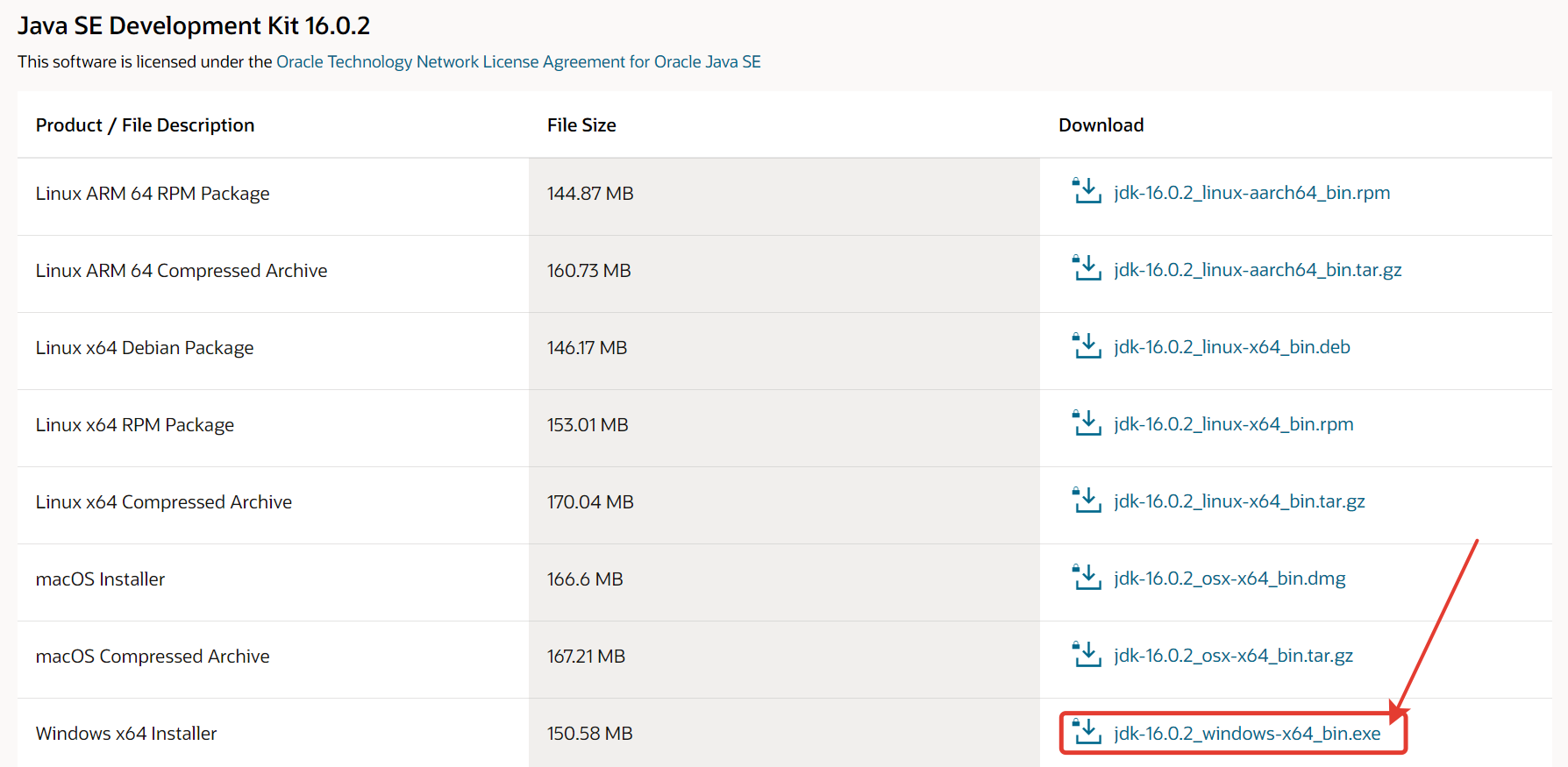
Занятие 1. Настройка среды программирования

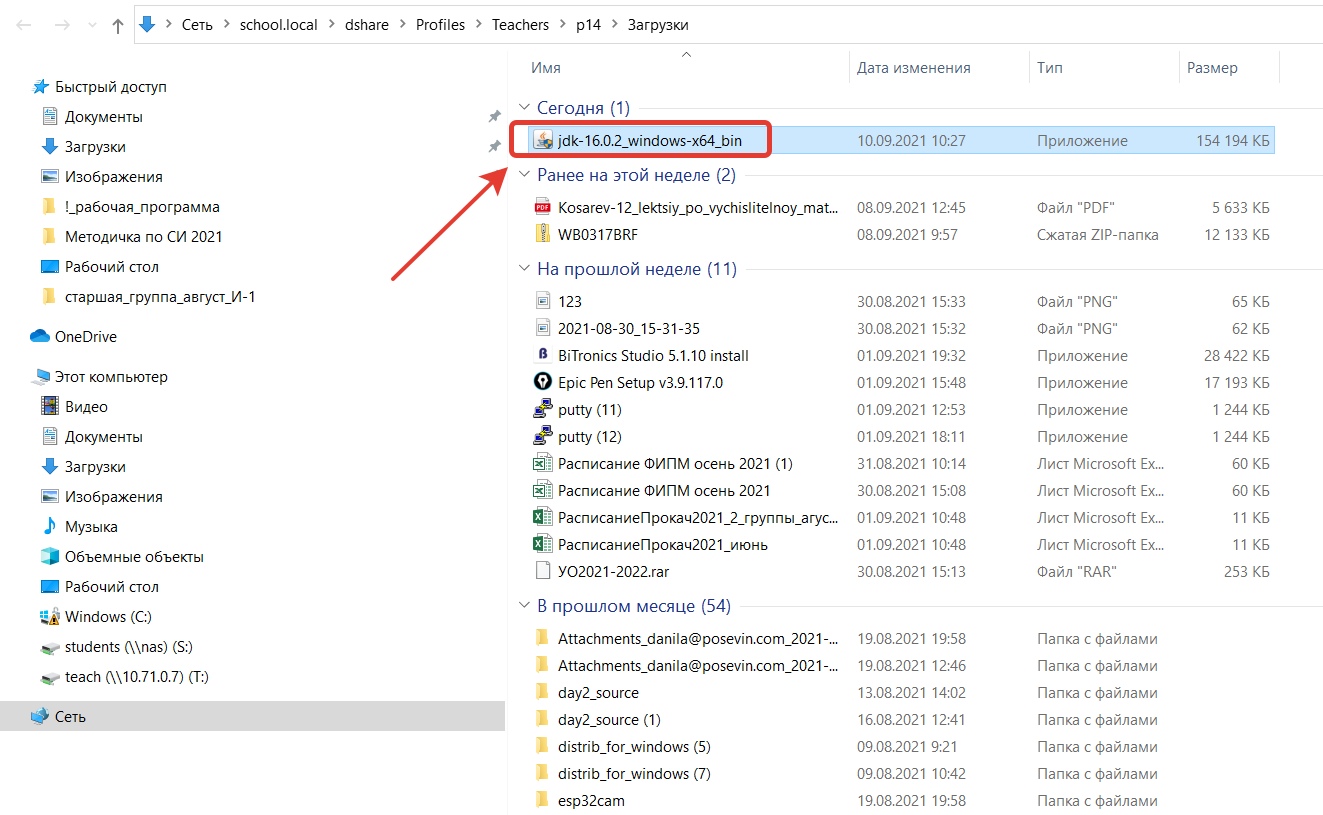
Установка Java Development Kit (JDK)

Идем по ссылке <https://www.oracle.com/java/technologies/javase-downloads.html>

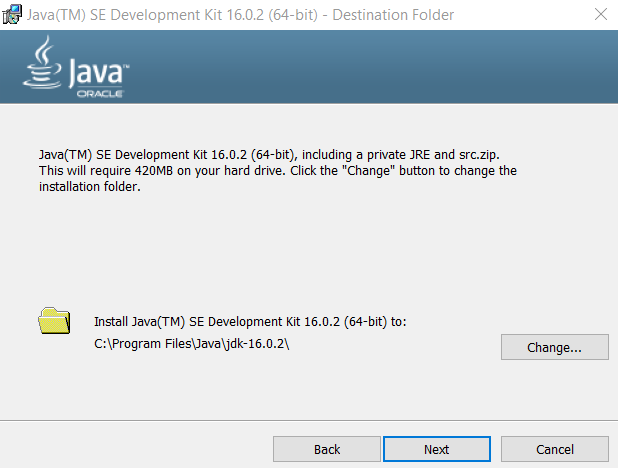
Далее идем <https://www.oracle.com/java/technologies/javase-jdk16-downloads.html>



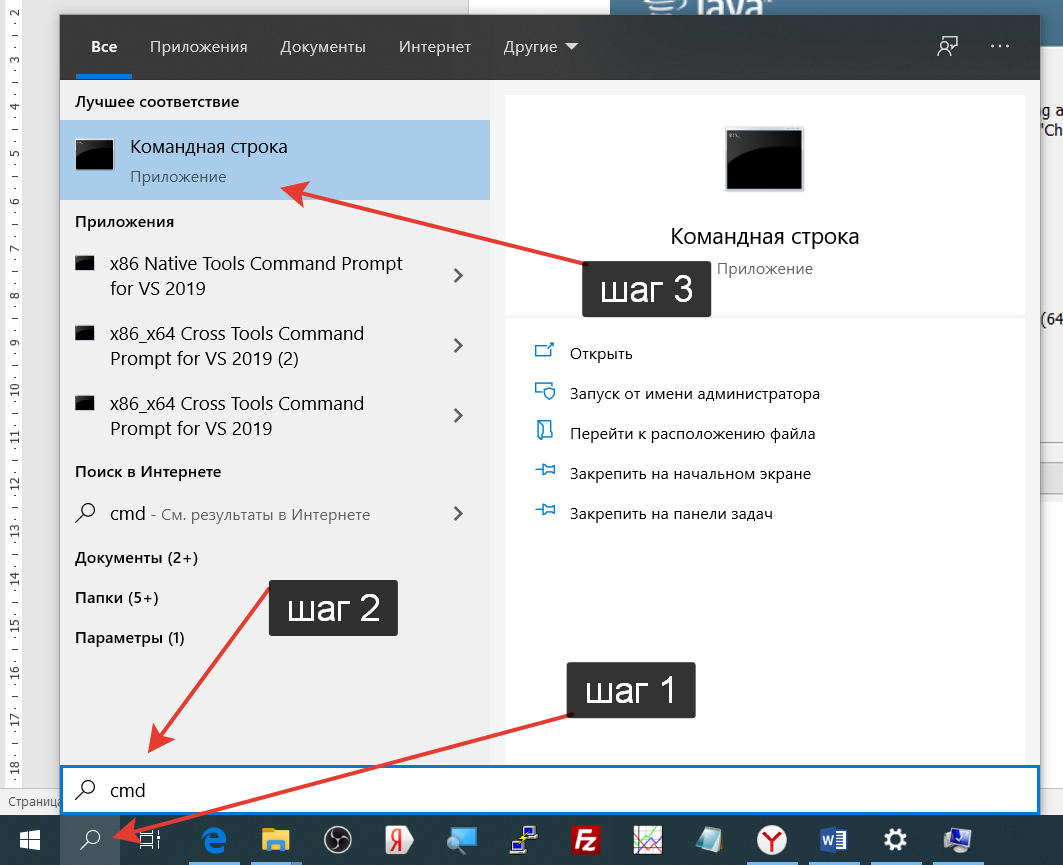
Устанавливаем



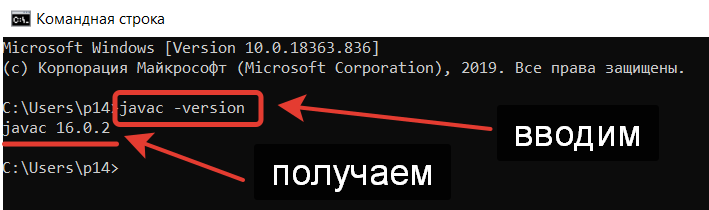
Нажимаем далее



Проверяем установился ли компилятор:



Проверяем установку

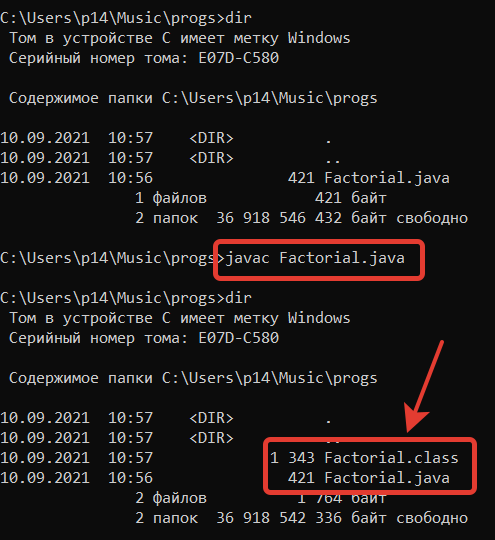


Откомпилируйте исходный код (см ниже пример кода) с помощью команды   
*javac Factorial.java*

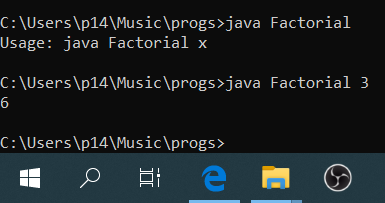
Убедитесь, что в текущем каталоге появился файл   
*Factorial.class*

и запустите его командой   
*java Factorial 3*

Программа должна вывести число 6.



Запускаем приложение:



Примеры программ

Пример 1. HelloWorld

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

}

Пример 2. Factorial

import java.util.stream.IntStream;

public class Factorial

{   
 public static void main(String [] args)

{   
 if (args.length == 0)   
 {   
 System.out.println("Usage: java Factorial x");   
 }   
 else   
 {   
 int n = Integer.parseInt(args [0]);  
 var numbers = IntStream.range(1, n+1);   
 int f = numbers.reduce(1, (r,x)-> r\*x);   
 System.out.println(f);   
 }   
 }   
}

Пример 3. Использование Stream API

Пример использования **static** IntStream range(**int** startInclusive, **int** endExclusive)

import java.util.\*;  
import java.util.stream.IntStream;

public class Demo {  
 public static void main(String[] args) {  
 IntStream intStream = IntStream.range(20, 30);  
 intStream.forEach(System.out::println);  
 }  
}

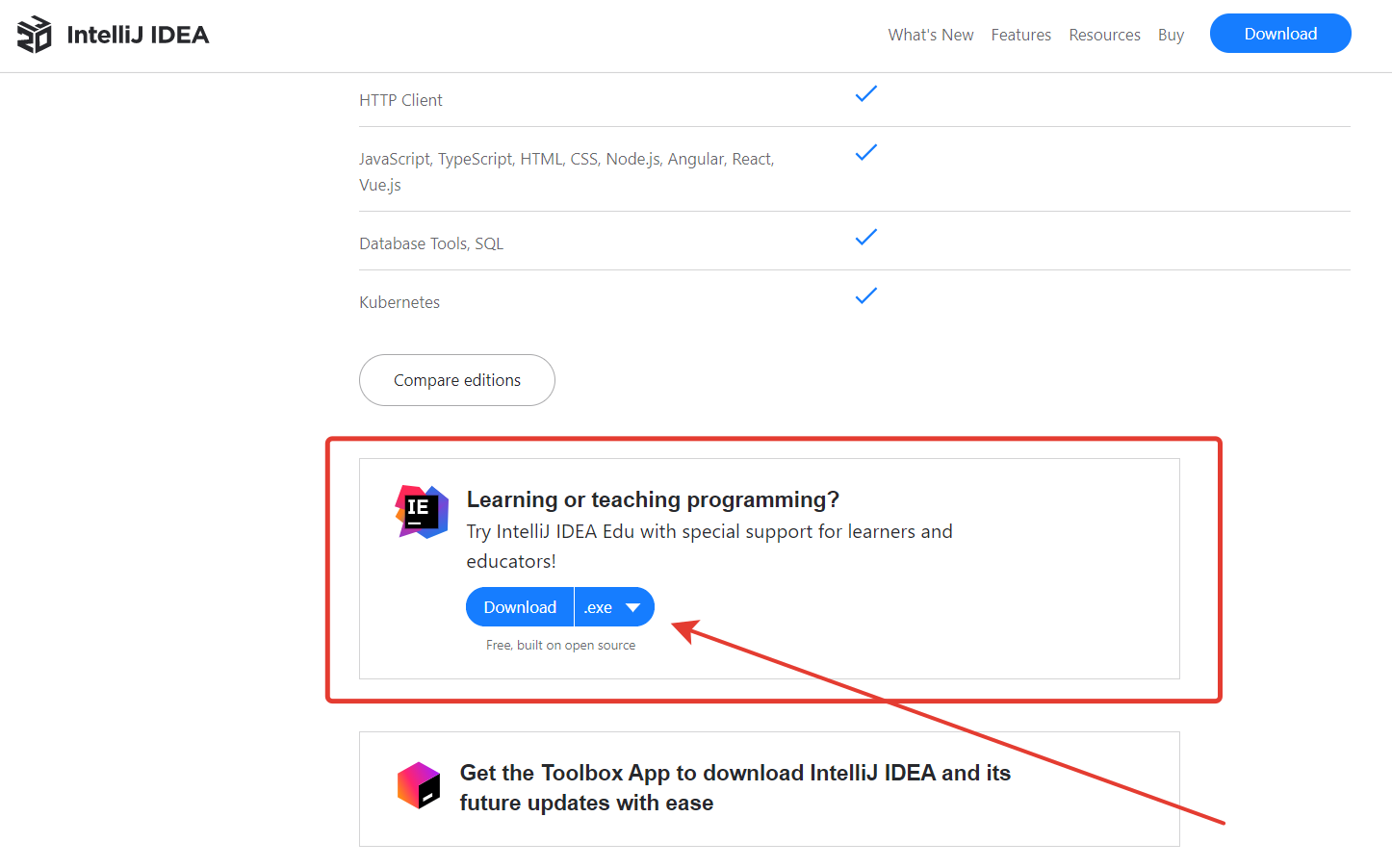
Пример4. Использование Reduce функции

Метод reduce позволяет выполнять агрегатные функции на всей коллекцией (такие как сумма, нахождение минимального или максимального значение и т.п.), он возвращает одно значение для стрима, функция получает два аргумента — значение полученное на прошлых шагах и текущее значение.  
  
**Условие**: Дана коллекция чисел Arrays.asList(1, 2, 3, 4, 2) выполним над ними несколько действий используя reduce.

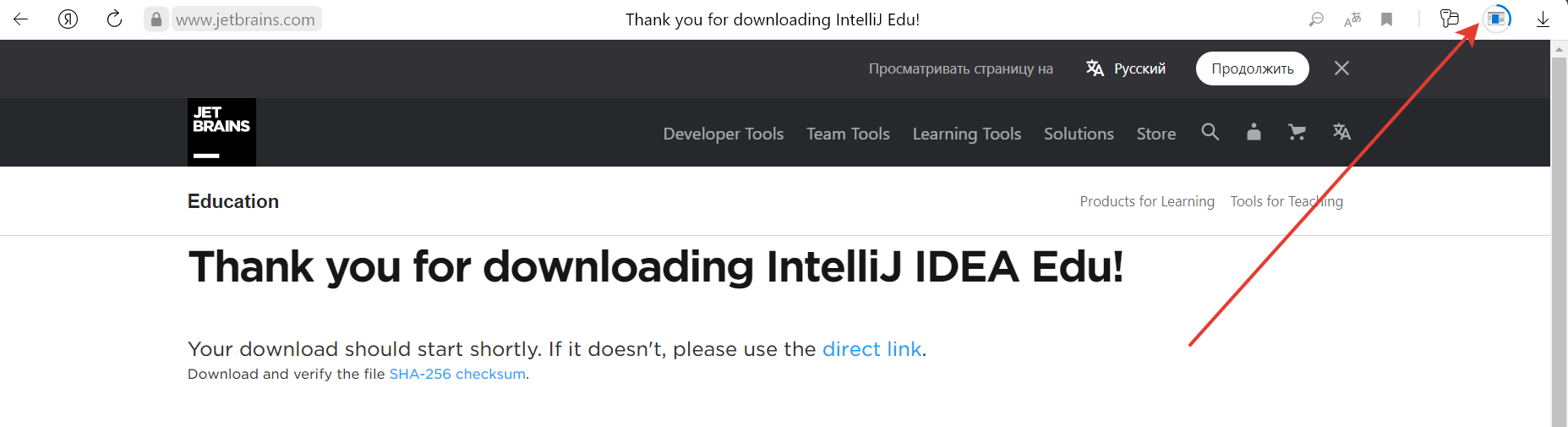
|  |  |  |
| --- | --- | --- |
| **Задача** | **Код примера** | **Результат** |
| Получить сумму чисел или вернуть 0 | collection.stream().reduce((s1, s2) -> s1 + s2).orElse(0) | 12 |
| Вернуть максимум или -1 | collection.stream().reduce(Integer::max).orElse(-1) | 4 |
| Вернуть сумму нечетных чисел или 0 | collection.stream().filter(o -> o % 2 != 0).reduce((s1, s2) -> s1 + s2).orElse(0) | 4 |

Установка интегрированной среды IntelliJ IDEA Edu

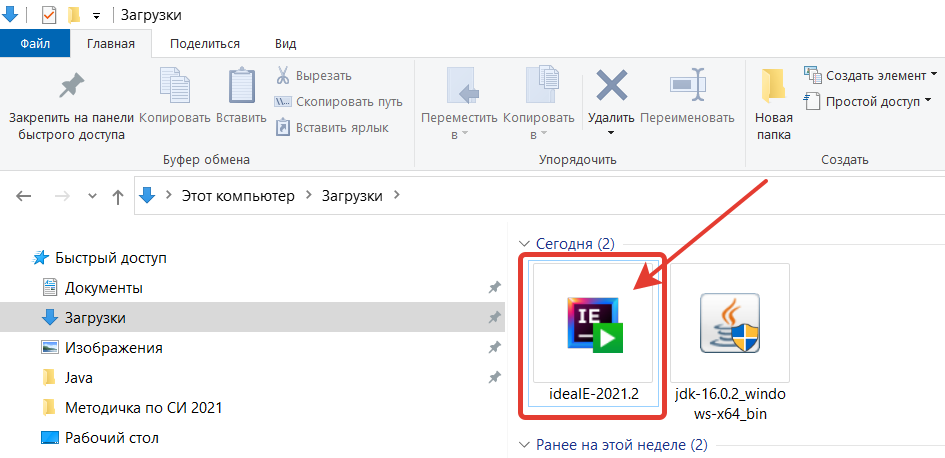
Дистрибутив интегрированной среды разработки IntelliJ IDEA Edu нужно взять со страницы [www.jetbrains.com/idea/download/](http://www.jetbrains.com/idea/download/)



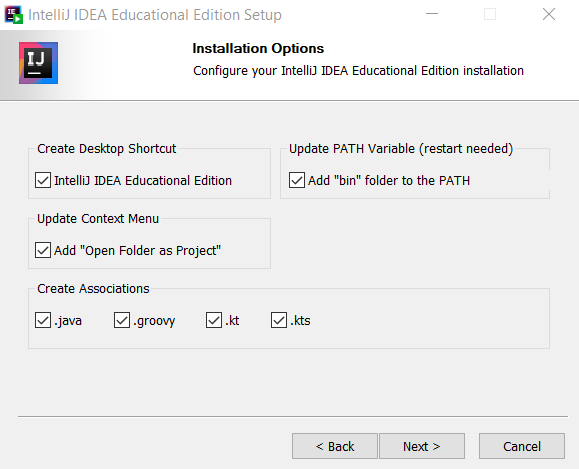
Скачиваем



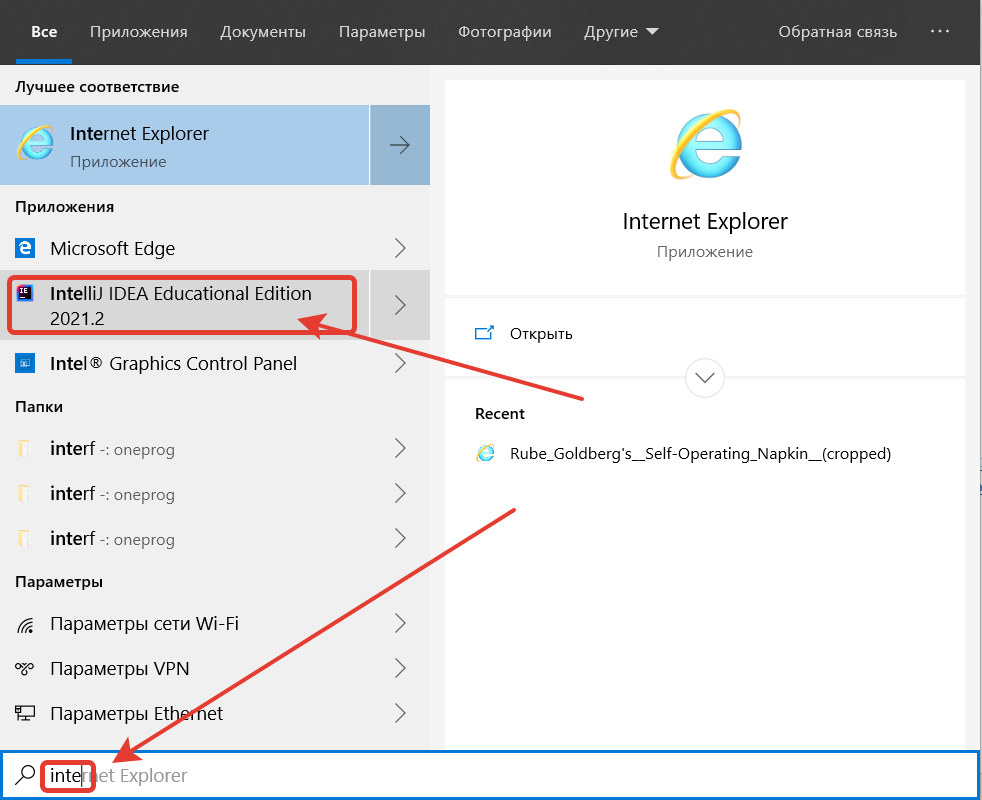
Устанавливаем



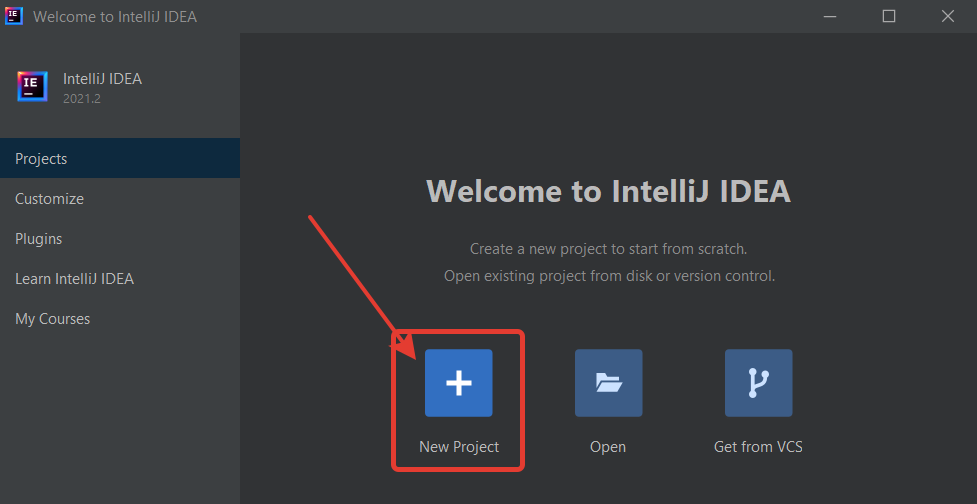
Ставим все галочки:

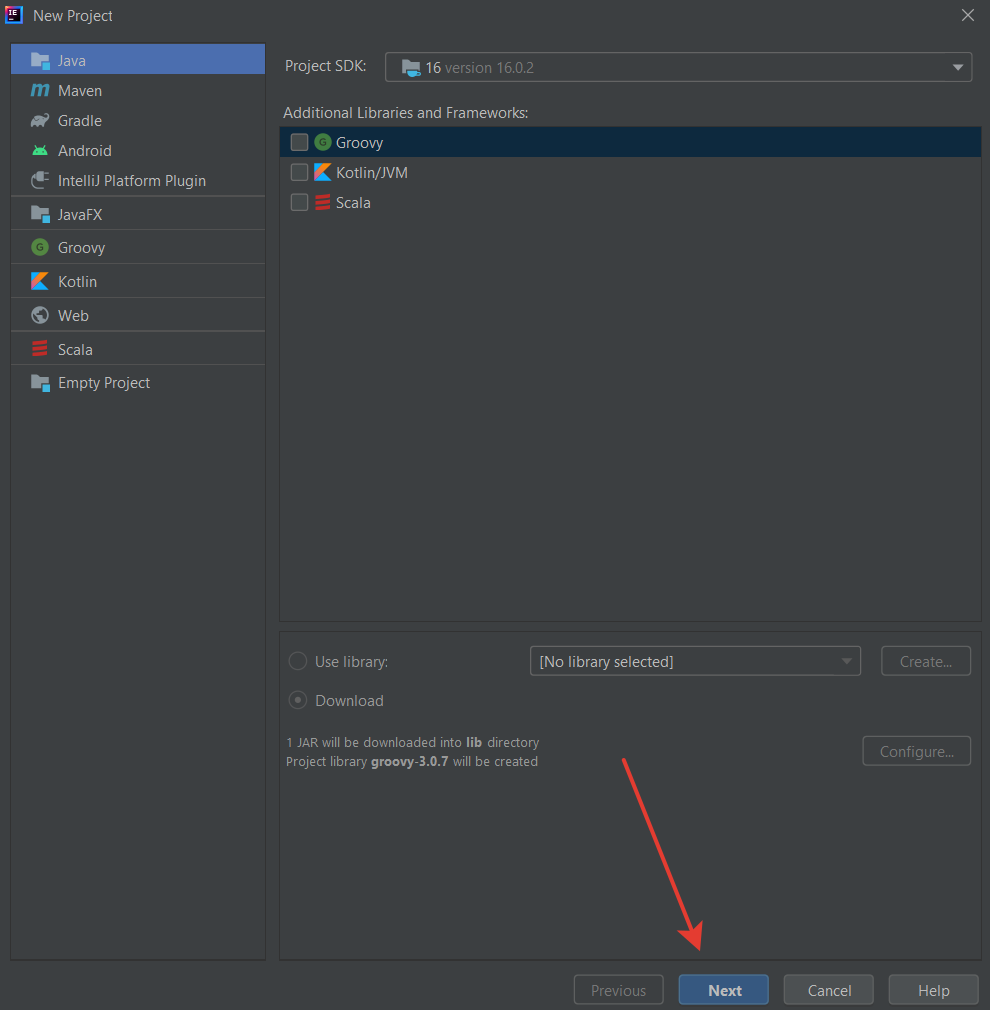


Запускаем IntelliJ IDEA

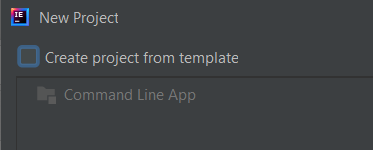


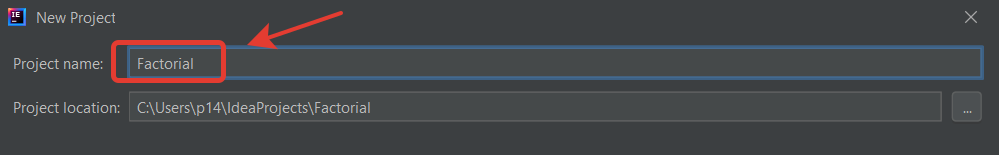
Создаем новый проект:



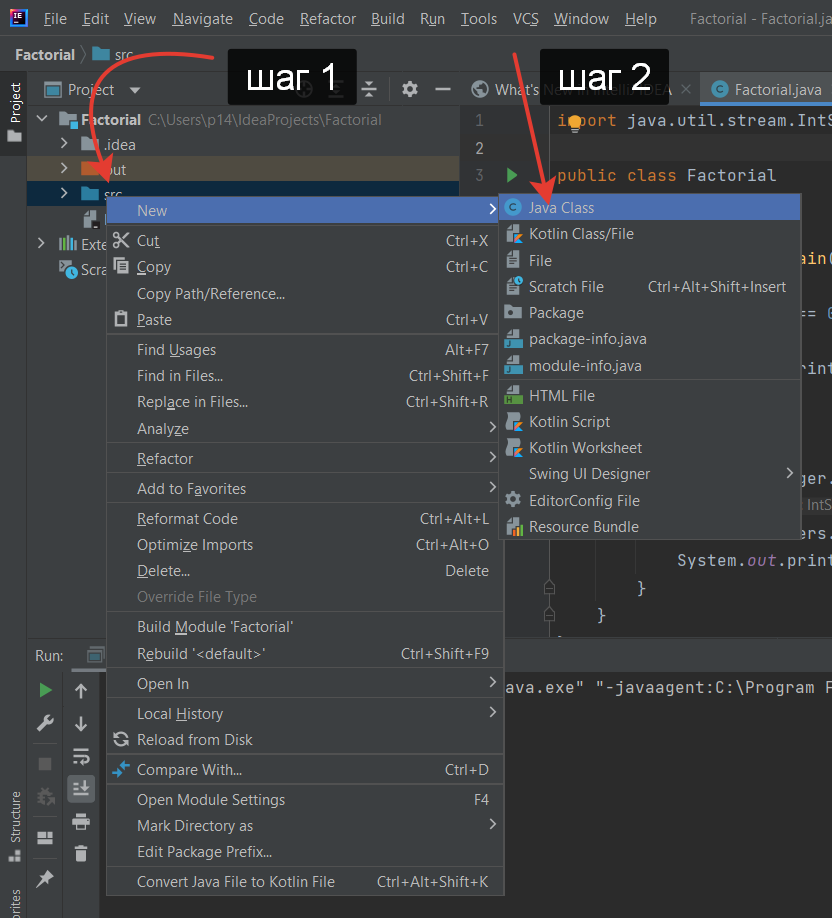


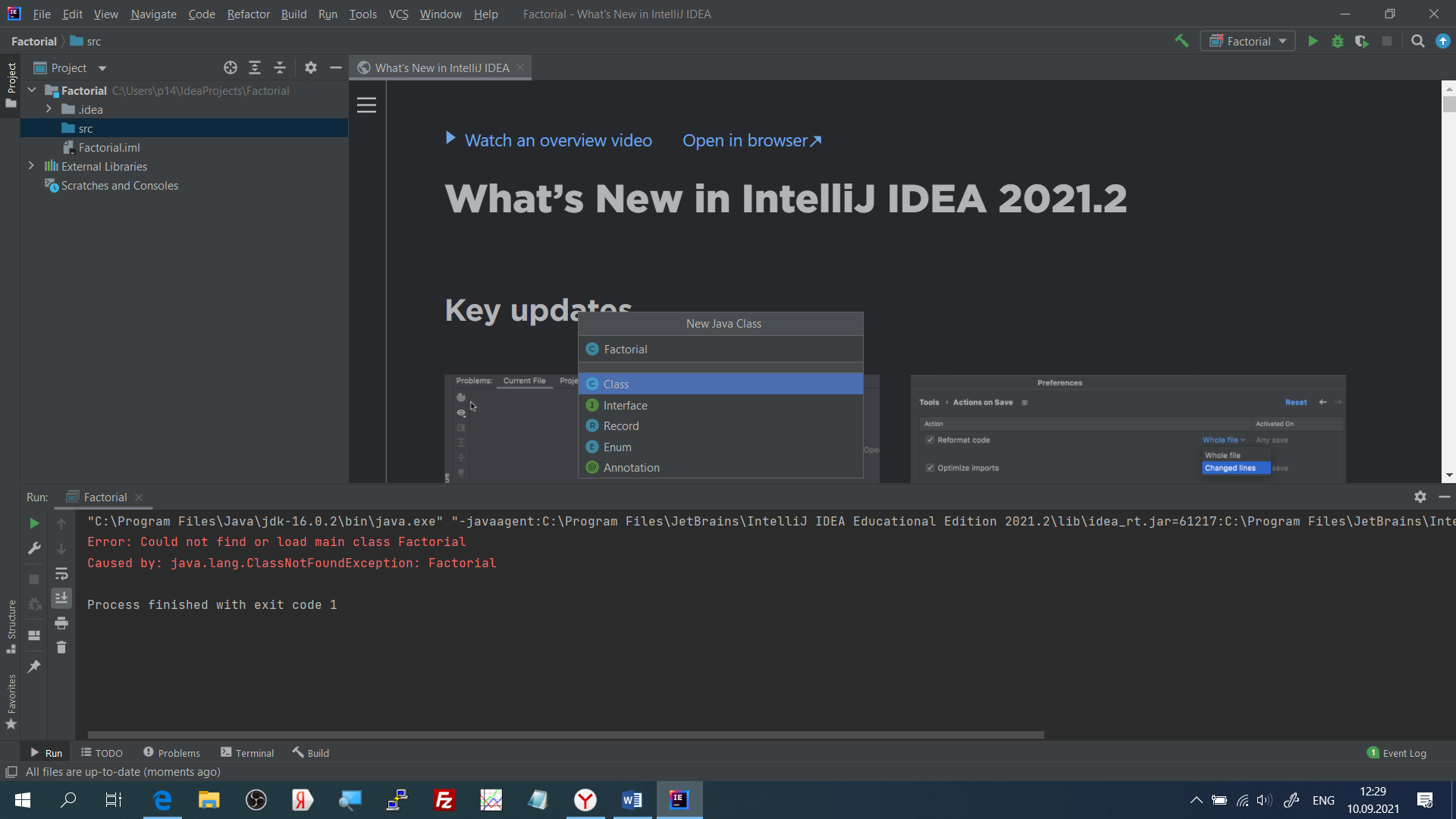
В checkbox-e (см ниже) ОБЯЗАТЕЛЬНО ставим галочку:





Создаем класс Factorial:



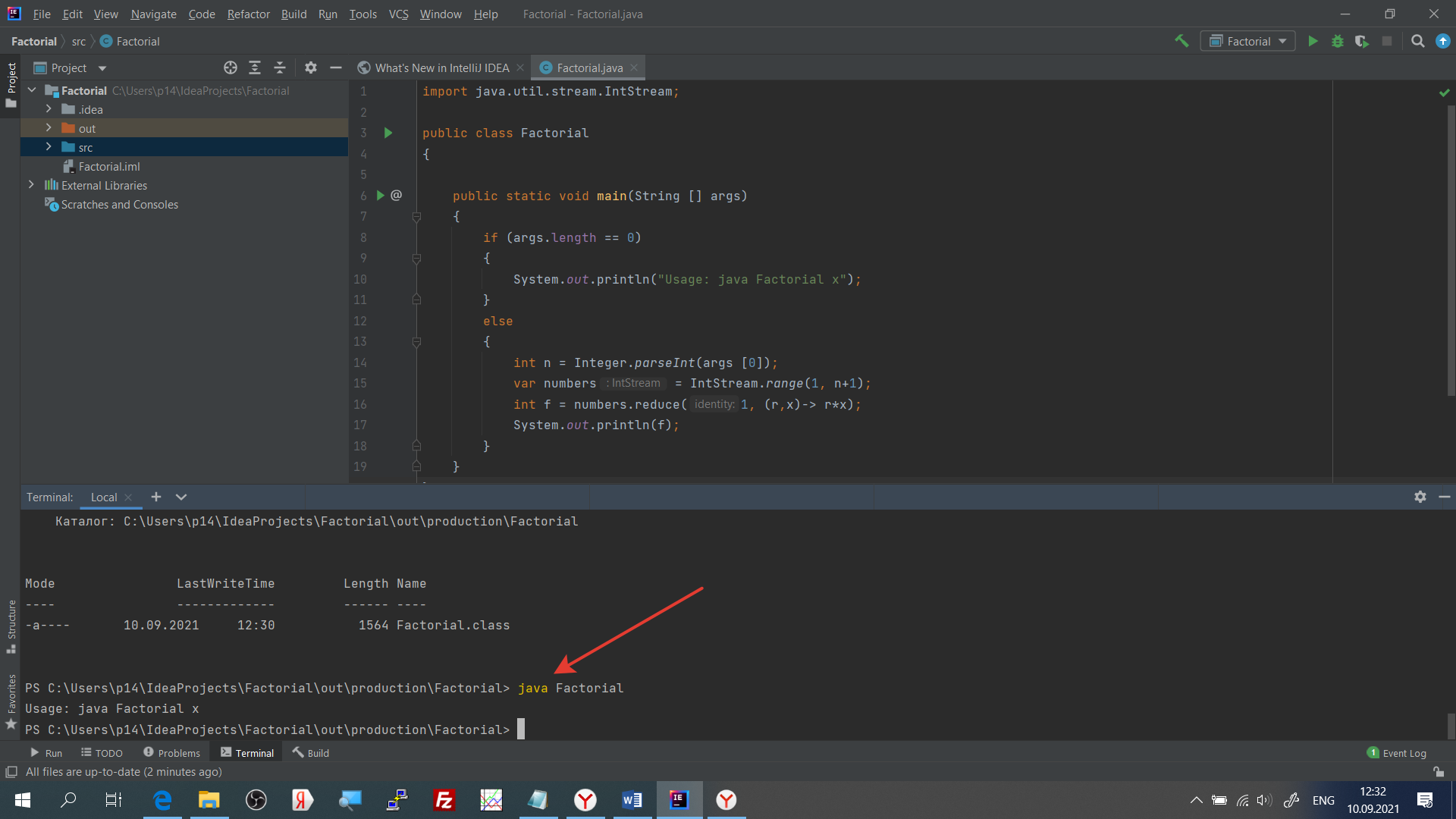


Вставляем исходный код:

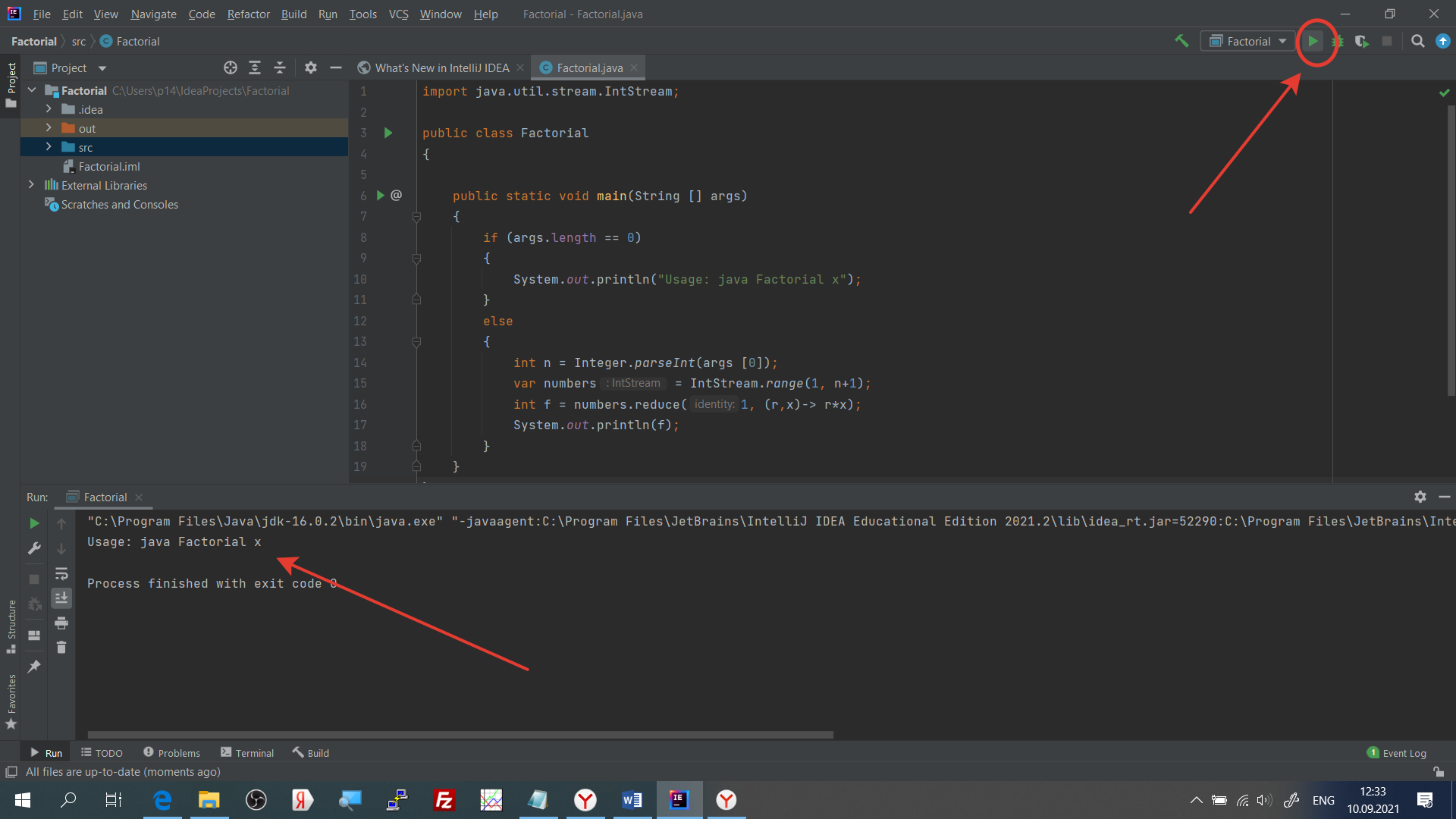
import java.util.stream.IntStream;

public class Factorial   
{   
 public static void main(String [] args)   
 {   
 if (args.length == 0)   
 {   
 System.out.println("Usage: java Factorial x");   
 }   
 else   
 {   
 int n = Integer.parseInt(args [0]);  
 var numbers = IntStream.range(1, n+1);   
 int f = numbers.reduce(1, (r,x)-> r\*x);   
 System.out.println(f);   
 }   
 }   
}

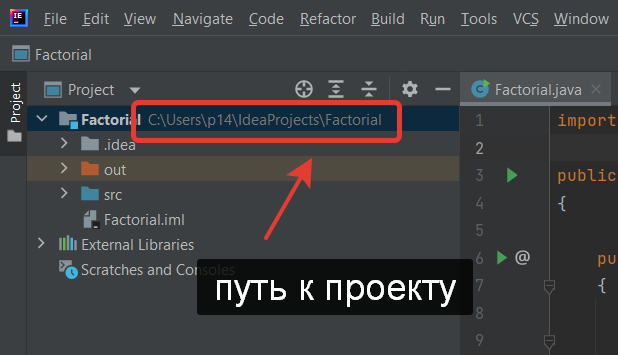
Запускаем из консоли



Запускаем из IntelliJ IDEA Edu



Проект будет создан в этой папке:



Литература

1. <https://habr.com/ru/company/luxoft/blog/270383/> - шпаргалка java программиста
2. <https://javarush.ru/groups/posts/2203-stream-api> - Stream API