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

ПРАКТИЧЕСКИЕ РАБОТЫ ПО ДИСЦИПЛИНЕ

«РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»

Листов: 12

|  |  |
| --- | --- |
| Выполнил студент  Группы: П50-4-21  Игошев Ростислав Вадимович | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_К. А. Дзюба  «\_\_\_\_\_\_» \_\_\_\_\_\_2023 года |

Москва 2023

# ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: «Введение в Java»

Цель: Выбрать подходящую виртуальную среду для заданных целей, установить виртуальную среду для языка Java, настроить среду и создать проект с работающим кодом.

В качестве виртуальной среды было принято решение о выборе IntelliJ IDEA от компании Jet Brains.

Для скачивания заходим на сайт компании, находящийся по адресу: <https://www.jetbrains.com/idea/download/#section=windows>.

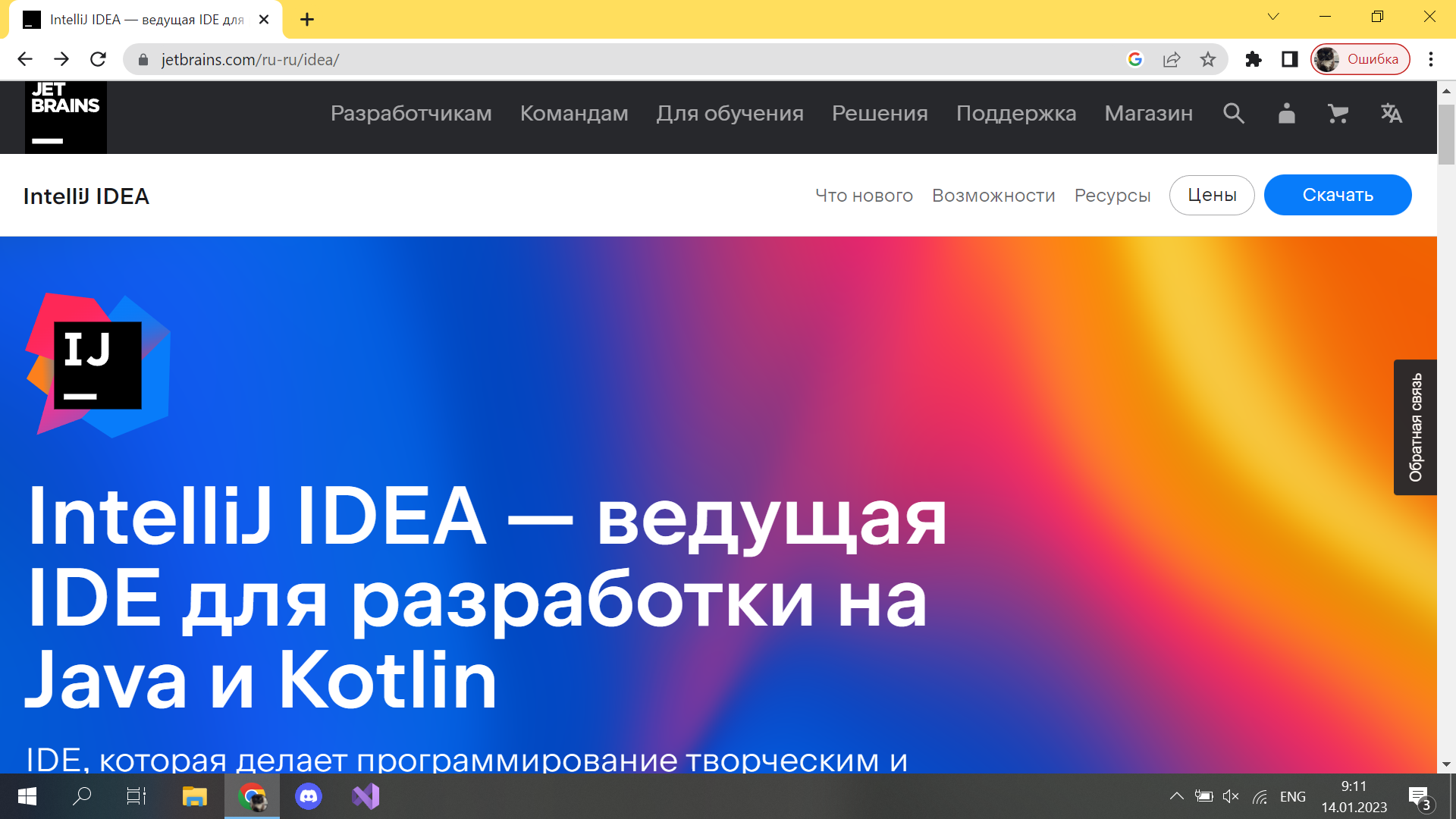


Рисунок 1 – Веб-страница среды разработки

После, нажимаем на кнопку скачивания для перехода на страницу скачивания файла.

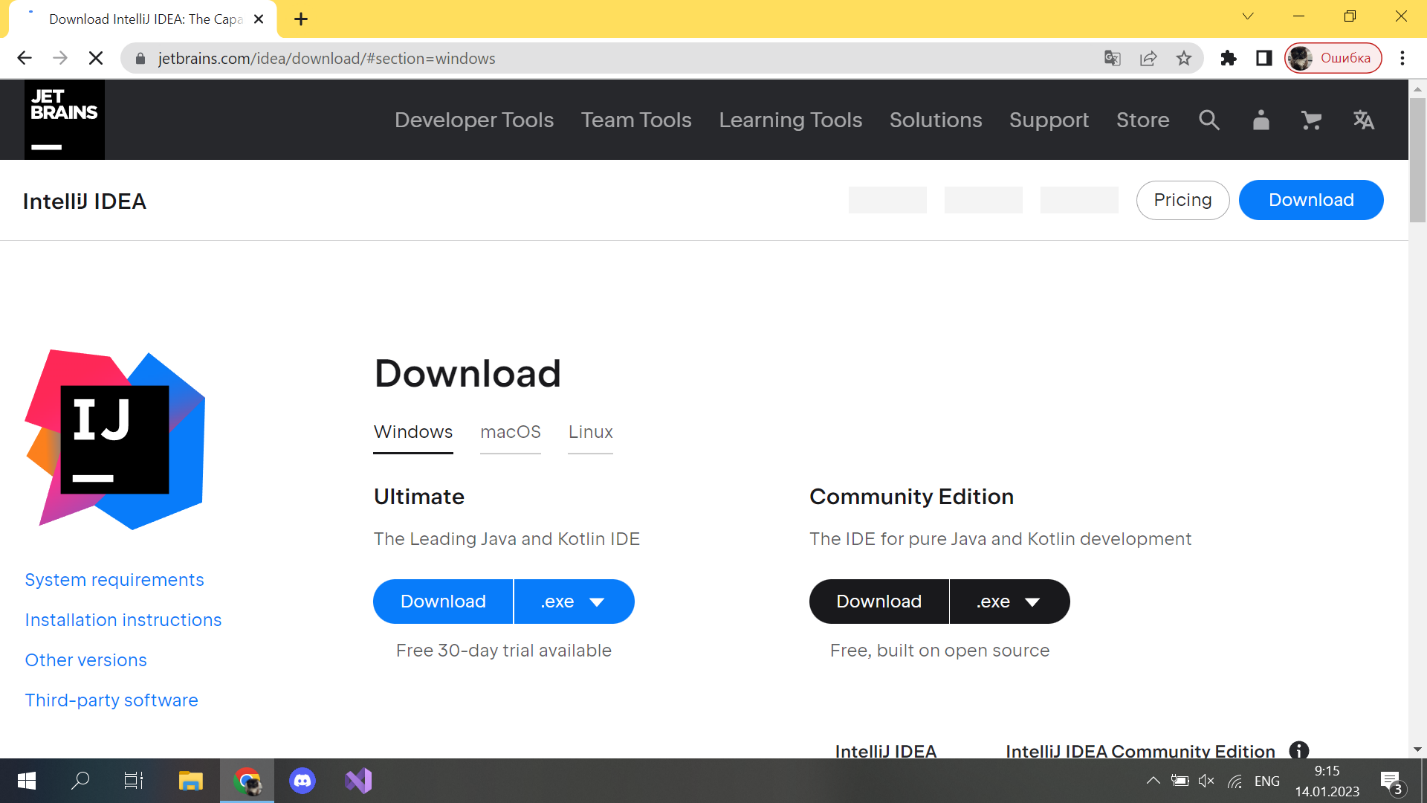


Рисунок 2 – Страница скачивания

На сайте выбираем нашу ОС и нажимаем кнопку Download. Скачивать Ultimate или Community Edition – ваш выбор.

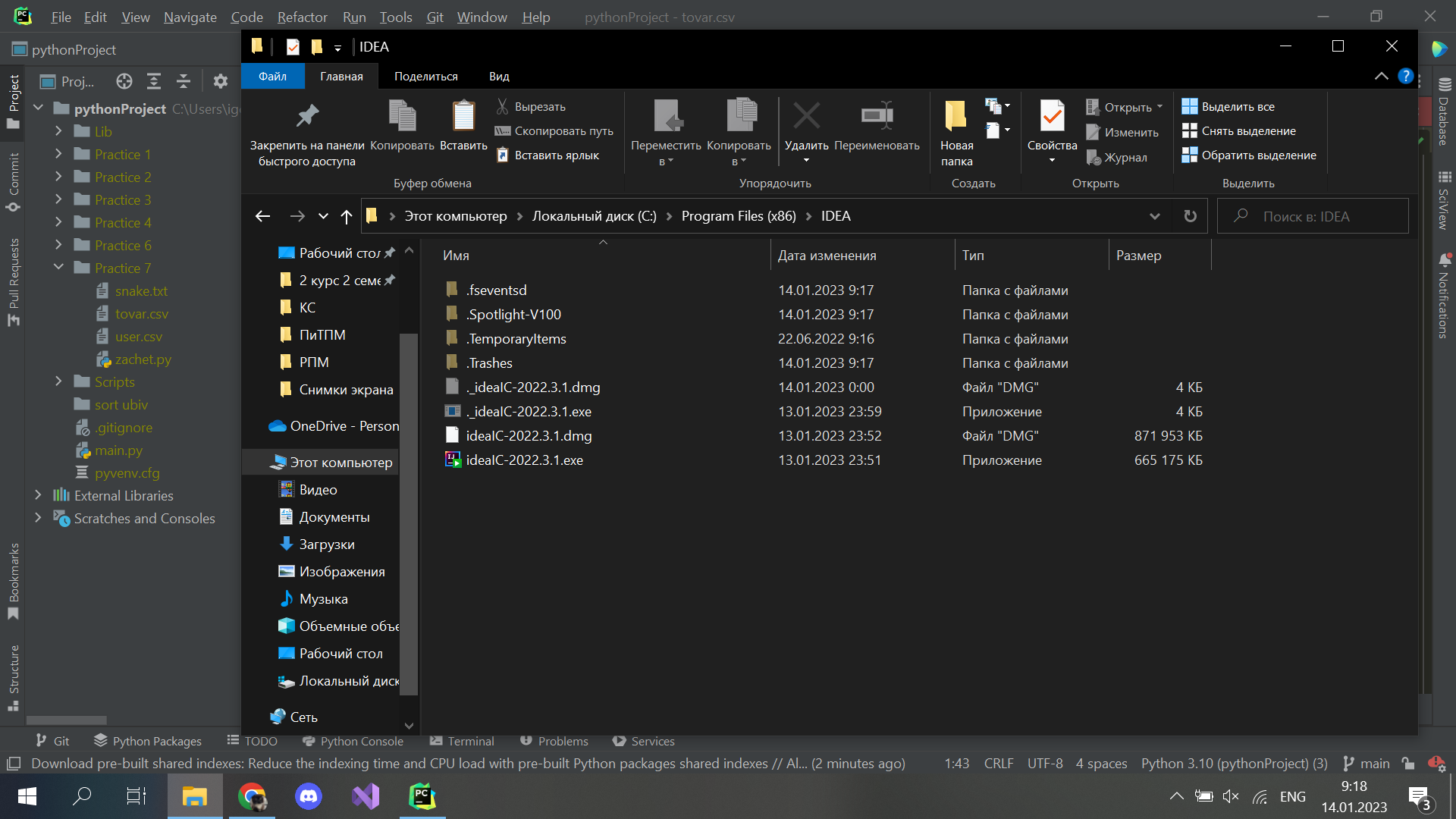


Рисунок 3 – Скачанные файлы

После открытия скачанных файлов видим следующий список. В нём необходимо запустить файл установки с расширением .exe и установить программу.

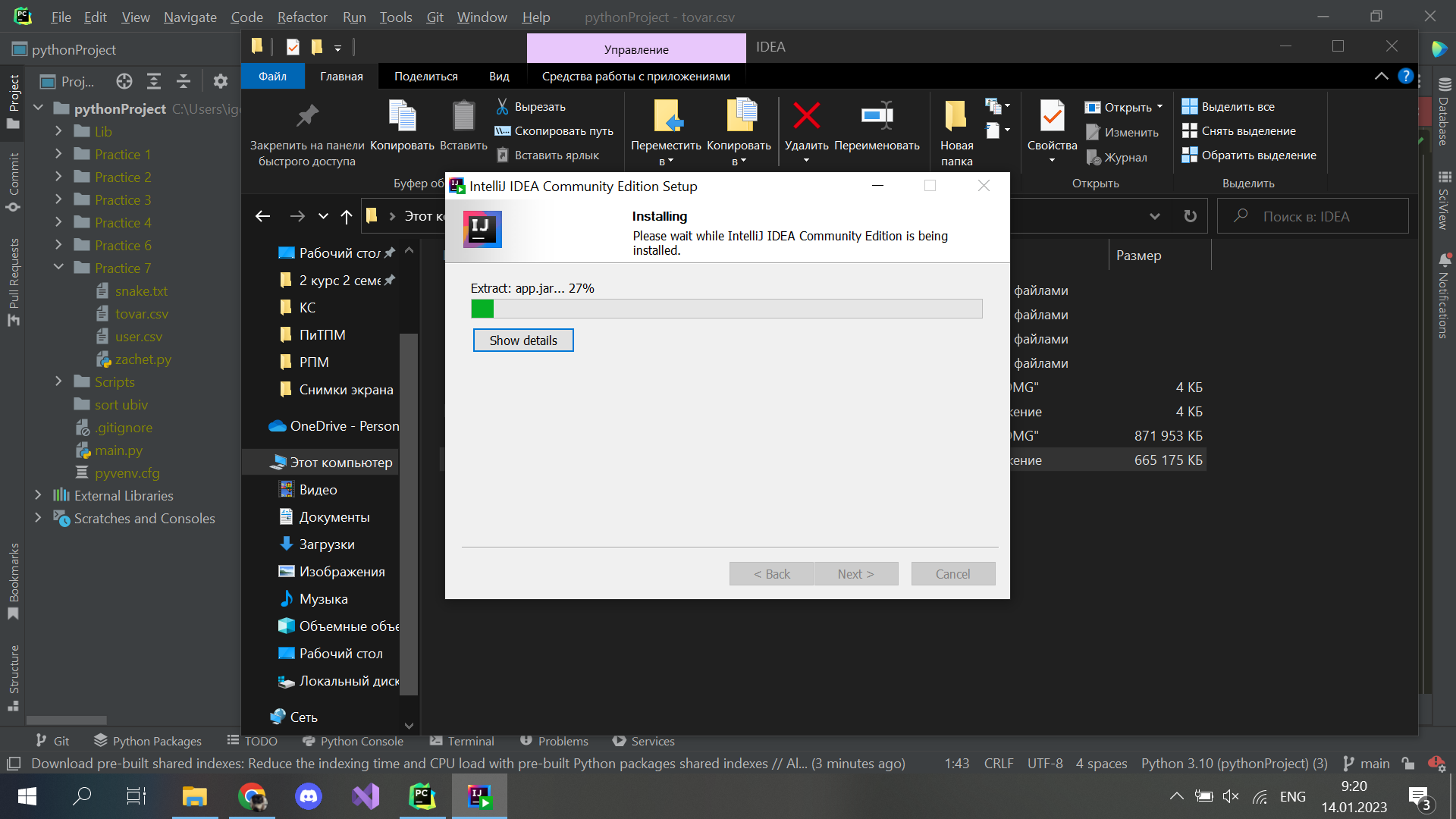


Рисунок 4 – Установка программы

После установки запускаем программу и видим меню с возможностью создать проект, чем мы и воспользуемся.

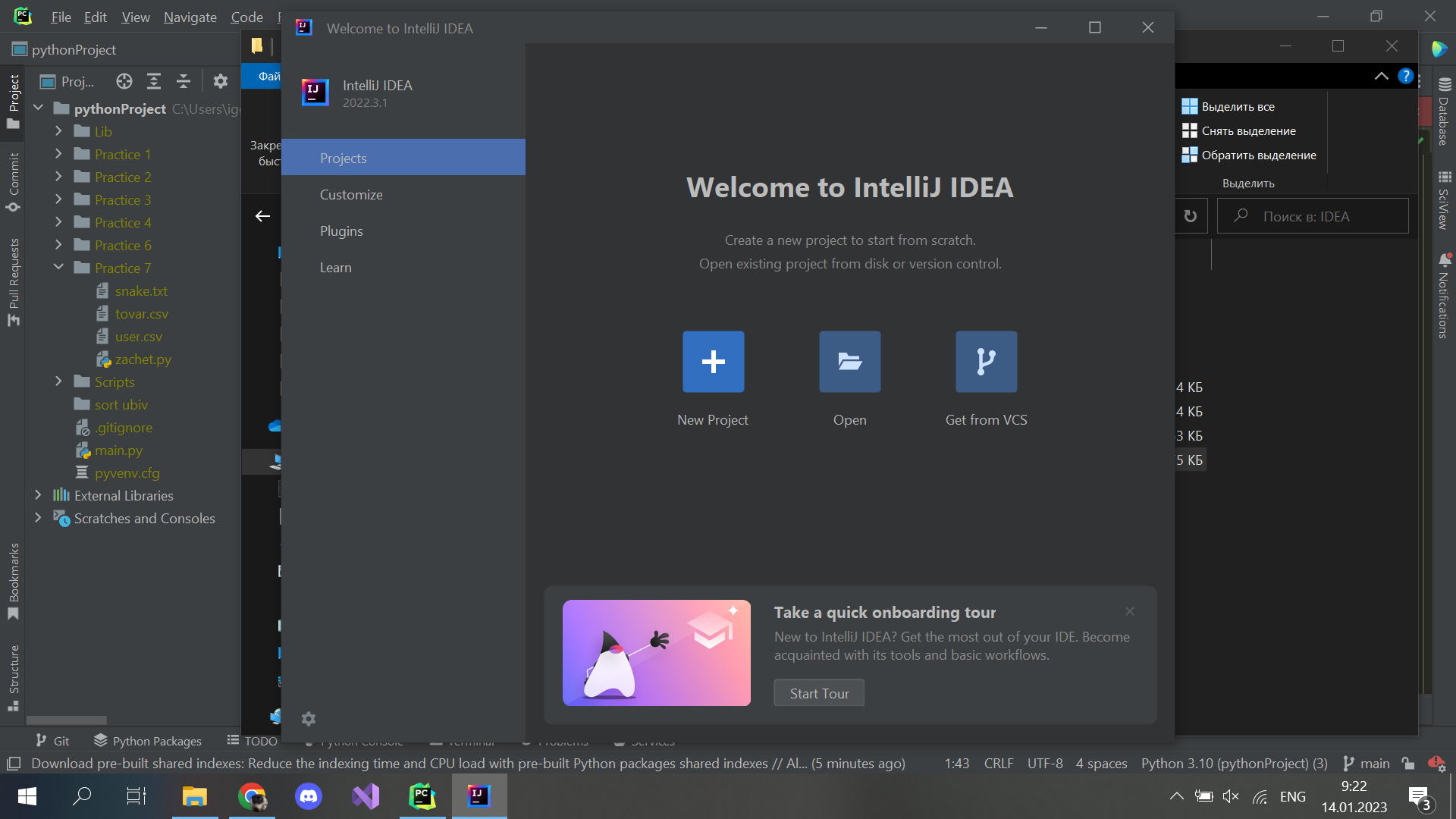


Рисунок 5 – Запущенная программа

Нажав на кнопку, перед нами возникают настройки будущего проекта. В качестве языка указываем Java, в качестве системы сборки – IntelliJ. Остальное – на ваше усмотрение.

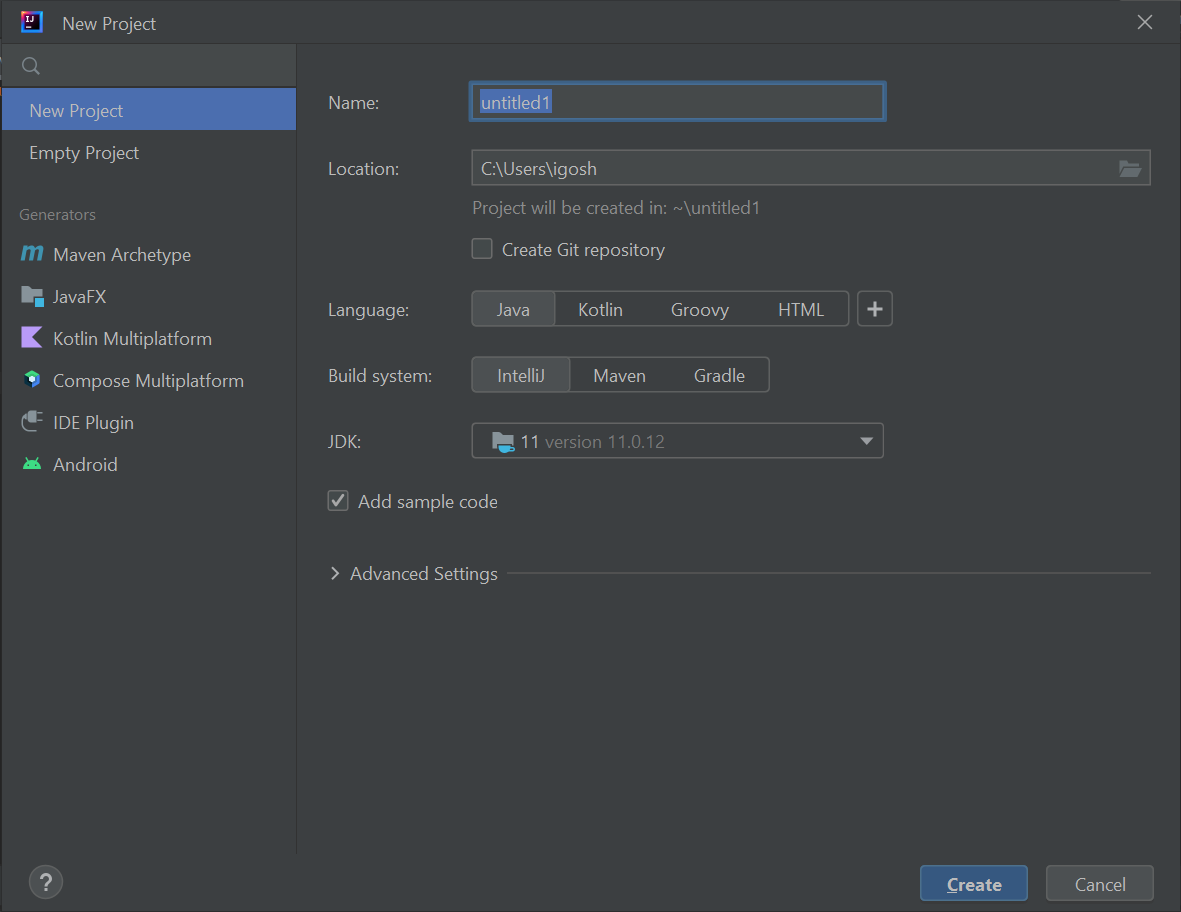


Рисунок 6 – Настройки программы

Создав проект, попробуем написать небольшую программу. В примере созданы несколько переменных с разными типами данных, выведены некоторые тестовые сообщения, проведены математические расчеты, реализован пользовательский ввод, условие с расчетами и выполнение действий в случае невыполнения условия.

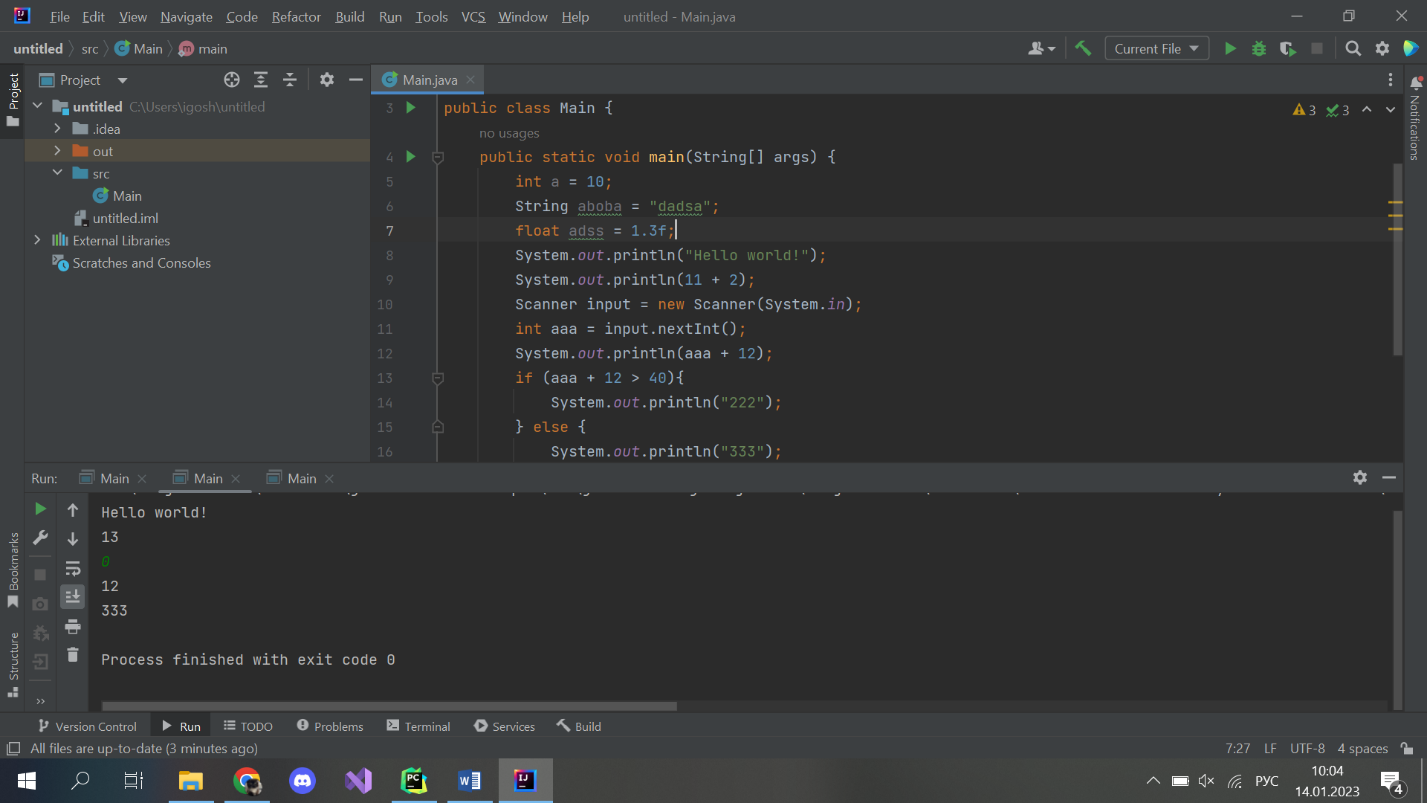


Рисунок 7 – Созданная мини-программа

Далее, для выполнения задания, создадим консольный калькулятор.

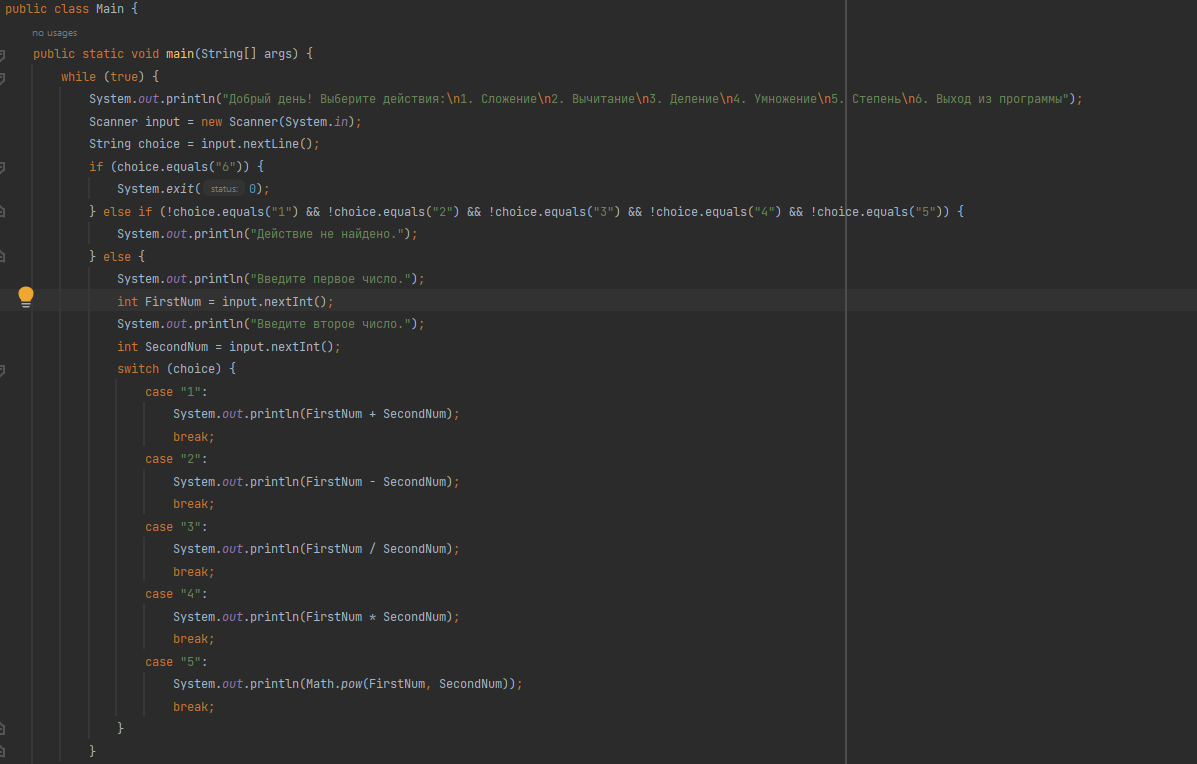


Рисунок 8 – Консольный калькулятор

В калькуляторе использовались переменные типа String и Integer, пользовательский ввод, считывание ввода, условия, математические уравнения, выход из программы.

Результаты работы программы:

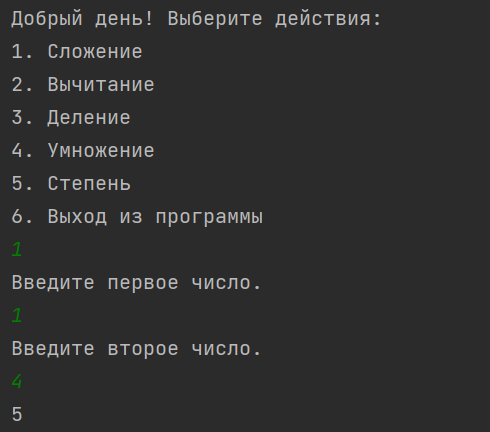


Рисунок 9 – Результат работы сложения

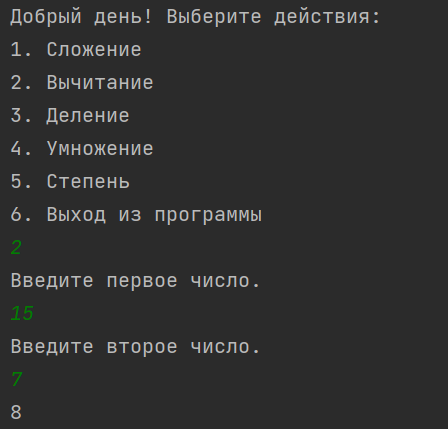


Рисунок 10 – Результат работы вычитания

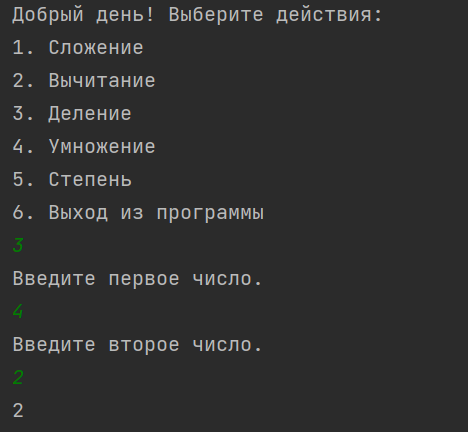


Рисунок 11 – Результат работы деления



Рисунок 12 – Результат работы умножения

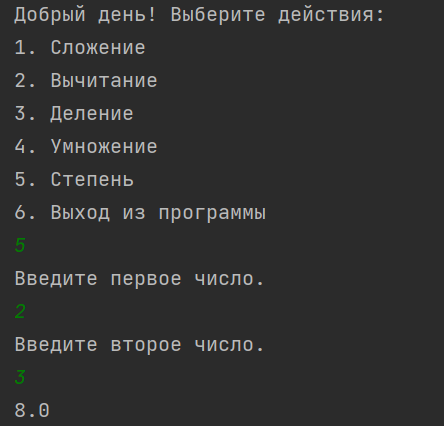


Рисунок 13 – Результат работы возведения в степень

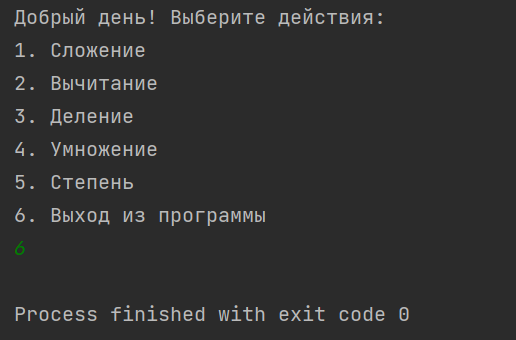


Рисунок 14 – Выход из программы

Вывод: Выбрали подходящую виртуальную среду для заданных целей, установили виртуальную среду для языка Java, настроили среду и создали проект c работающим кодом.

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: «Циклы»

Цель работы: научиться работать с циклами в языке программирования Java, выполнить пару задач на отработку изложенного материала.

Условия:

For.12: Дано целое число N (> 0). Найти произведение 1.1 · 1.2 · 1.3 · . . . (N сомножителей).

While.17: Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, вывести все его цифры, начиная с самой правой (разряда единиц).

Решение задач:

Было принято решение вынести две задачи в отдельные методы и выбирать между ними, запуская одну программу.



Рисунок 15 – Выбор задачи

Далее, приступаем к решению самих задач. Первая из задач For.12:

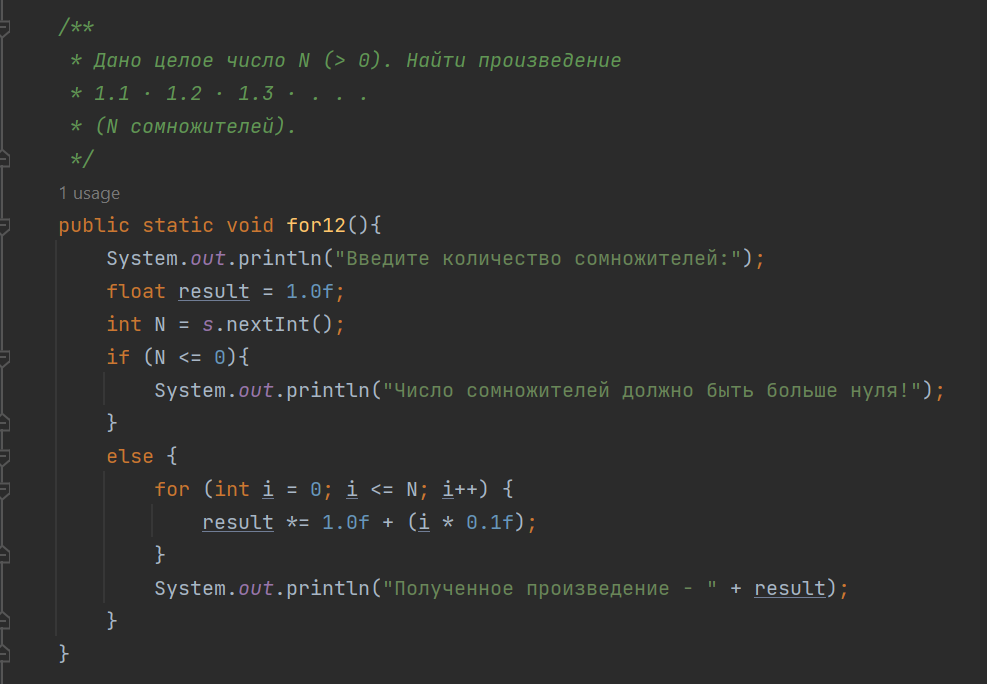


Рисунок 16 – For.12

Здесь был использован цикл for и алгоритмизация умножения.

Следующая задача While.17:

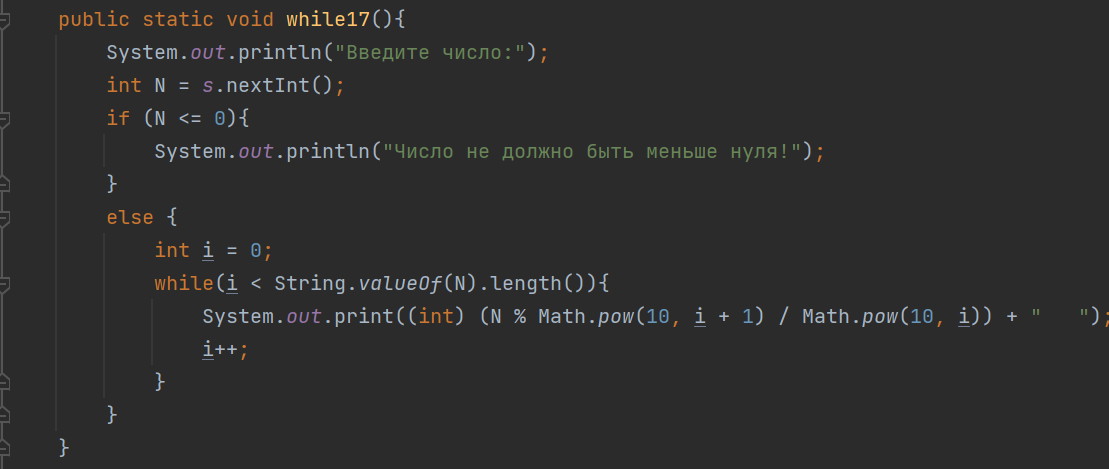


Рисунок 17 – While.17

Здесь же, был использован цикл while, деление без остатка и операция нахождения остатка от деления.

Результаты работы:

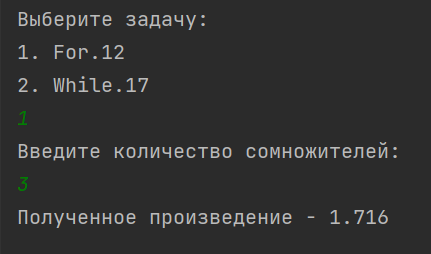


Рисунок 18 – For.12

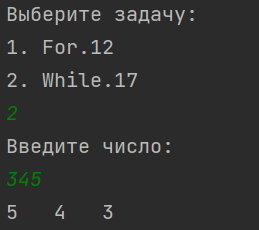


Рисунок 19 – While.17

Вывод: научились работать с циклами в языке программирования Java, выполнили пару задач на отработку изложенного материала.

ПРАКТИЧЕСКАЯ РАБОТА №3

Тема: «Массивы и матрицы»

Цель работы: научиться работать с массивами и матрицами, решить две задачи из задачника Абрамяна в группах Array и Matrix в соответствии со своим прямым и обратным номером.

Условия задач:

Array12: Array12. Дан массив A размера N (N — четное число). Вывести его элементы с четными номерами в порядке возрастания номеров: A2, A4, A6, . . ., AN Условный оператор не использовать.

Matrix17: Дана матрица размера M × N и целое число K (1 ≤ K ≤ M). Найти сумму и произведение элементов K-й строки данной матрицы.

Решения задач:

Было принято решение вынести две задачи в отдельные методы и выбирать между ними, запуская одну программу.



Рисунок 20 – Консольное меню для запуска задач

Array12:

Ниже вы видите код, с помощью которого была реализована программа:

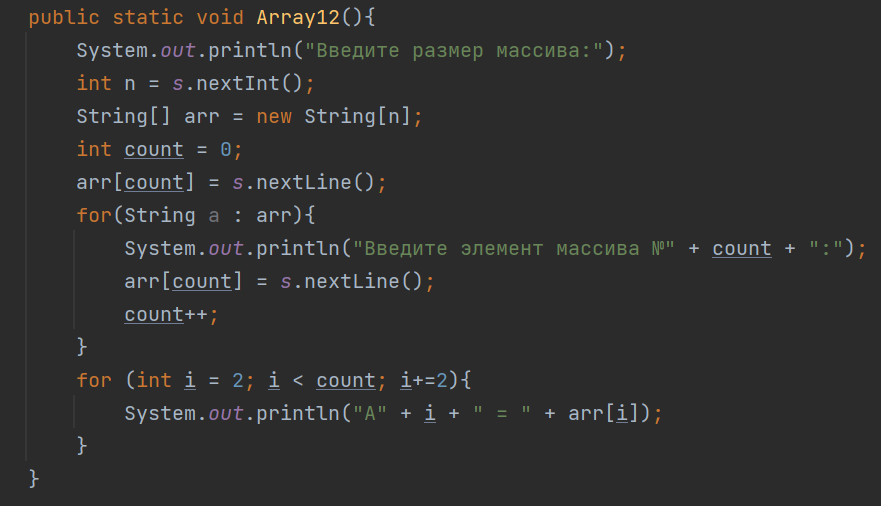


Рисунок 21 – Метод Array12

Здесь был применен цикл foreach для ввода пользователем элементов массива и здесь же был реализован счетчик его длины. Также, был реализован вывод значений четных индексов массивов.

Matrix17:

Ниже вы видите код, с помощью которого была реализована программа:

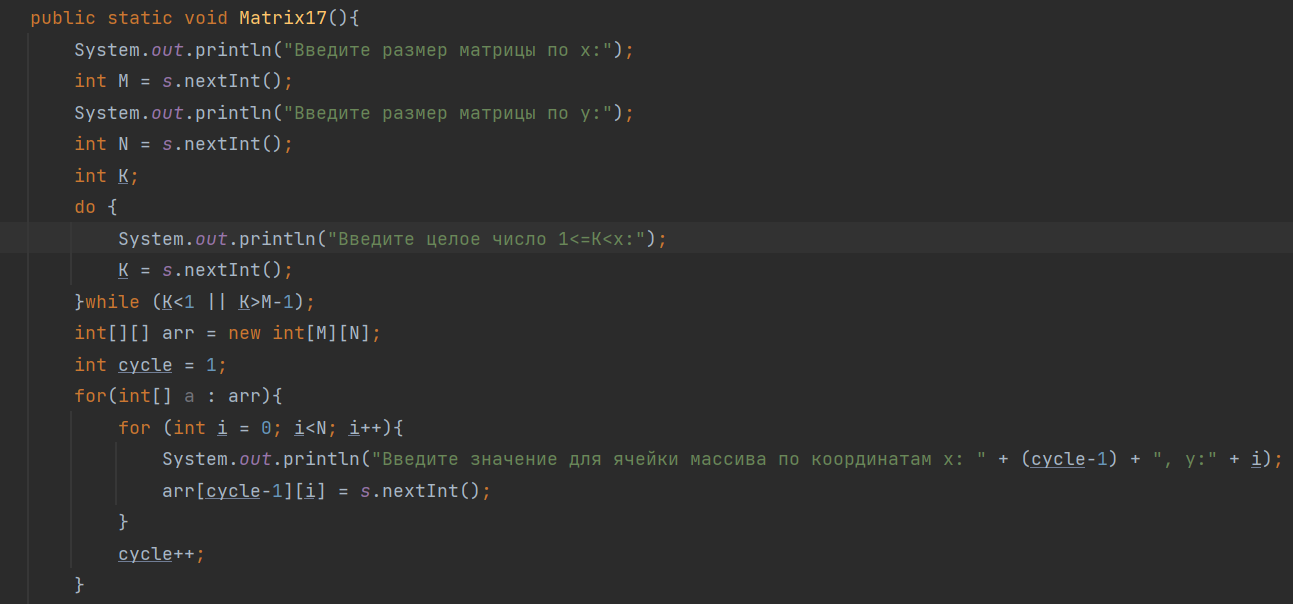


Рисунок 22 – Цикл с заполнением

Здесь был применен двумерный массив, цикл для заполнения массива, и цикл для расчета суммы, и произведения.

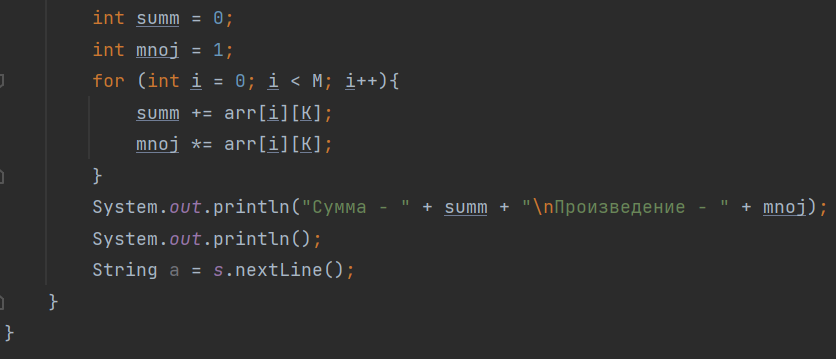


Рисунок 23 – Цикл с расчетом суммы и произведения

Результаты работы:

Пример работы консольного меню:

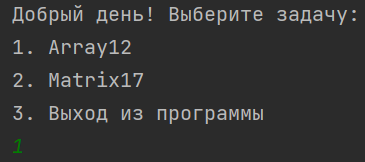


Рисунок 24 – Результат работы консольного меню

Array12:

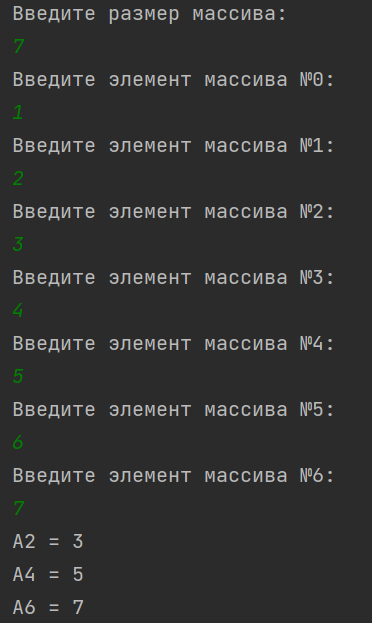


Рисунок 25 – Array12

Matrix17:



Рисунок 26 – Matrix17

Вывод: научились работать с массивами и матрицами, решили две задачи из задачника Абрамяна в группах Array и Matrix в соответствии со своим прямым и обратным номером.

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: «Методы и классы»

Цель работы: научиться работать с методами и классами в Java, описать не менее трех классов в соответствии с сущностями вашей предметной области.

Наш основной класс – Main. Из него будут запускаться все наши классы и их объекты. Класс тестирования работы с объектами выглядит следующим образом, включая в себя два метода:

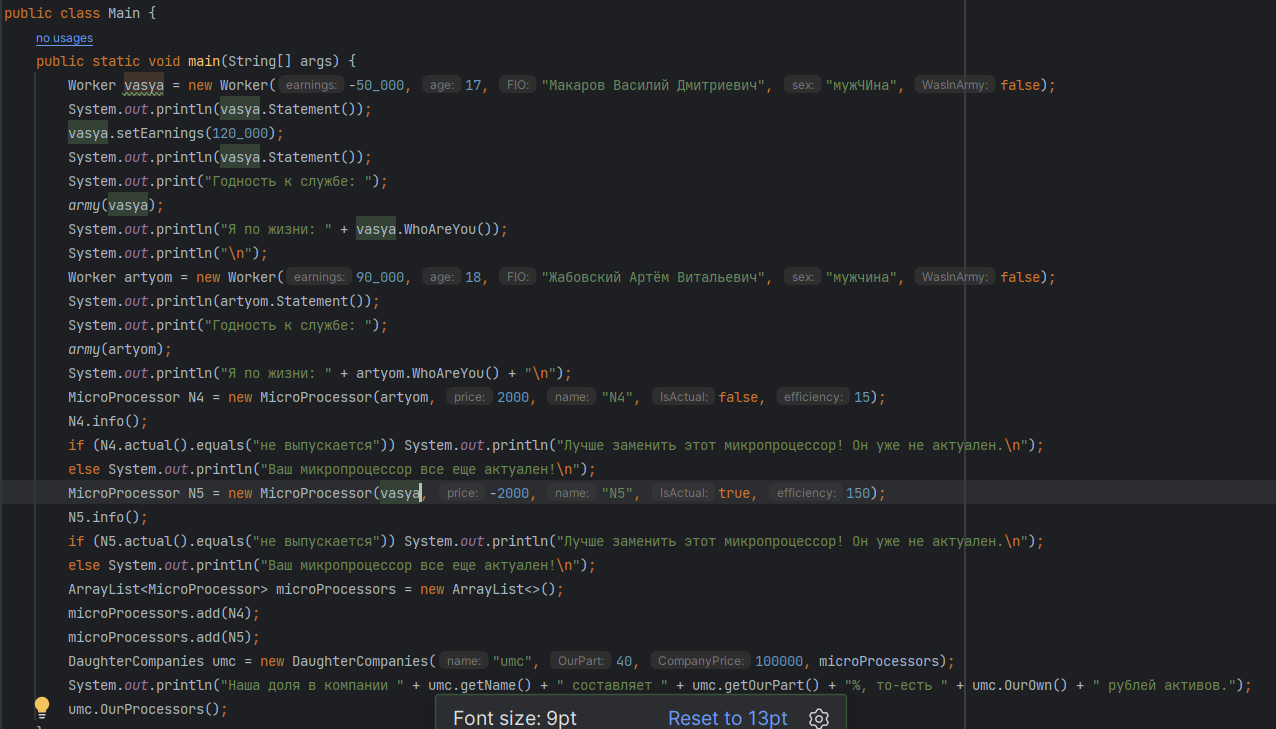


Рисунок 27 – Функция main класса Main

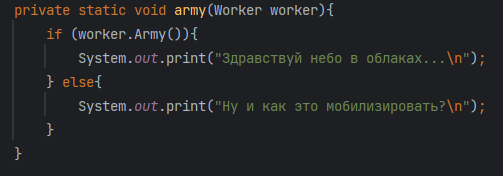


Рисунок 28 – Функция army класса Main

Теперь, реализуем первый класс. Он будет отвечать за сотрудников компании и содержать в себе несколько полей, методов, и конструкторов.

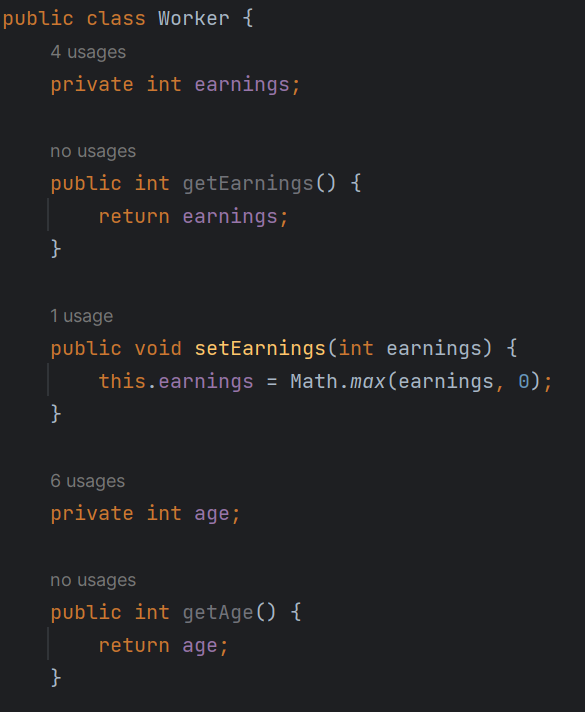


Рисунок 29 – Поля и get-set конструкции класса Worker



Рисунок 30 - Поля и get-set конструкции класса Worker

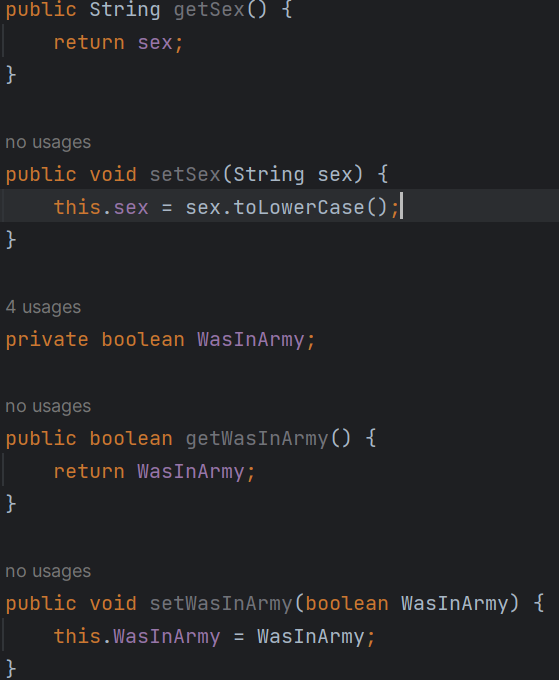


Рисунок 31 - Поля и get-set конструкции класса Worker

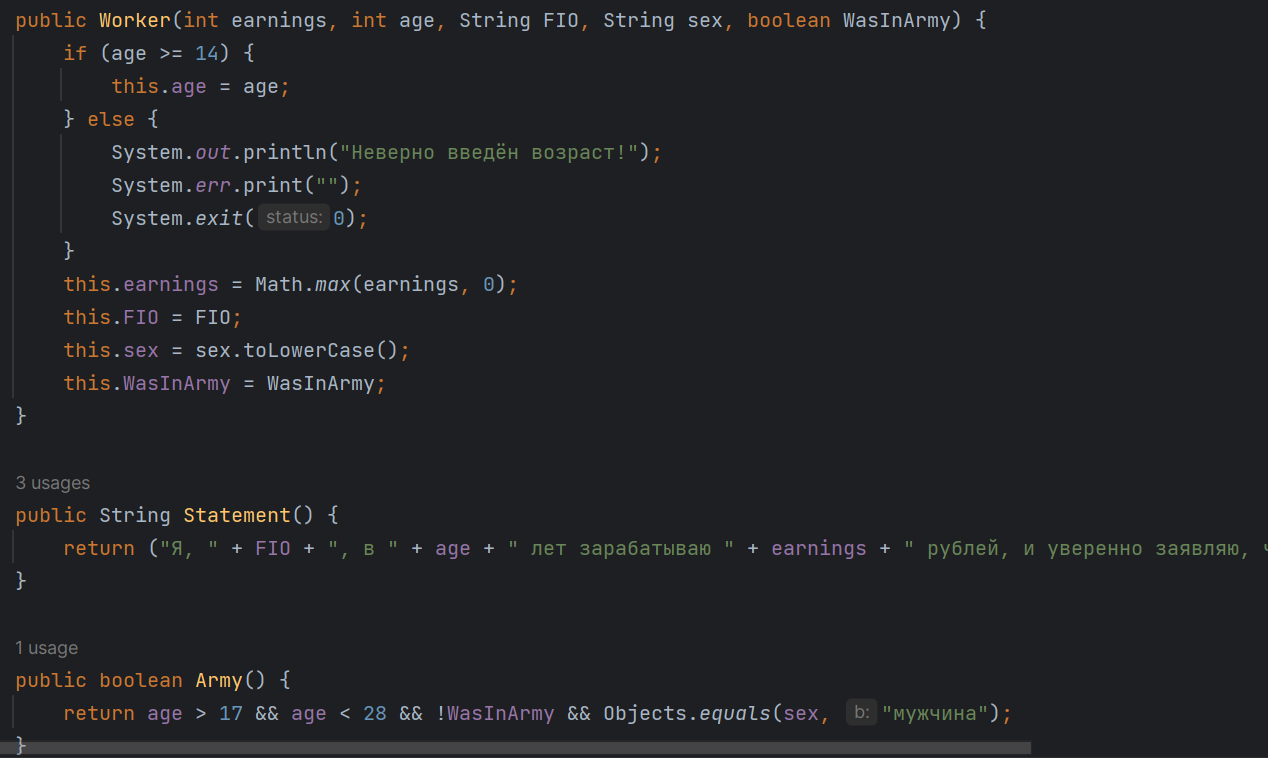


Рисунок 32 – Конструктор и методы класса Worker

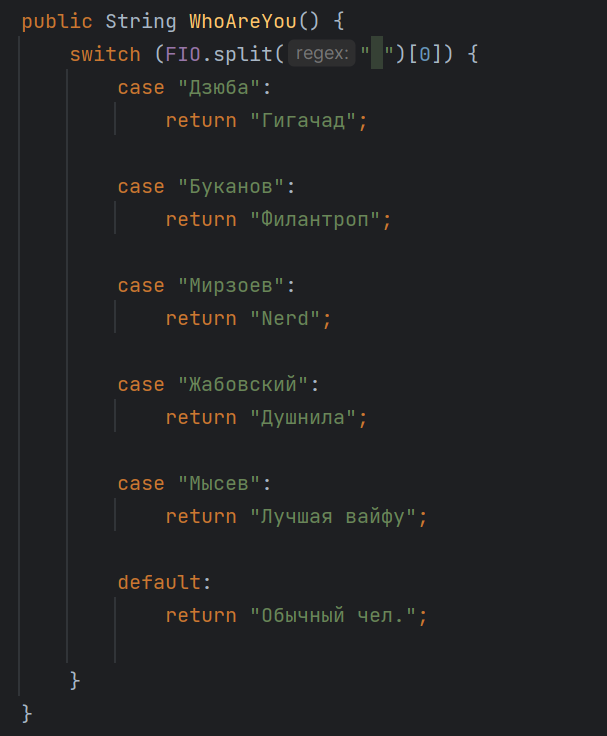


Рисунок 33 – Метод WhoAreYou класса Worker

Следующим методом будет метод, описывающий микропроцессоры, производимые компанией. У них есть несколько полей, отвечающих за х-ки микропроцессора и прочие вещи.

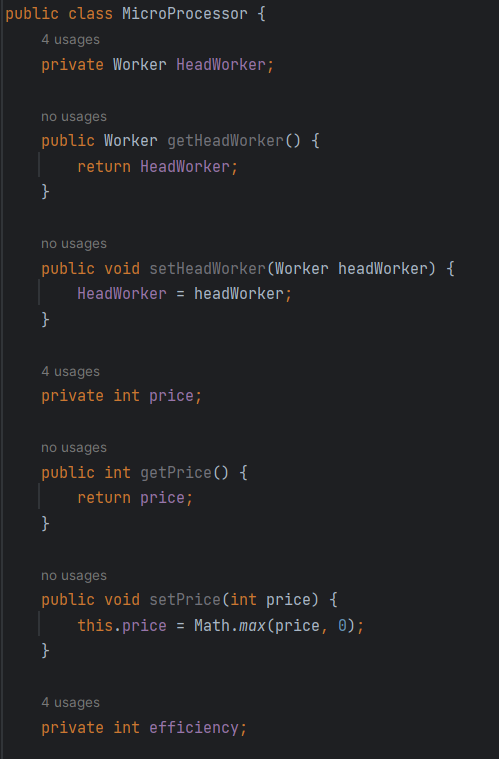


Рисунок 34 – Поля класса MicroProcessor

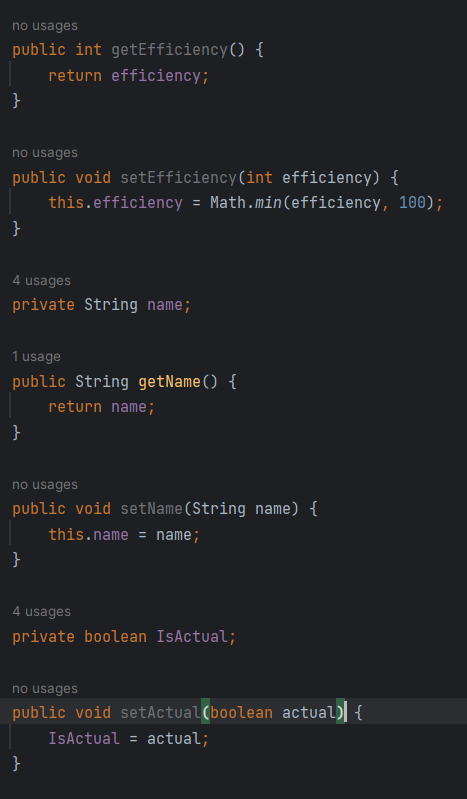


Рисунок 35 – Поля и get-set конструкции класса MicroProcessor



Рисунок 36 – Методы и конструктор класса MicroProcessor

Третий класс – DaughterCompanies. В нем описаны дочерние компании, часть которых принадлежит нашей компании. У нее есть поля своей ценности, нашу долю в ней, список изготовляемых микропроцессоров и имя компании. Кроме того, реализованы два метода с бизнес-логикой.

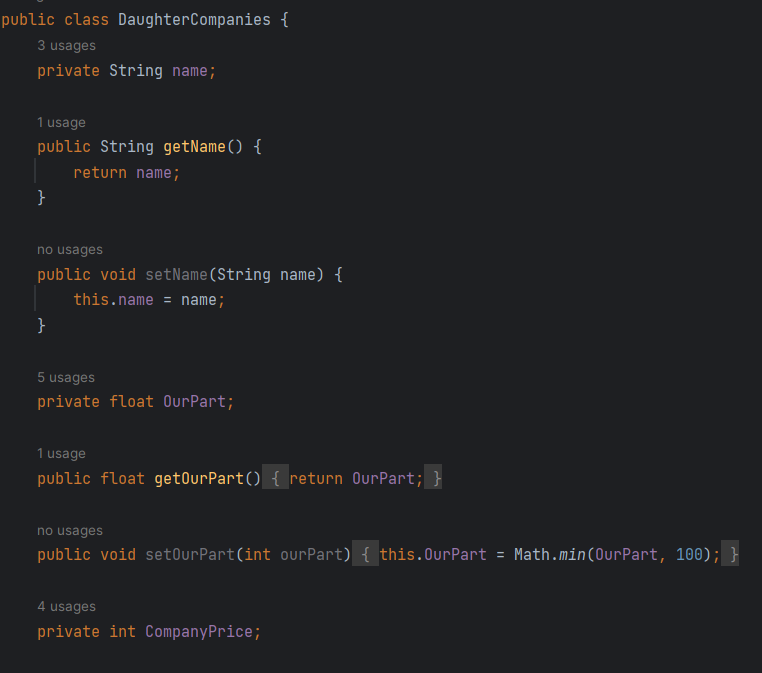


Рисунок 37 – Поля класса DaughterCompanies

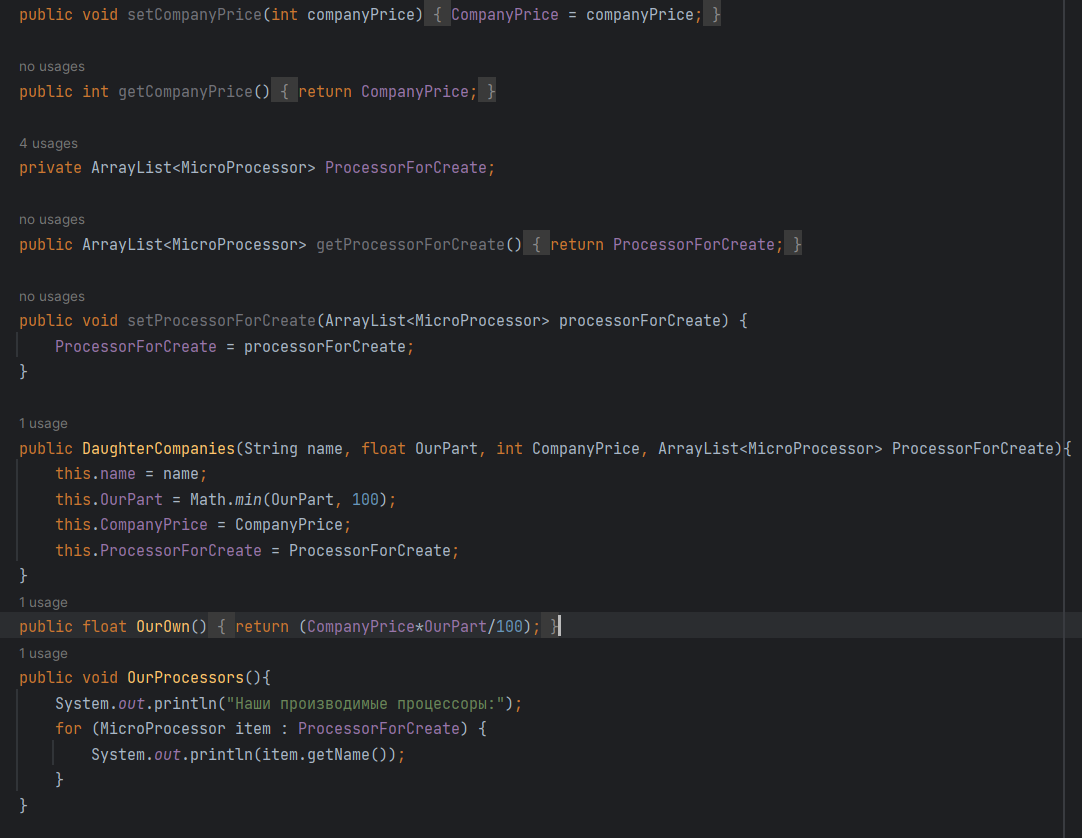


Рисунок 38 – Методы и конструктор класса DaughterCompanies

Теперь, посмотрим на результат работы. Для начала, протестируем класс Worker:



Рисунок 39 – Результат работы класса Worker

Как видим, методы и установки работают верно. Теперь, рассмотрим микропроцессоры.

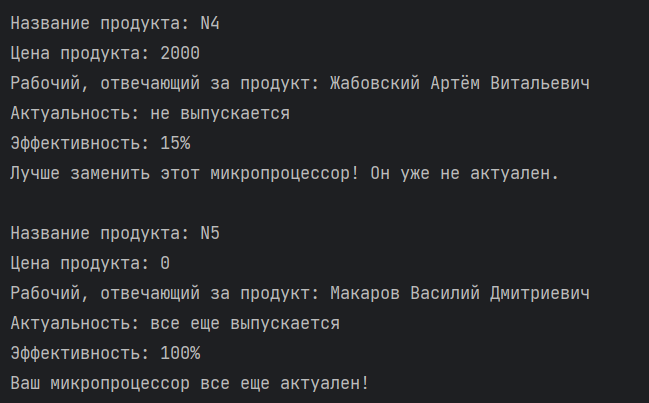


Рисунок 40 – Результат работы класса MicroProcessor

Здесь тоже проблем нет. Ну, и последний класс:

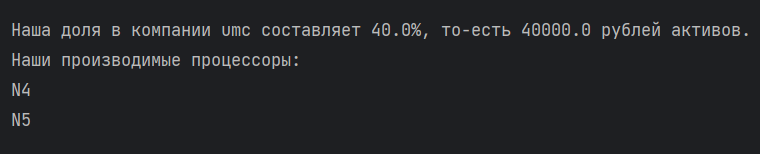


Рисунок 41 – Результат работы класса DaughterCompanies

Здесь тоже все работает верно, в том числе и список процессоров.

Вывод: научились работать с методами и классами в Java, описали не менее трех классов в соответствии с сущностями вашей предметной области.

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: «Создание приложения»

Цель работы: Создать приложение, состоящее из трех экранов: экран авторизации, профиля и настроек, научившись работать в Android Studio, разобравшись в его структуре.

Для начала, следует создать поля авторизации, в которых пользователь будет вводить логин и пароль, заранее прописанные в коде.



Рисунок 42 – Поля для ввода логина и пароля

Теперь, нужно создать кнопку, при нажатии которой наш логин и пароль будут считываться и проводиться авторизация в случае успешного совпадения данных.

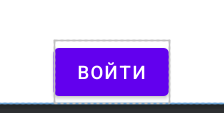


Рисунок 43 – Кнопка «Войти»

Теперь же – создадим новое окно. Это будет окно личного кабинета. В нем будет находится надпись «Личный кабинет» и кнопка, по нажатию которой будет открыто окно настроек.

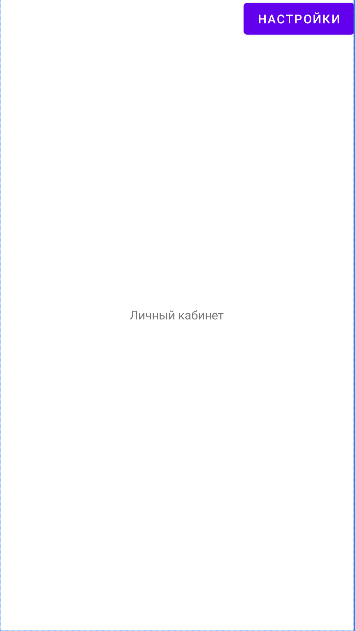


Рисунок 44 – Окно Личного кабинета

Ну, и под конец, создадим это самое окно настроек. Оно мало чем будет отличаться от личного кабинета, за исключением того, что расположена кнопка будет в другом месте и пересылать она будет в поле авторизации.

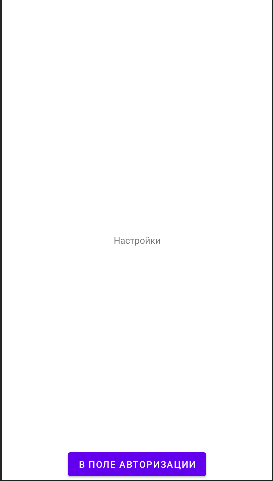


Рисунок 45 – Окно настроек

С дизайном покончено, теперь создадим логику для наших компонентов.

Код работы:

Здесь было принято решения создать несколько переменных, отвечающих за элементы на экране. Кроме того, здесь присутствует логическая проверка на данные. Если данные не совпадают – пользователю выводится сообщение о том, что логин или пароль были введены неверно.



Рисунок 46 – Код основного окна

Здесь и в следующем окне описан лишь переход из одного окна в другой при нажатии кнопки.

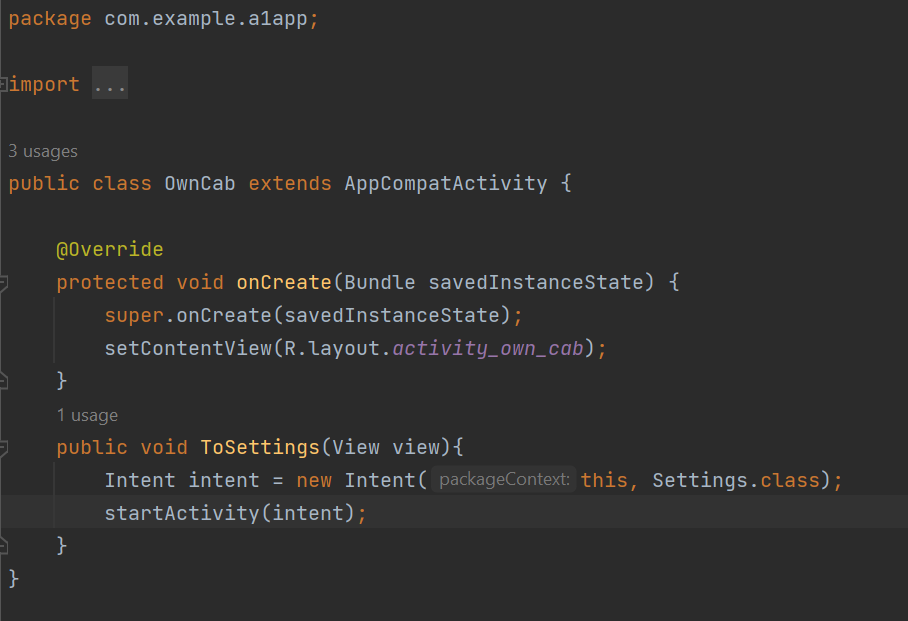


Рисунок 47 – Код окна Личного кабинета

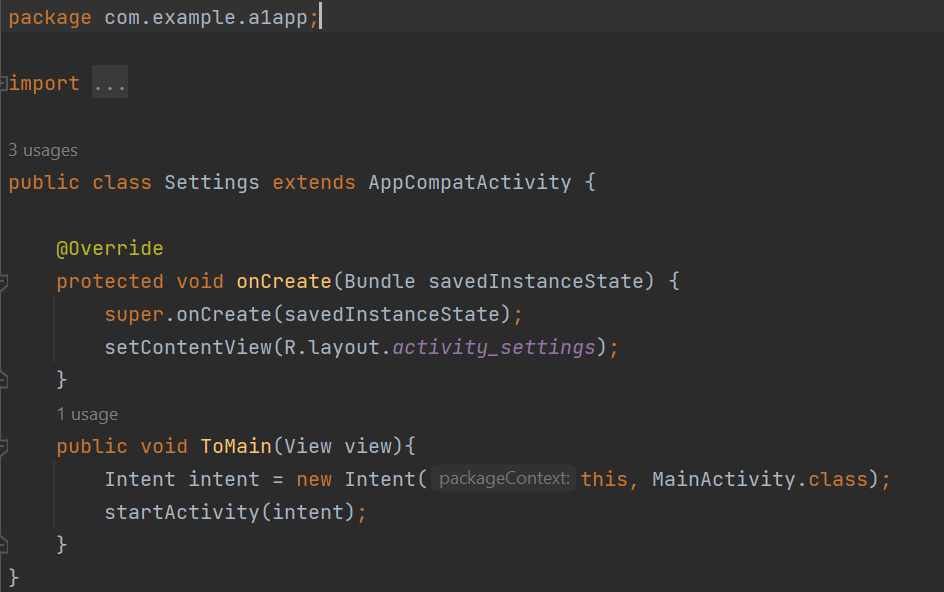


Рисунок 48 – Код окна настроек

Вывод: Создали приложение, состоящее из трех экранов: экран авторизации, профиля и настроек, научившись работать в Android Studio, разобравшись в его структуре.

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема «Проверка на ввод»

Цель работы: научиться работать с введенными данными пользователя в языке Java используя Android Studio.

В пределах этой практической работы необходимо дополнить практическую работу под номером 5 с помощью проверок логина и пароля пользователя, а также передачей данных между окнами программы. Для начала, создадим проверки на логин и пароль. Условия: логин должен быть в формате электронной почты, электронная почта не должна быть пустой, пароль должен содержать спецсимволы, пароль должен содержать большие буквы.

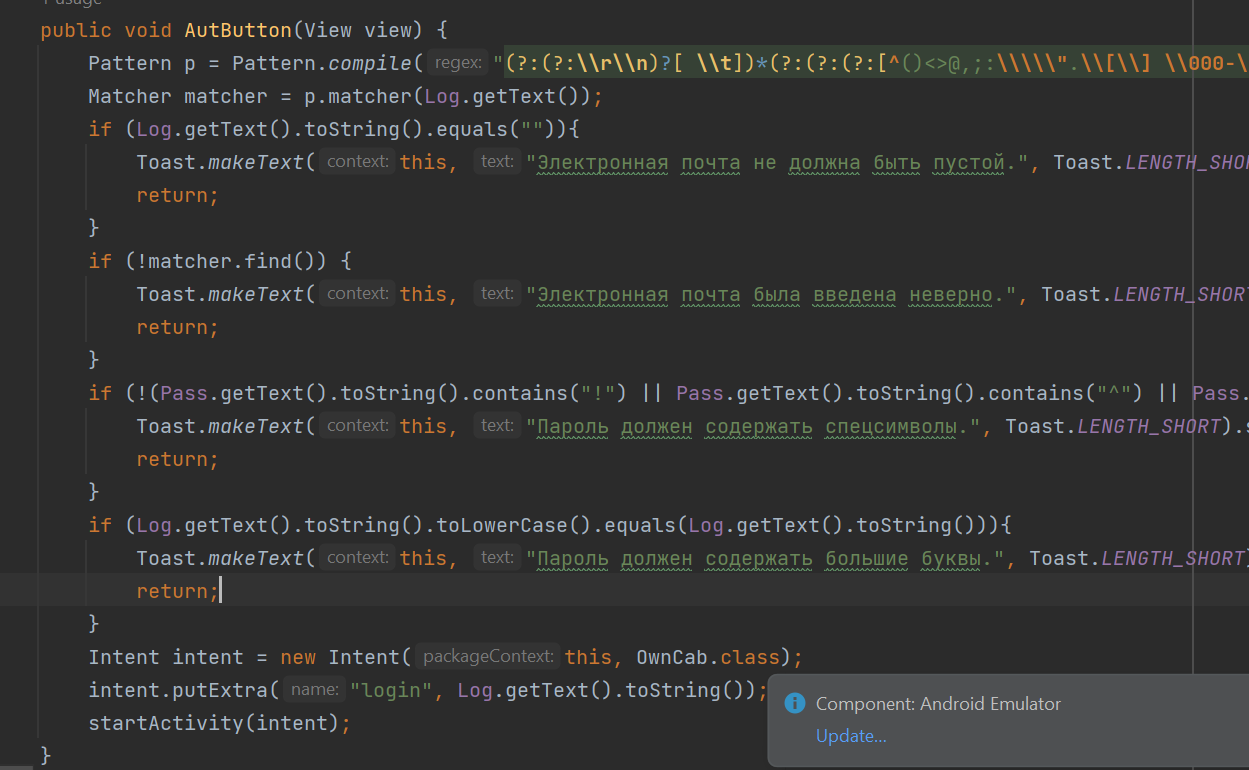


Рисунок 49 – Проверка на введенный пароль

Переданные данные из окна ввода пароля нужно получить из другого окна. Вот как это сделано:

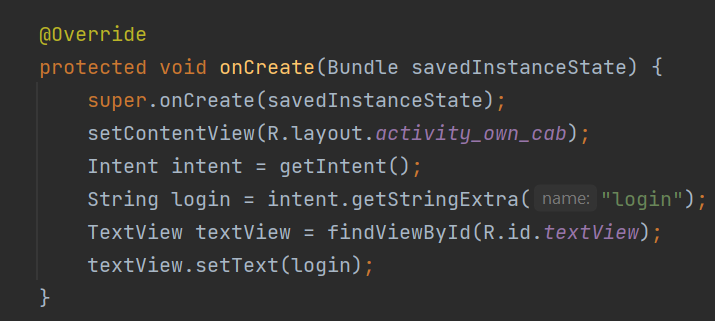


Рисунок 50 – Получение данных из прошлого окна

Вывод: научились работать с введенными данными пользователя в языке Java используя Android Studio.

ПРАКТИЧЕСКАЯ РАБОТА №7

Тема «Калькулятор»

Цель: разработать калькулятор, выполняющий расчеты и изменяющийся в зависимости от ориентации экрана пользователя, сохраняя введенные ранее данные.

Для начала, рассмотрим созданные интерфейсы. Их два- книжный, и альбомный. Различаются они наличием расширенных функций.

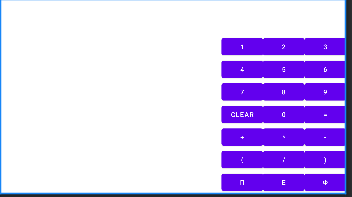


Рисунок 51 – Интерфейс при альбомной ориентации

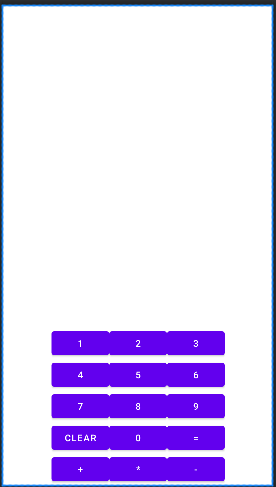


Рисунок 52 – Интерфейс при книжной ориентации

Теперь, рассмотрим код. В данной части кода находятся различные переменные, необходимые в коде работы. Кроме того, здесь есть метод onSaveInstanceState, сохраняющий данные при изменении ориентации.

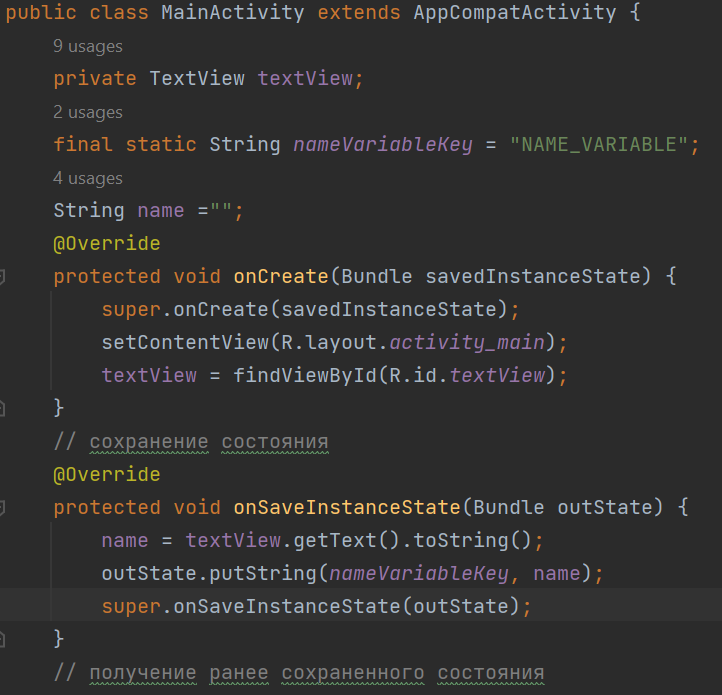


Рисунок 53 – Начало функции main

Здесь – выполняется присвоение полю сохраненного ранее текста при смене ориентации.

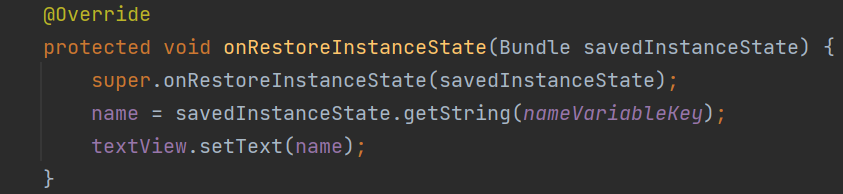


Рисунок 54 – Установка сохраненного ранее текста из поля

Здесь выполняются расчеты при помощи библиотеки.



Рисунок 55 – Функция расчета выражения

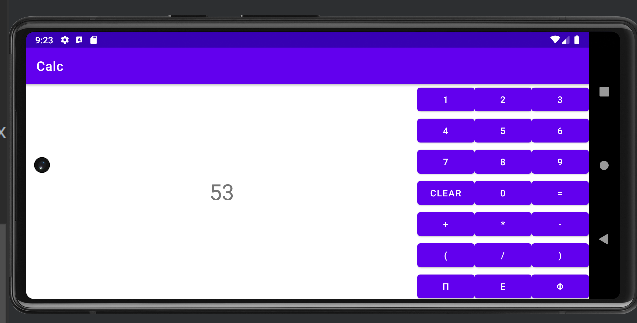


Рисунок 56 – Результат работы

Вывод: разработали калькулятор, выполняющий расчеты и изменяющийся в зависимости от ориентации экрана пользователя, сохраняя введенные ранее данные.