Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г. В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность 09 02 07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА

ПО «РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»

Страниц: 13

|  |  |
| --- | --- |
| Выполнил студент  Группы П50-6-21  Стукалов Илья Олегович | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_ К. А. Дзюба  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 года |

**Тема**: Введение в Java.

**Цель работы**: научиться устанавливать инструменты разработки и получить базовые навыки работы с Java. Написать калькулятор.

**Ход работы**:

1. Установим среду разработки IntelliJ IDEA

­­Легче и удобнее всего это сделать с помощью JetBrains Toolbox. В этом приложении есть все инструменты от JetBrains, которые также можно установить. Заходим на сайт <https://www.jetbrains.com/toolbox-app/> и скачиваем Toolbox.

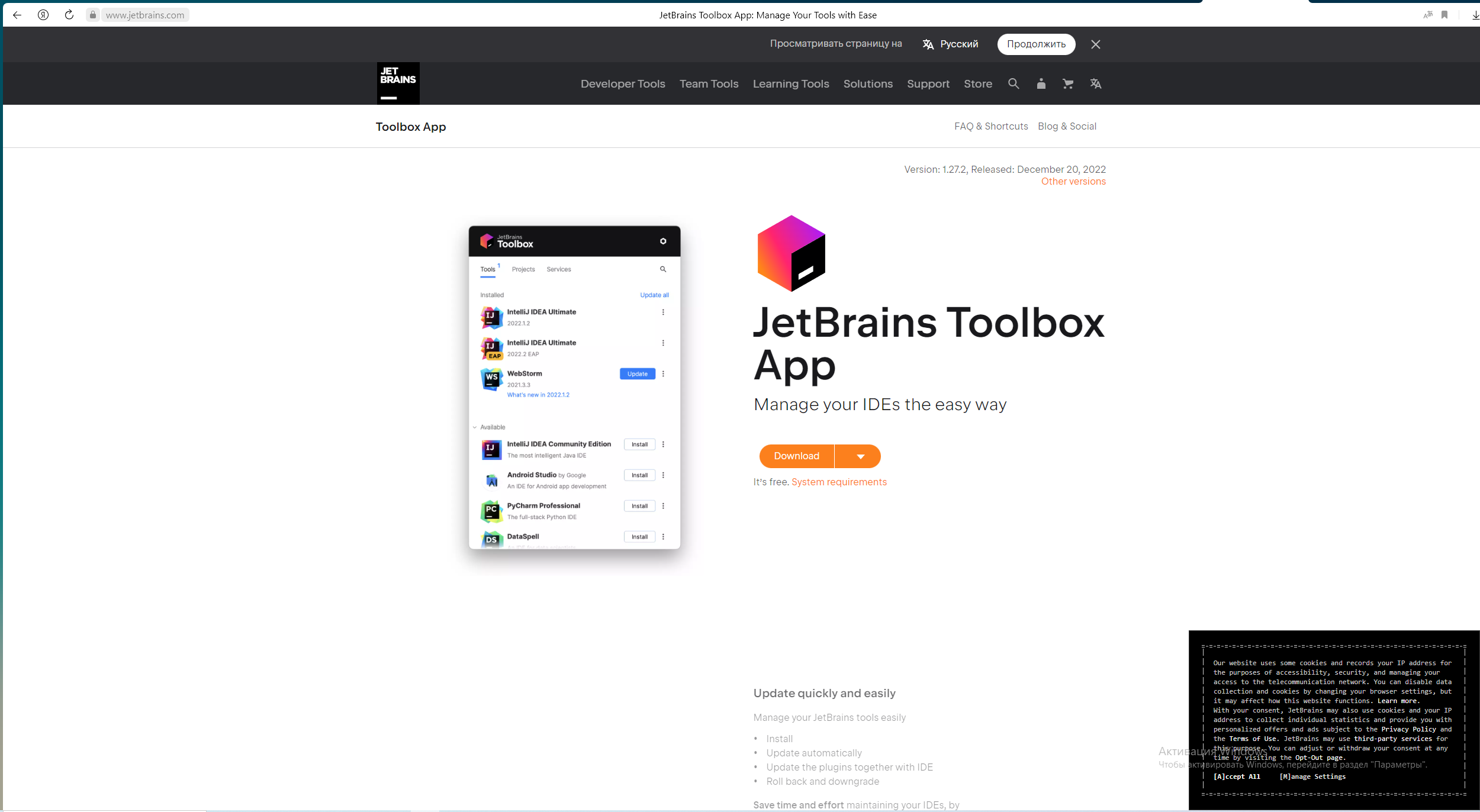


Рис. 1. Сайт Toolbox.

Далее просто открываем установщик и нажимаем установить. Все, Toolbox скачан на компьютер.

1. Создадим проект для реализации простого калькулятора.

Для этого откроем Toolbox и запустим IntelliJ IDEA.

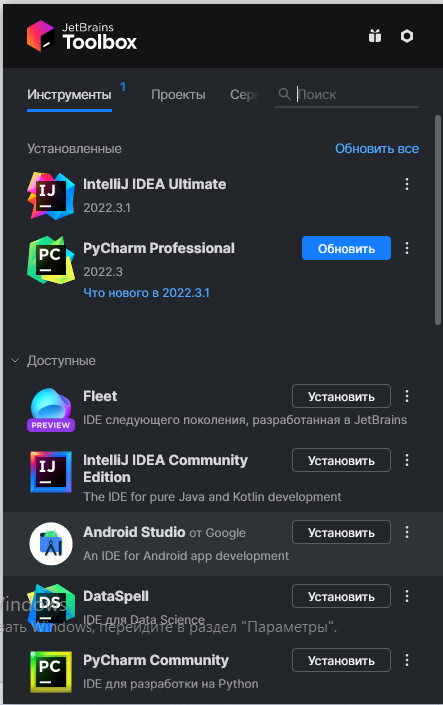


Рис. 2. Интерфейс Toolbox.

Перед нами появляется приветственное окно. Нажимаем на кнопку New Project

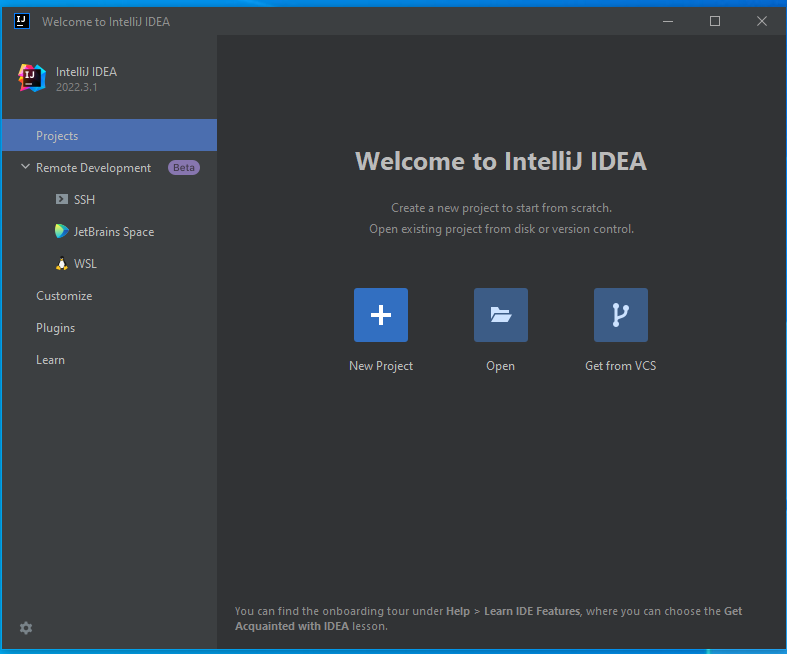


Рис. 3. Приветственное окно IntelliJ IDEA.

Далее перед нами появляется окно с настройкой проекта. Нам нужно хорошо подумать над названием, а также обязательно скачать JDK, который необходим для компиляции и запуска Java файлов, а также содержит стандартные библиотеки. Если у вас нет JDK, то здесь же можно его скачать. В выпадающем меню нажимаем Download JDK и перед нами появляется окно с выбором JDK. На данный момент вендор для нас не важен. Версию желательно скачивать одну из последних. У меня уже скачан JDK 17.

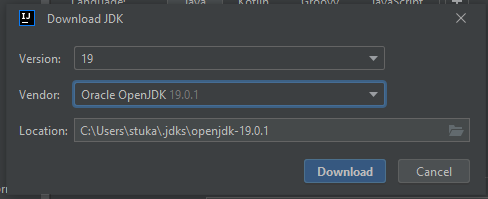


Рис. 4. Окно настройки JDK.

Выбираем путь до проекта. Чекбокс «Add sample code» отвечает за добавление в проект Main класса с main методом. Я оставлю галочку.

Далее просто нажимаем Create.

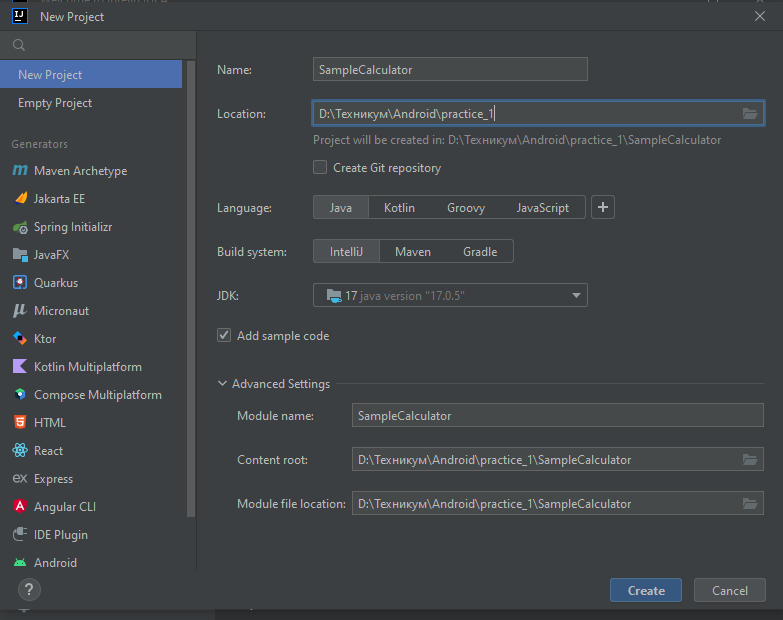


Рис. 5. Окно с настройкой проекта для создания.

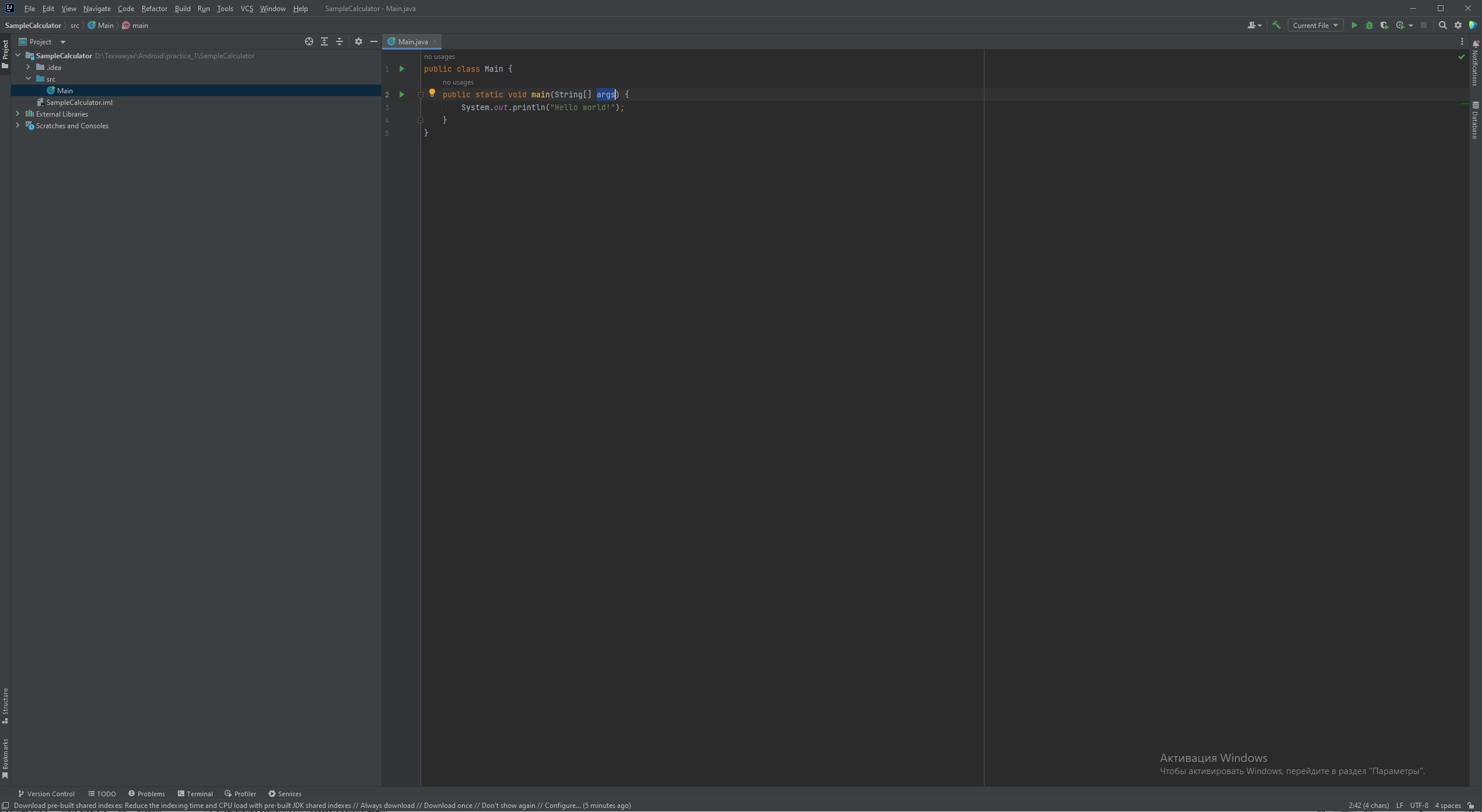
Перед нами появилась рабочая область с уже открытым файлом Main. Тут мы видим метод main, который будет точкой входа для программы. Он обязательно должен принимать массив строк как аргументы командной строки.

Рис. 6. Рабочая область IntelliJ IDEA.

Далее нам необходимо создать и настроить конфигурацию для запуска программы. В правом верхнем углу (рядом с кнопкой запуска программы) нажимаем на Current file и в выпадающем списке нажимаем Edit configurations. Перед нами появляется такое окно.

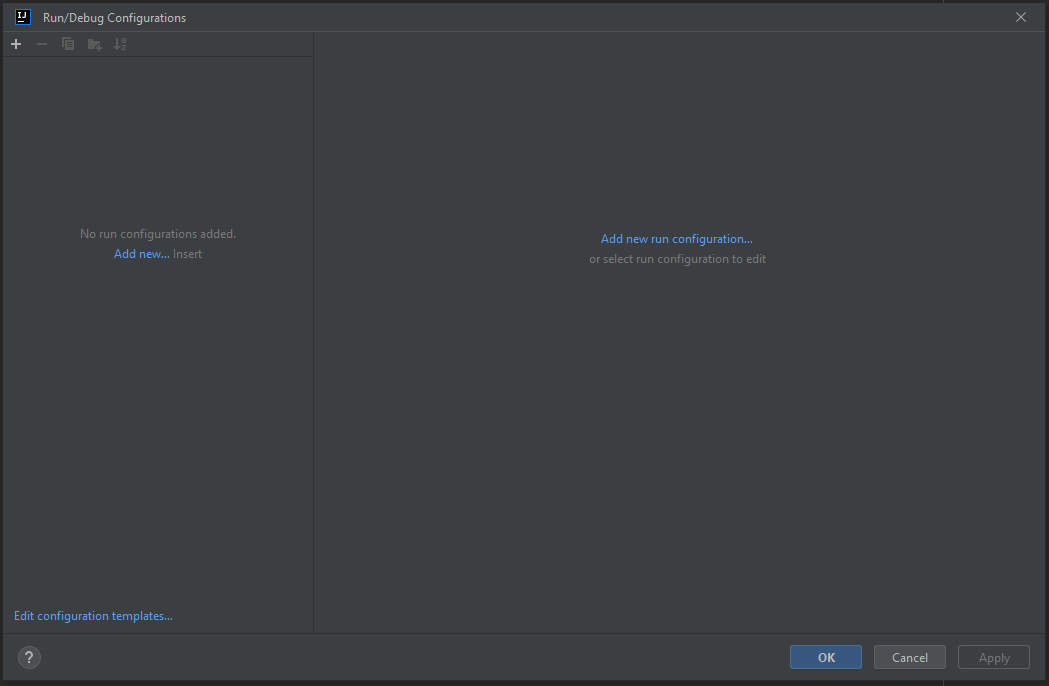


Рис. 7. Окно редактирования конфигураций.

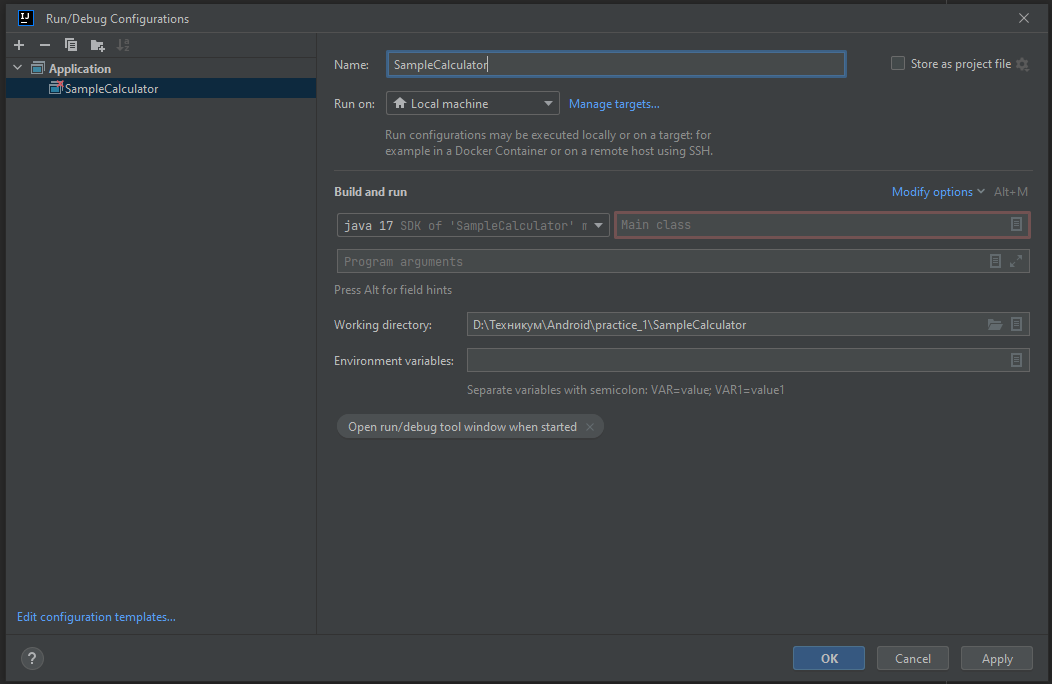
Нам нужно нажать на плюсик в левом верхнем углу и выбрать там Application. Создается новая ненастроенная конфигурация. Здесь мы можем назвать её как хотим, выбрать для неё JDK, выбрать Main класс и много другого. Я назвал конфигурацию также, как и проект – SampleCalculator. Для него я выбрал JDK 17, который выставляется по умолчанию, если был установлен для этого проекта. 

Рис. 8. Настройка конфигурации.

Далее нам нужно выбрать Main класс. Слева от поля с надписью Main class нажимаем на кнопку и появляется такое окно:

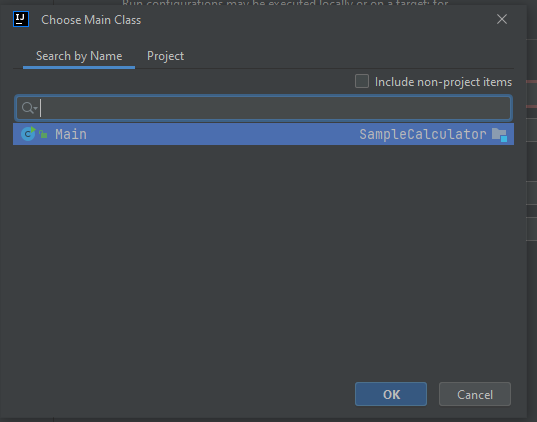


Рис. 9. Выбор Main класса.

IntelliJ уже за нас нашла Main класс, так как мы реализовали в нем функцию main. Также в правой вкладке можно вручную выбрать Main класс.

Далее нажимаем OK и в предыдущем окне тоже OK. Теперь мы можем запустить программу, нажав на зеленую кнопку Run в правом верхнем углу. В консоли снизу нам вывелось то, что мы передали в метод println.

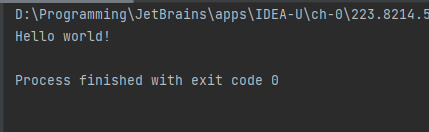


Рис. 10. Первый запуск.

1. Написание калькулятора.

Для начала создадим глобальную переменную inScanner. inScanner нужен для ввода данных из консоли.



Рис. 11. Сканер для ввода данных из консоли.

Для начала сделаем метод, который будет из String пробовать получить int. Возвращать он будет полученное число, а также по ссылке “возвращает” результат получения числа (true или false).

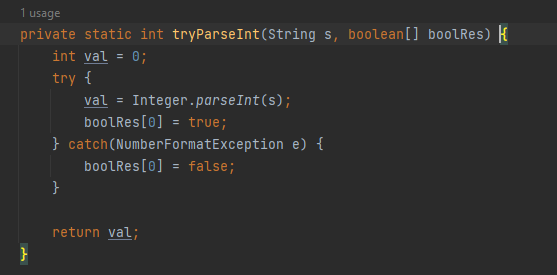


Рис. 12. Реализация метода tryParseInt

Затем сделаем метод для получения числа от пользователя. При этом ввод от пользователя будет запрашиваться до тех пор, пока пользователь не введет число. Как раз для этого нам нужен boolean результат от метода tryParseInt.

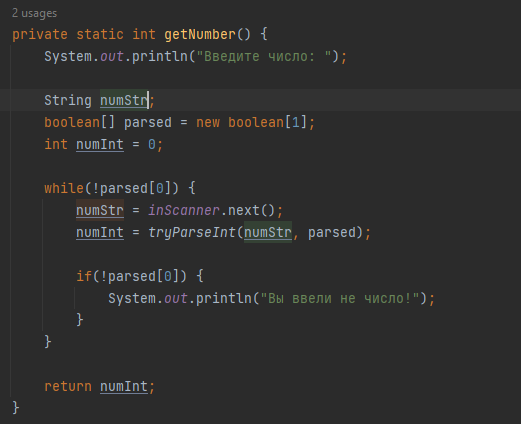


Рис. 13. Реализация метода getNumber.

Осталось реализовать main метод. Для начала выводим информацию о программе. Далее в бесконечном цикле сначала запрашиваем от пользователя первое число, затем знак, затем второе число. У ввода для знака есть проверка, что он является поддерживаемым знаком. Ввод для знака запрашивается, пока пользователь не введет один из поддерживаемых знаков. Далее происходит расчет результата. Затем выводится результат.

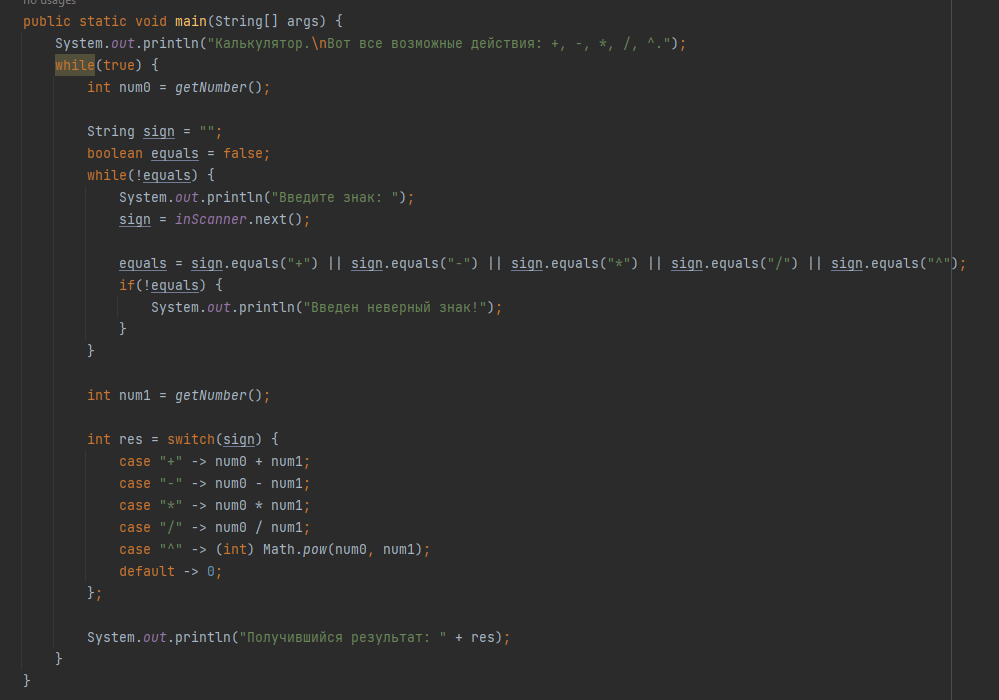


Рис. 14. Реализация main метода.

**Вывод**: научились устанавливать инструменты разработки и получили базовые навыки работы с Java. Написали калькулятор.