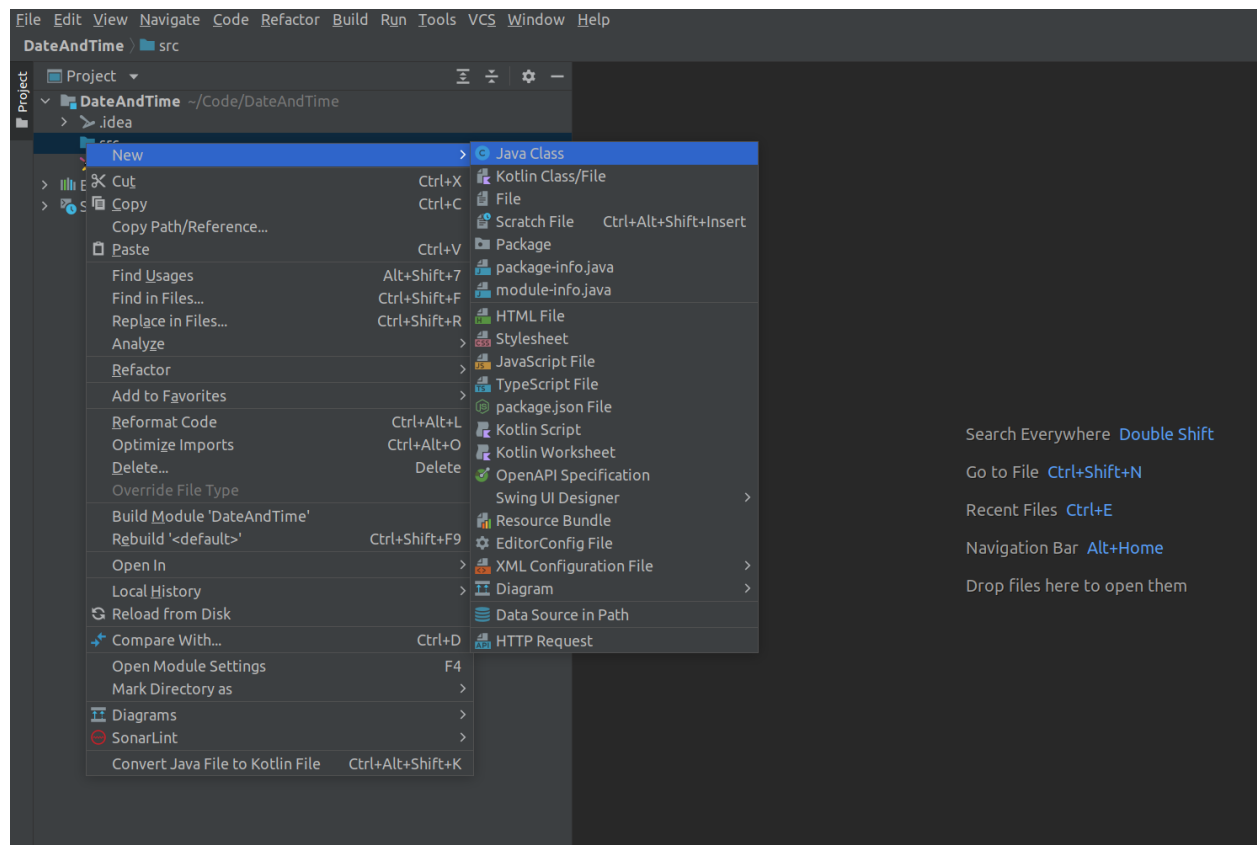
На следующем шаге задаём проекту имя `DateAndTime`. Здесь же можно выбрать папку на компьютере, в которой будут храниться проекты.



Завершим создание проекта, нажав кнопку Finish.

Создание класса Main и метода main()

Напишем программу, которая будет сообщать текущую дату и время. Чтобы начать писать программный код, нужно создать класс (файл). Создаём файл Main.



В классе Main создаём блок кода, который называется main. Это просто: достаточно написать main и нажать Enter:

```
1 public class Main {  
2  
3     public static void main(String[] args) {  
4         |  
5     }  
6 }  
7
```

Любая программа на Java всегда начинается с этого блока кода. Поэтому тот код, который будет выполняться первым, нужно писать внутри фигурных скобок метода `main()`.

Обратите внимание, что среда разработки сгенерировала этот блок автоматически, и в нём написано много незнакомых слов: `public static void main(String[] args)`. Сейчас мы пропустим этот момент, но в следующих модулях познакомимся со значением всех этих слов.

Написание кода программы

Первой напомним команду `System.out.println()`. Её можно сгенерировать в Идее автоматически, написав четыре буквы `sout` и нажав `Enter` или `Tab`:

```
1 public class Main {
2
3     public static void main(String[] args) {
4
5         sout
6     }
7 }
8
```

sout Prints a string to System.out
soutm Prints current class and method names to System.out
soutp Prints method parameter names and values to System.out
soutv Prints a value to System.out

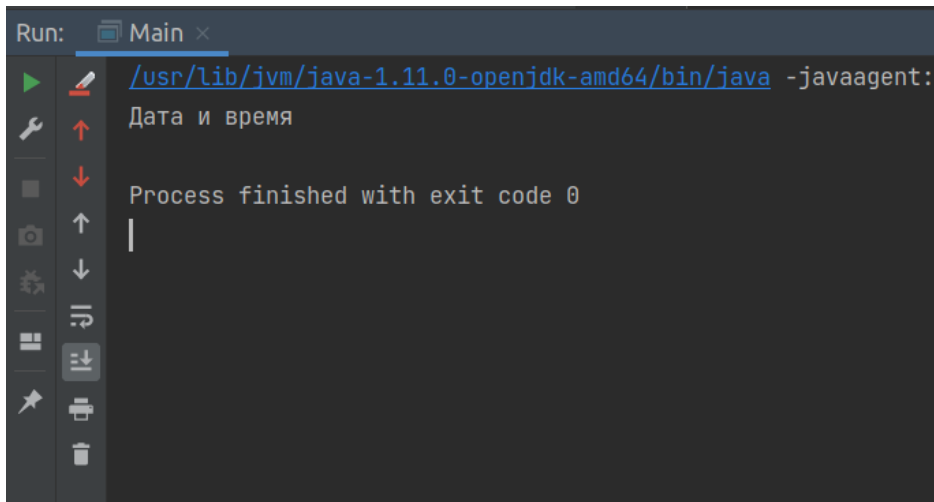
Press Ctrl+Space to see non-imported classes Next Tip

Эта команда важна, так как выводит в консоль текст, переданный в её фигурные скобки. Консоль — это место, где можно общаться с программой: вводить текст и читать сообщения.

Выведем в консоль фразу «Дата и время» и запустим программу зелёным треугольником на полях редактора кода:

```
1 public class Main {
2
3     public static void main(String[] args) {
4
5         System.out.println("Дата и время");
6     }
7 }
8
9
10
11
12
13
```

Run 'Main.main()' Ctrl+Shift+F10
Debug 'Main.main()'
Run 'Main.main()' with Coverage
Run 'Main.main()' with 'Java Flight Recorder'
Run 'Main.main()' with 'Async Profiler'
Modify Run Configuration...




```
Run: Main x
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:
Дата и время
Process finished with exit code 0
```

Видим, что всё работает. Осталось вывести текущую дату и время. Это можно сделать, написав:

```
System.out.println(LocalDate.now());
```

В скобках указана команда Java, которая возвращает текущую дату и время

Если слово `LocalDateTime` выделено красным, то нужно навести на него курсор и во всплывающем окне нажать `Import class`.

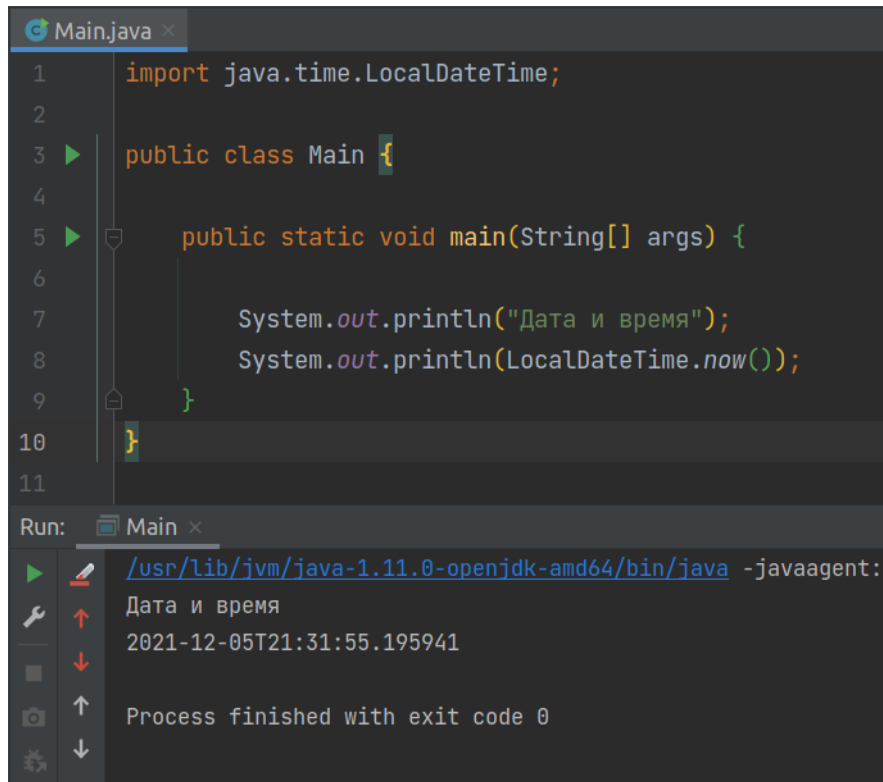


```
1 public class Main {
2
3 public static void main(String[] args) {
4
5     System.out.println("Дата и время");
6     System.out.println(LocalDate.now());
7 }
8 }
9
```

Cannot resolve symbol 'LocalDateTime'

[Import class](#) Alt+Shift+Enter [More actions...](#) Alt+Enter

Снова запустим программу и проверим результат:

The image shows a screenshot of an IDE with a dark theme. At the top, a tab labeled 'Main.java' is open. The editor displays the following Java code:

```
1 import java.time.LocalDateTime;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7         System.out.println("Дата и время");
8         System.out.println(LocalDateTime.now());
9     }
10 }
11
```

Below the editor, a 'Run' panel is visible, showing the execution of the program. The command used is `/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:`. The output consists of two lines: 'Дата и время' and '2021-12-05T21:31:55.195941'. At the bottom, it states 'Process finished with exit code 0'.

Итоги

Первое, что мы сделали, — скачали среду разработки с официального сайта. Выбрали бесплатную версию — Community Edition. Далее мы установили её, каждый раз нажимая кнопку Next и не меняя параметры установки по умолчанию.

Затем мы начали создавать новый проект, в процессе скачали и установили JDK — Java Development Kit, специальное программное обеспечение для разработки программ на Java. После этого задали имя проекту и начали работать над приложением.

Первым делом создали в папке `src` новый файл. При этом в выпадающем списке выбрали Class и задали файлу имя Main. Затем в созданном файле внутри фигурных скобок написали слово `main`, нажали на появившуюся подсказку. После этого среда разработки автоматически создала блок кода, в котором мы продолжили писать программу. Этот блок кода называется методом `main`. С него начинается выполнение любой программы в Java.

В одном из следующих модулей мы отдельно рассмотрим, что такое классы и методы.

Затем мы написали команду `System.out.println()`. Она выводит переданный ей в скобках текст в консоль. Мы вывели текст «Дата и время». После этого написали вторую такую же команду, но ей передали `LocalDateTime.now()` — специальный код, который возвращает текущие дату и время.

В следующем видео мы напишем более сложный программный код.