

ВВЕДЕНИЕ

Работа с IntelliJ IDEA

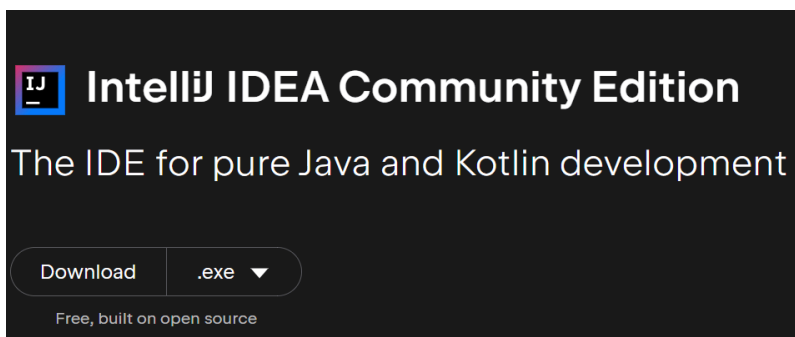
IntelliJ IDEA представляет собой **интегрированную среду разработки (IDE)**, созданную для работы с языком программирования Java. Она предоставляет множество инструментов для удобной и эффективной разработки Java-приложений.

Однако, перед началом работы в IntelliJ IDEA необходимо установить **Java Development Kit (JDK)**, который содержит необходимые инструменты и библиотеки для разработки на Java. JDK включает в себя компилятор, исполнительное окружение и другие ресурсы.

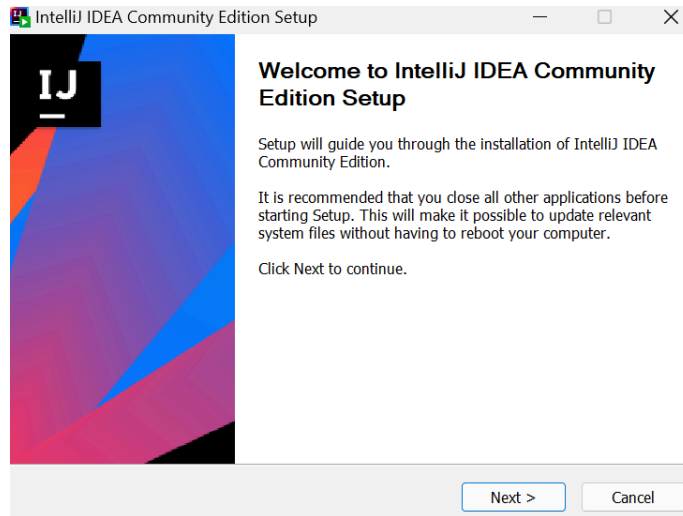
Интересно, что IntelliJ IDEA позволяет установить JDK непосредственно из IDE, что делает настройку проще и удобнее. Это облегчает начало работы с Java-разработкой, даже если у вас не было предварительно установлено JDK.

Инструкция по установке и запуску первого проекта

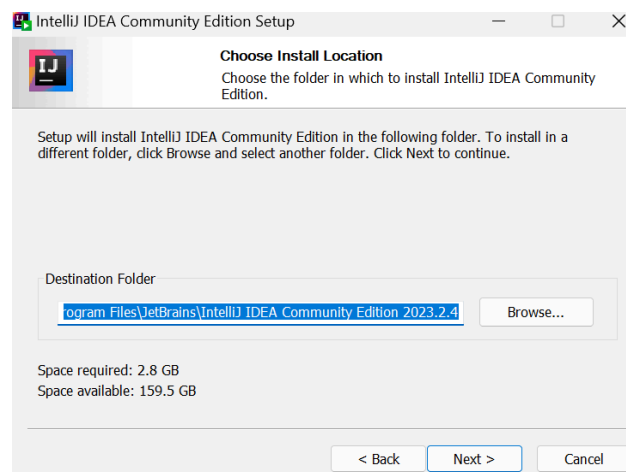
1. Переходим по ссылке - <https://www.jetbrains.com/idea/download/>
2. Прокручиваем страницу вниз
3. Находим кнопку для скачивания Community-версии - **Download**



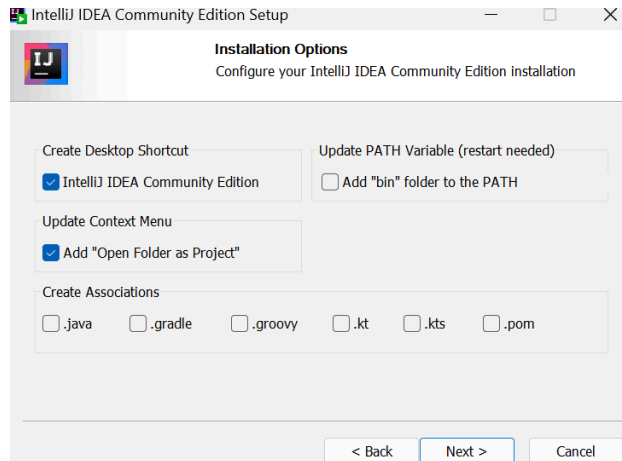
4. Запускаем скачанный установщик и нажимаем **Next**



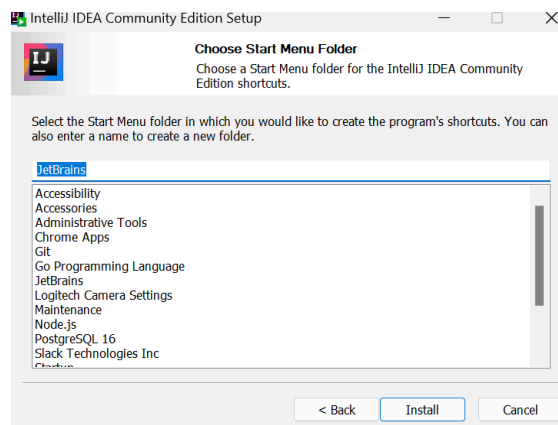
5. Можно указать собственный путь для установки, либо оставить тот, который предлагает установщик. Нажимаем **Next**



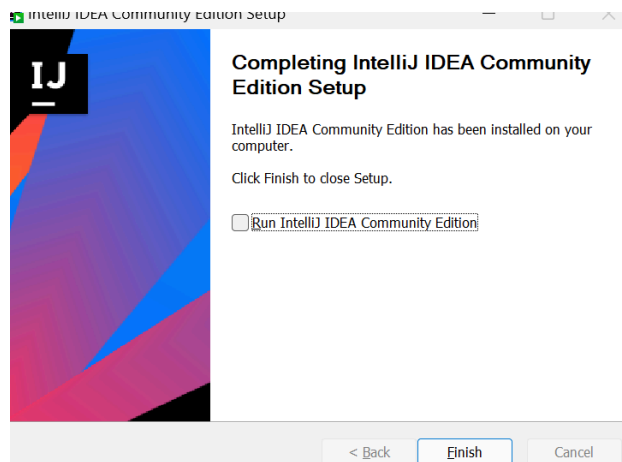
6. Укажите, что вам нужно создать ярлык на рабочем столе, а также обеспечить возможность открытия папок как проектов IntelliJ IDEA (синие отметки). **Next**



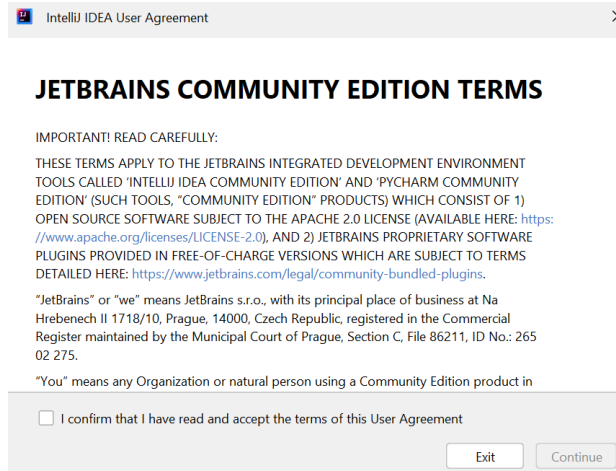
7. Нажимаем **Install**. Дождитесь окончания установки



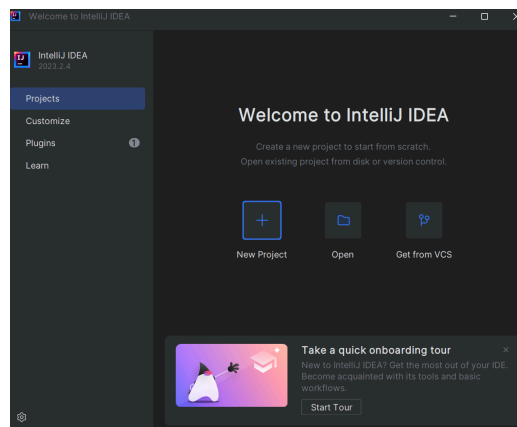
8. Нажимаем **Finish**. Если установить галочку “Run ...” - IntelliJ IDEA запустится сразу после закрытия окна. Также ее можно запустить с помощью значка на рабочем столе.



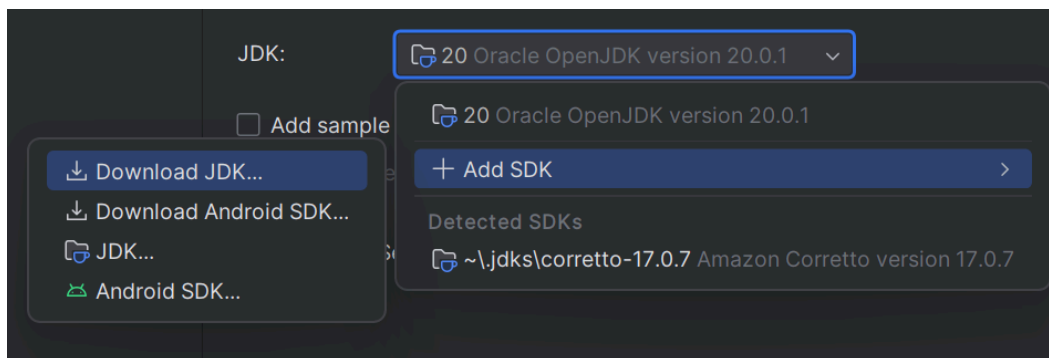
9. Ставим галочку “I confirm...” и нажимаем **Continue**



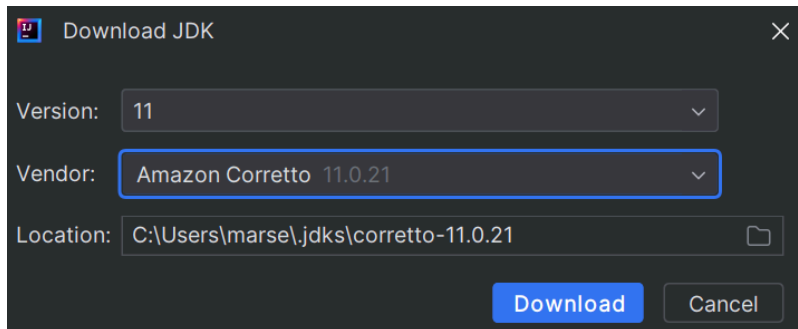
10. Нажимаем кнопку **New Project**



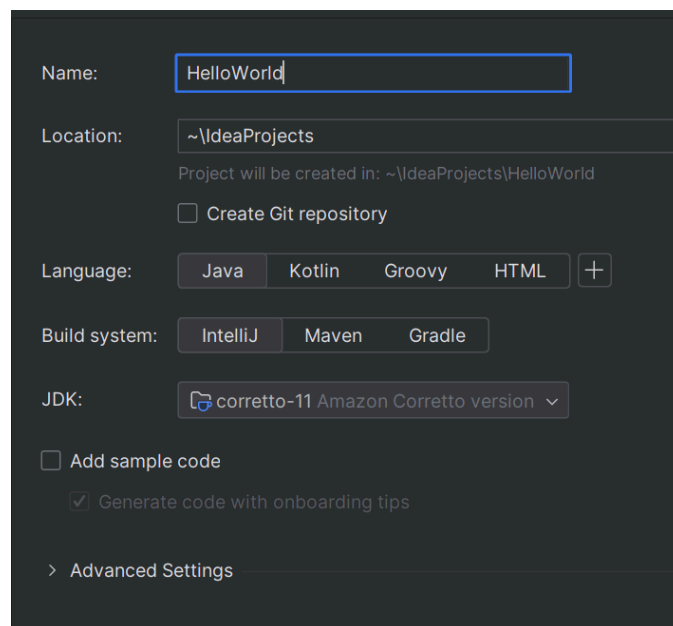
11. В появившемся окне находим пункт **JDK**, выбираем **Add SDK**, затем **Download JDK**



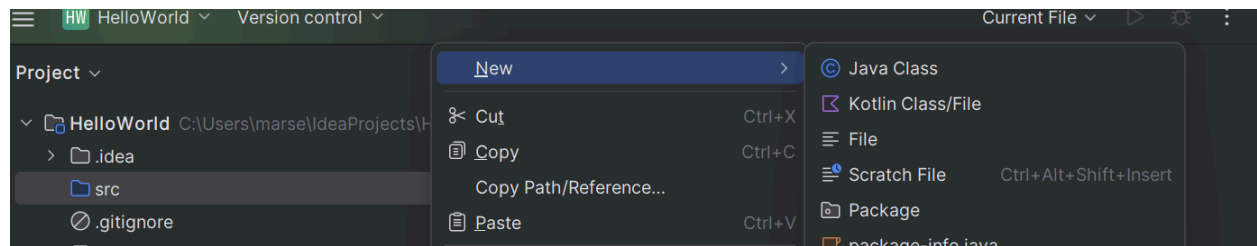
12. Выбираем версию **11** и **Amazon Corretto**. Нажимаем **Download**



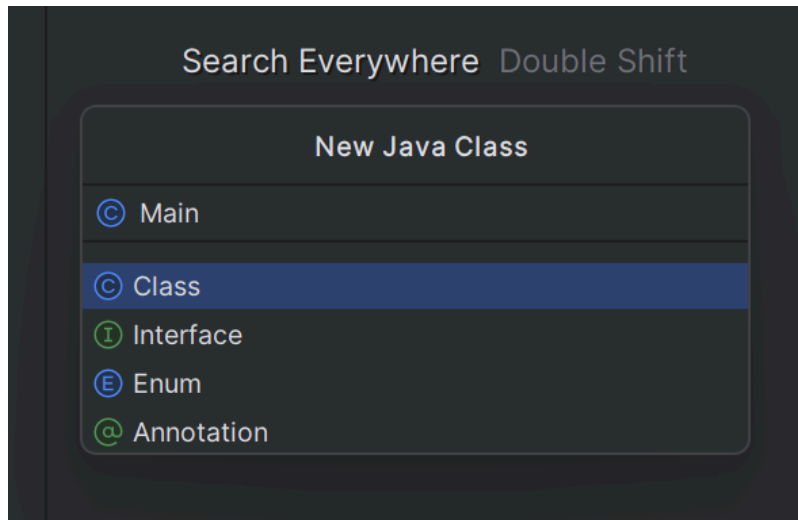
13. В строку **Name** необходимо вписать **HelloWorld**, убрать галочку “**Add sample code**”, далее нажимаем синюю кнопку **Create**



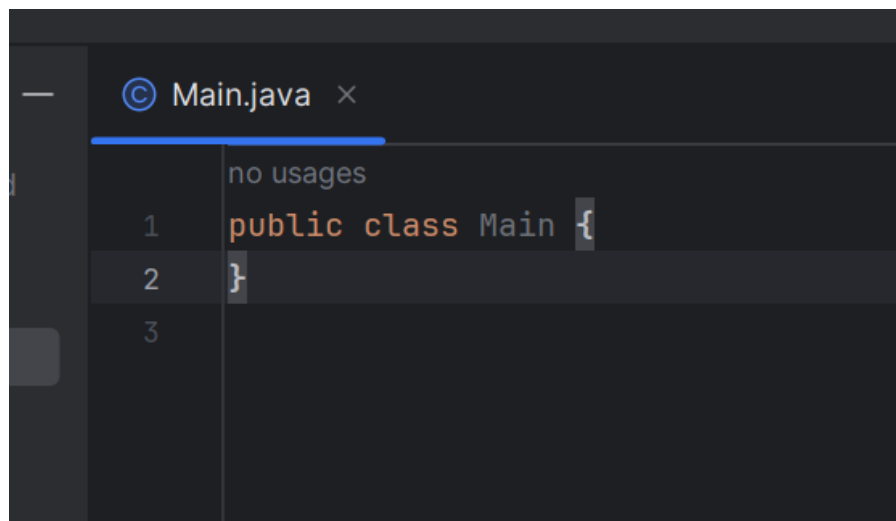
14. В левой части экрана находим синюю папку **src**, нажимаем правой кнопкой мыши, выбираем **New**, затем **Java Class**



15. В появившемся окне вписываем **Main** и нажимаем **Enter** на клавиатуре



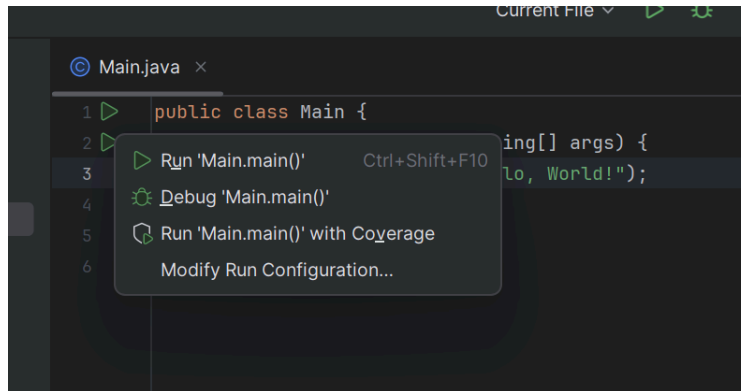
16. Мы создали файл, в котором следует писать наш код



Пример кода

```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

17. Необходимо нажать на любую из двух зеленых кнопок слева от **Main** и выбрать **Run 'Main.main()'**, получаем результат :)



Программа

Программа - это набор команд, позволяющих компьютеру решить какую-либо задачу и автоматизировать бизнес-процесс. При этом человек может написать программу на языке, понятном человеку (**язык программирования**). Компьютер же может выполнять программу на языке, понятном компьютеру (**машинный, бинарный код**). Следовательно, программа должна быть преобразована из кода, написанном на языке программирования, в бинарный код.

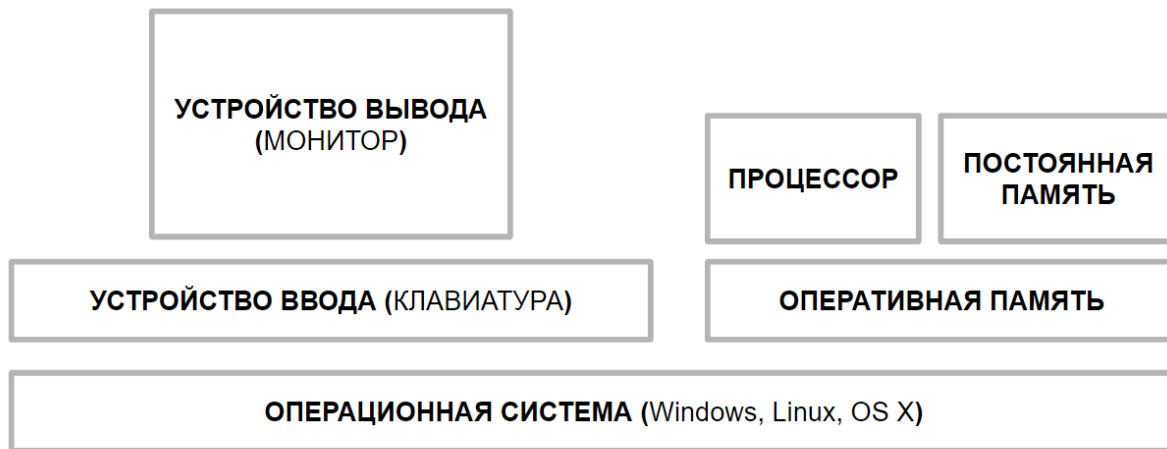
Важно помнить, что программа - это НЕ алгоритм, программа - это реализация алгоритма на конкретном языке программирования

Языки программирования

Рассмотрим примеры и отличительные особенности языков программирования:

- **Pascal, Basic** - подходят для обучения программированию
- **C/C++** - системное программное обеспечение (операционные системы, драйвера, ПО для роботов)
- **Python** - машинное обучение, обработка данных, прикладное ПО (как правило, стартапы)
- **PHP** - веб-сайты
- **JavaScript** - чаще всего используется для реализации браузерной логики в веб-приложениях
- **C#/Java** - универсальные языки программирования, используются для реализации Enterprise/корпоративных систем (высокие нагрузки, постоянные изменения, сложные бизнес-процессы, большое количество пользователей)

Архитектура компьютера



Процессор является основным "мозгом" компьютера, обрабатывая данные и выполняя задачи. **Оперативная память** представляет собой быструю, но малую по объему память, которая используется для временного хранения активных данных и программ. Она **зависит от энергоснабжения** и очень важна для быстрой работы компьютера.

Постоянная память, например жесткий диск или SSD, предоставляет **медленное** по сравнению с оперативной памятью, но более вместительное хранилище данных. Она является **энергонезависимой** и используется для долгосрочного хранения файлов и операционных систем.

Всеми устройствами компьютера управляет **операционная система**, которая обеспечивает взаимодействие между аппаратным обеспечением и пользовательскими программами, обеспечивая удобное и эффективное использование компьютера.