|  |  |
| --- | --- |
|  | Министерство образования и науки Российской Федерации |
| Федеральное государственное автономное образовательное учреждение высшего образования |
| «Уральский федеральный университет имени первого Президента России Б. Н. Ельцина» |
| Институт радиоэлектроники и информационных технологий – РтФ |
|  |
|  | |

**ОТЧЕТ**

О ВЫПОЛНЕНИИ

ЛАБОРАТОНОЙ РАБОТЫ № 1

Руководитель ст. пр. Н. А. Архипов

Студент гр. РИМ-130971 Е.В. Треглазов

Екатеринбург 2024

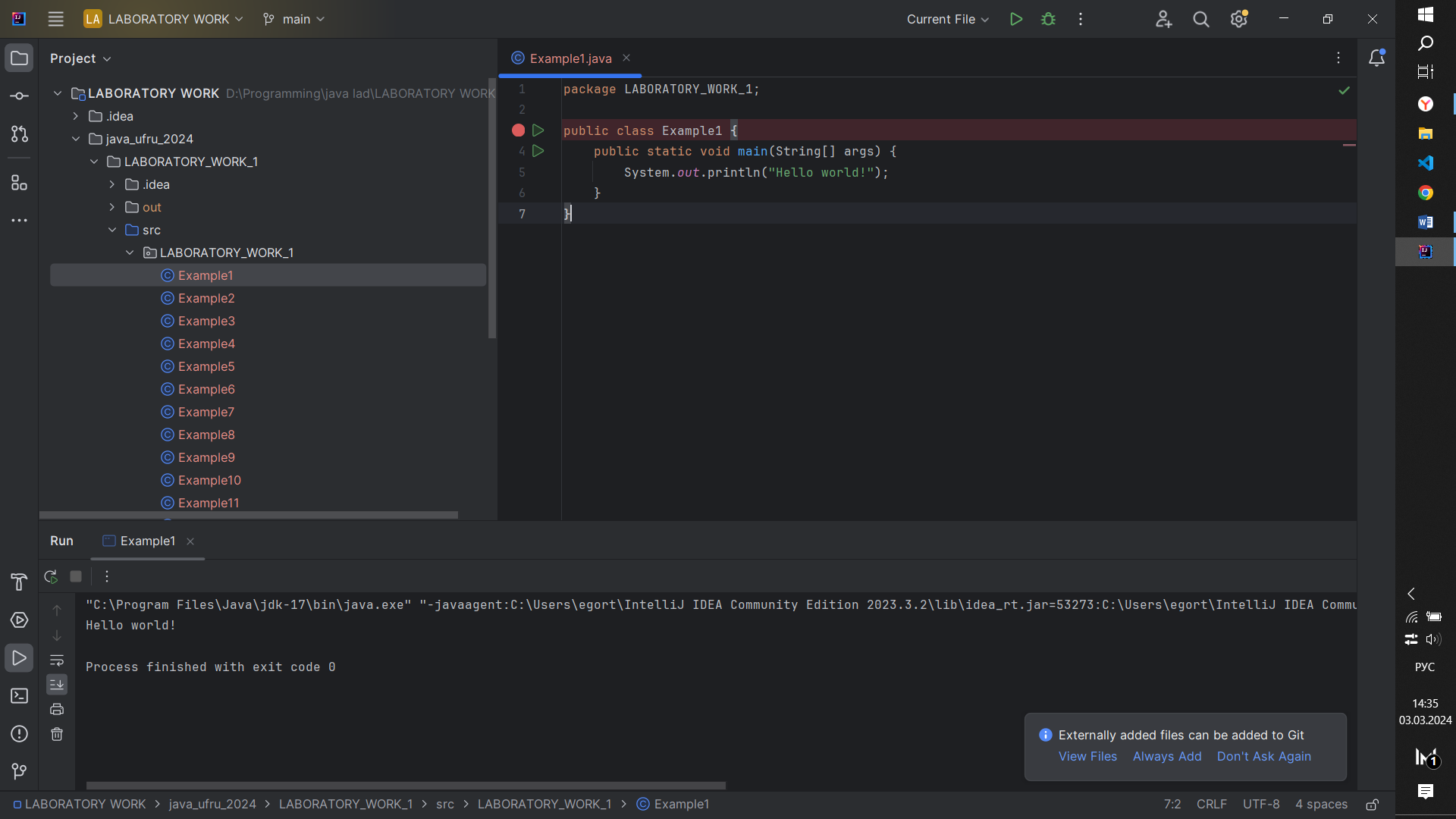
Цель: получить представление о написание программ на языке программирования Java с использованием командной строки, интегрированной среды разработки, а также решение задач на сайте timus.

Репозиторий: https://github.com/kuk86kuk/java\_ufru\_2024.

Example1

public class Example1 {  
 public static void main(String[] args) {  
 System.*out*.println("Hello world!");  
 }  
}

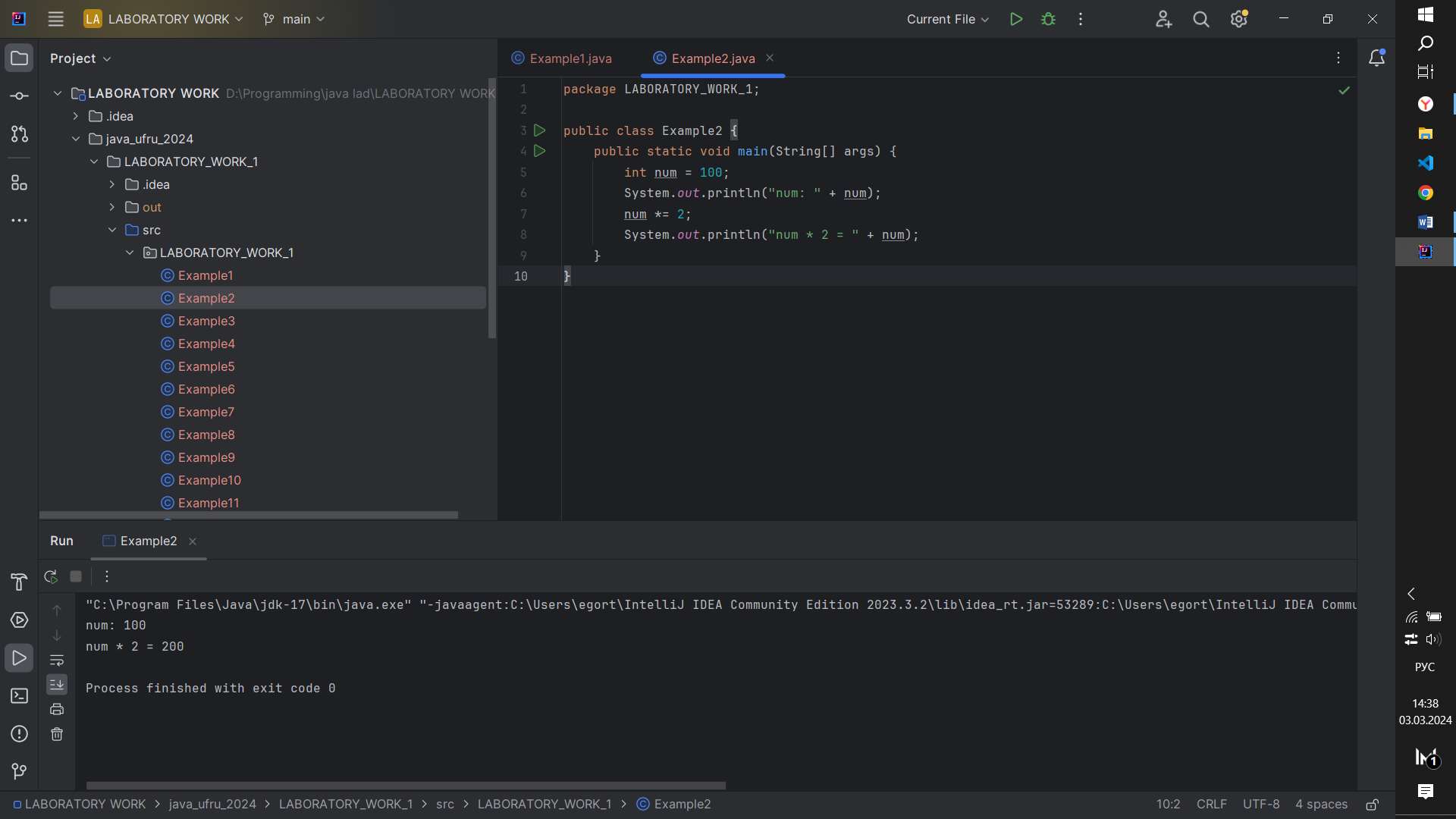
**Программа написана** на языке Java, которая выводит на экран сообщение "Hello world!". Программа состоит из одного класса Example1, который содержит метод main(). В теле метода используется команда System.out.println(), которая выводит строку "Hello world!" на консоль.



Example2

public class Example2 {  
 public static void main(String[] args) {  
 int num = 100;  
 System.*out*.println("num: " + num);  
 num \*= 2;  
 System.*out*.println("num \* 2 = " + num);  
 }  
}

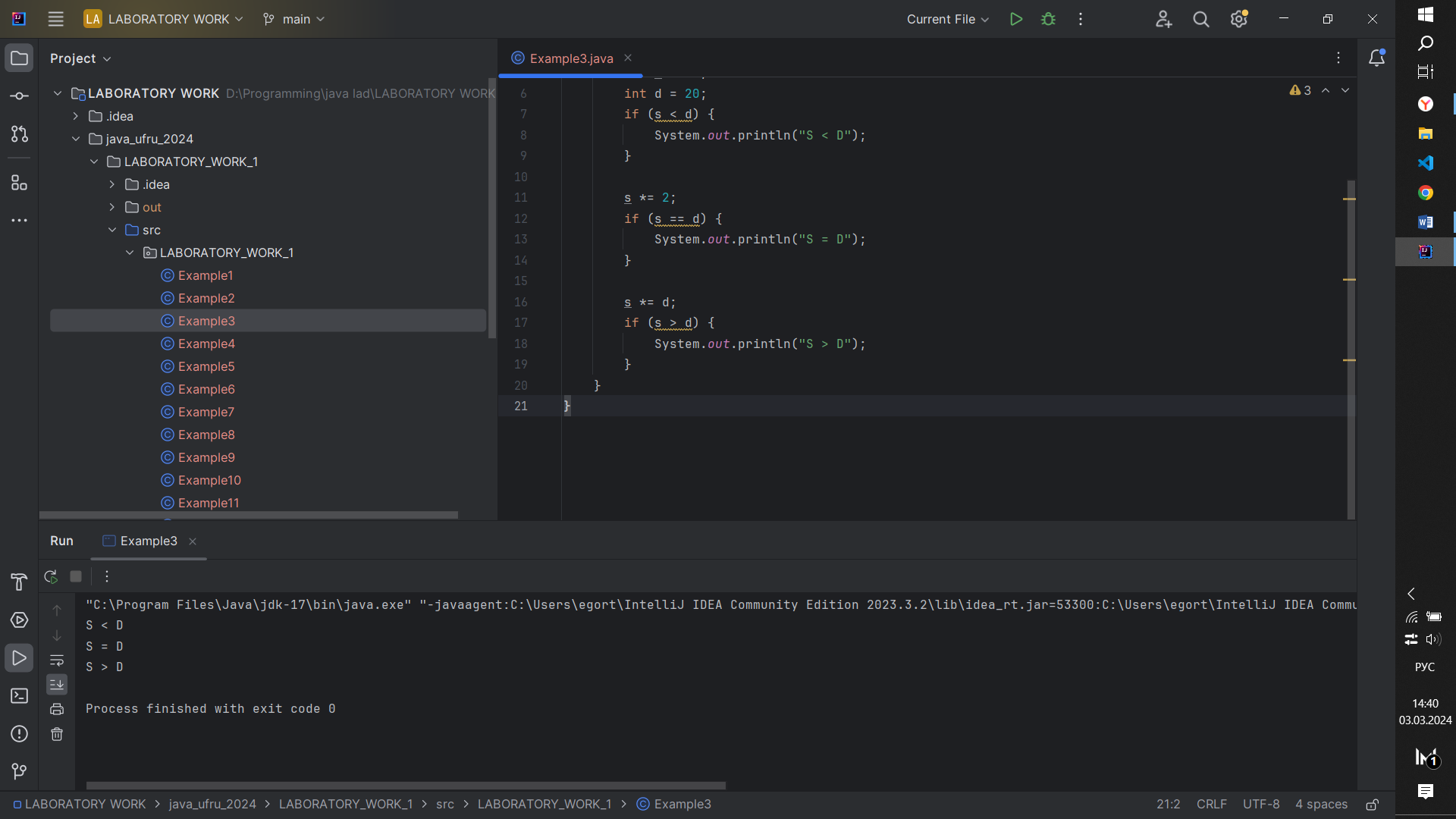
**Программа** выполняет простую математическую операцию умножения на два. Программа состоит из одного класса Example2, который содержит метод main(). В теле метода объявляется переменная num и присваивается ей значение 100. Затем выводится на экран значение num. Далее переменной num удваивается, и результат выводится на экран.



Example3

public class Example3 {  
 public static void main(String[] args) {  
 int s = 10;  
 int d = 20;  
 if (s < d) {  
 System.*out*.println("S < D");  
 }  
  
 s \*= 2;  
 if (s == d) {  
 System.*out*.println("S = D");  
 }  
  
 s \*= d;  
 if (s > d) {  
 System.*out*.println("S > D");  
 }  
 }  
}

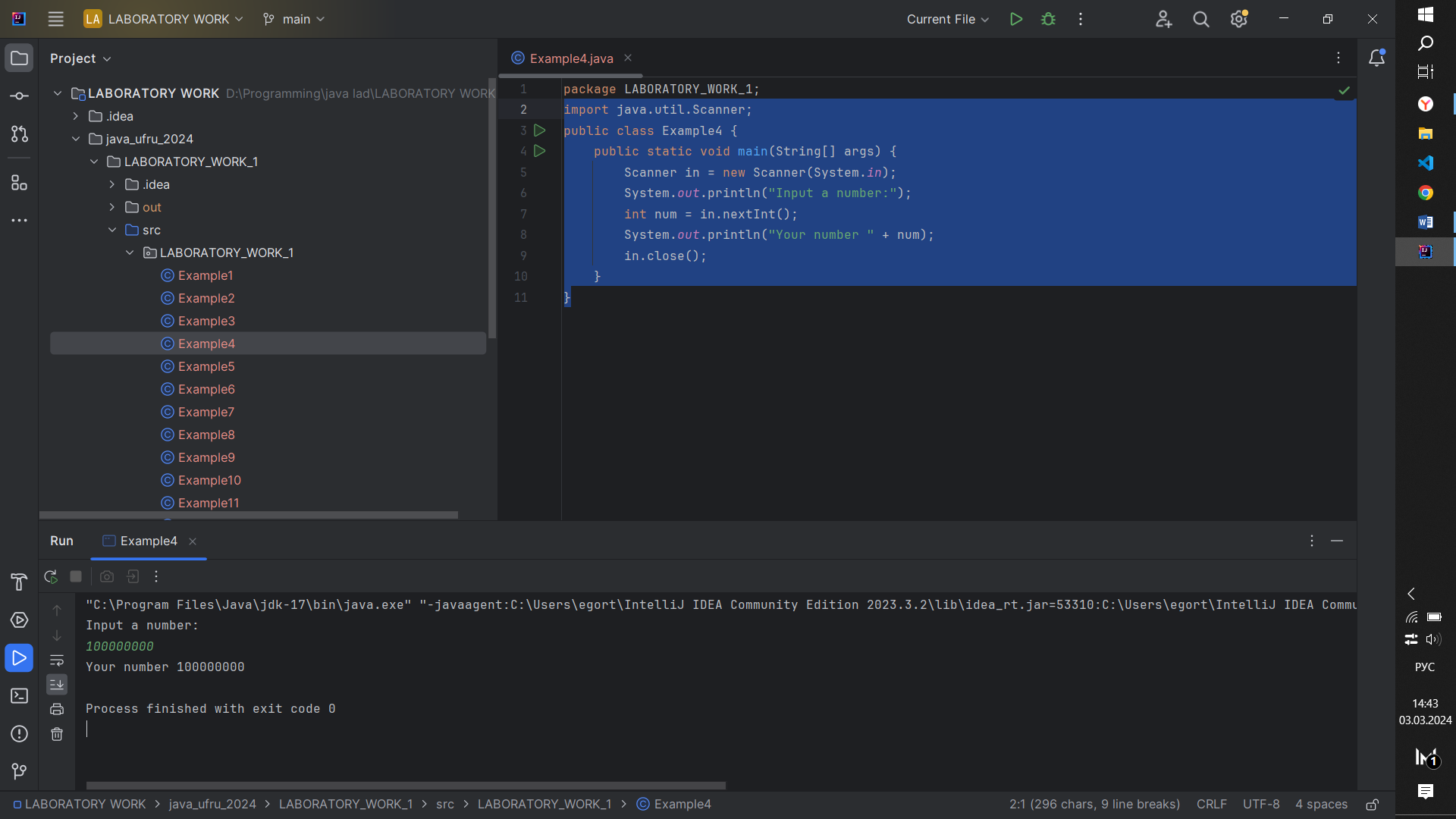
**Программа** выполняет простую логическую операцию сравнения двух чисел. Программа состоит из одного класса Example3, который содержит метод main(). В теле метода объявляются две переменные s и d и присваиваются им значения 10 и 20 соответственно. Затем выполняется условие, если s меньше d, то выводится на экран сообщение "S < D". Далее переменной s удваивается, и выполняется условие, если s равно d, то выводится на экран сообщение "S = D". Затем переменной s умножается на d, и выполняется условие, если s больше d, то выводится на экран сообщение "S > D".



Example4

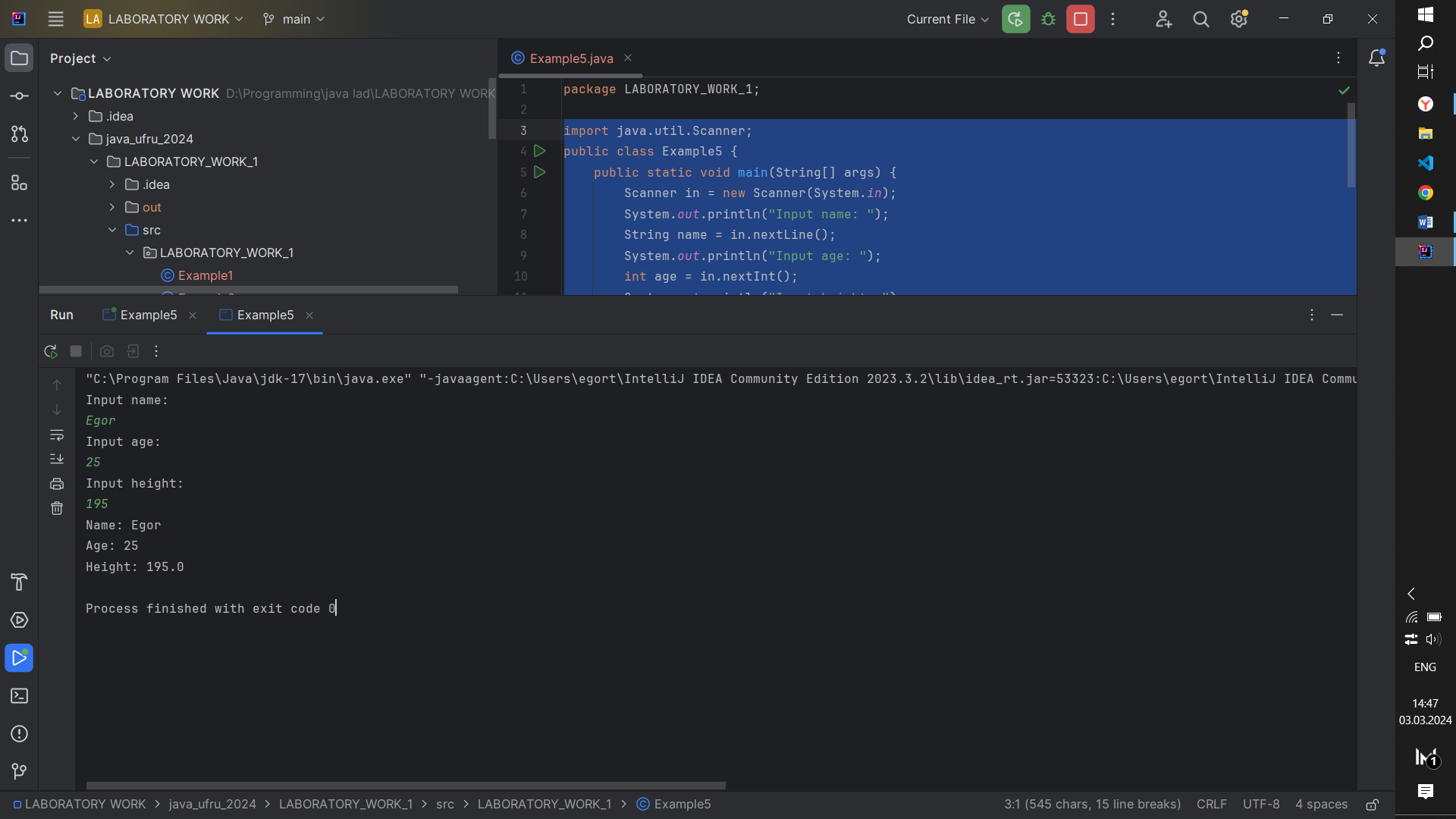
import java.util.Scanner;  
public class Example4 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Input a number:");  
 int num = in.nextInt();  
 System.*out*.println("Your number " + num);  
 in.close();  
 }  
}

**Программа** позволяет пользователю ввести число и выводит его на экран. В теле метода создается объект класса Scanner для чтения данных с консоли. Затем выводится на экран сообщение "Input a number:", и пользователь вводит число. Далее считывается число и сохраняется в переменной num. Затем выводится на экран сообщение "Your number " + num. В конце метода вызывается метод close() для закрытия потока ввода.



Example5

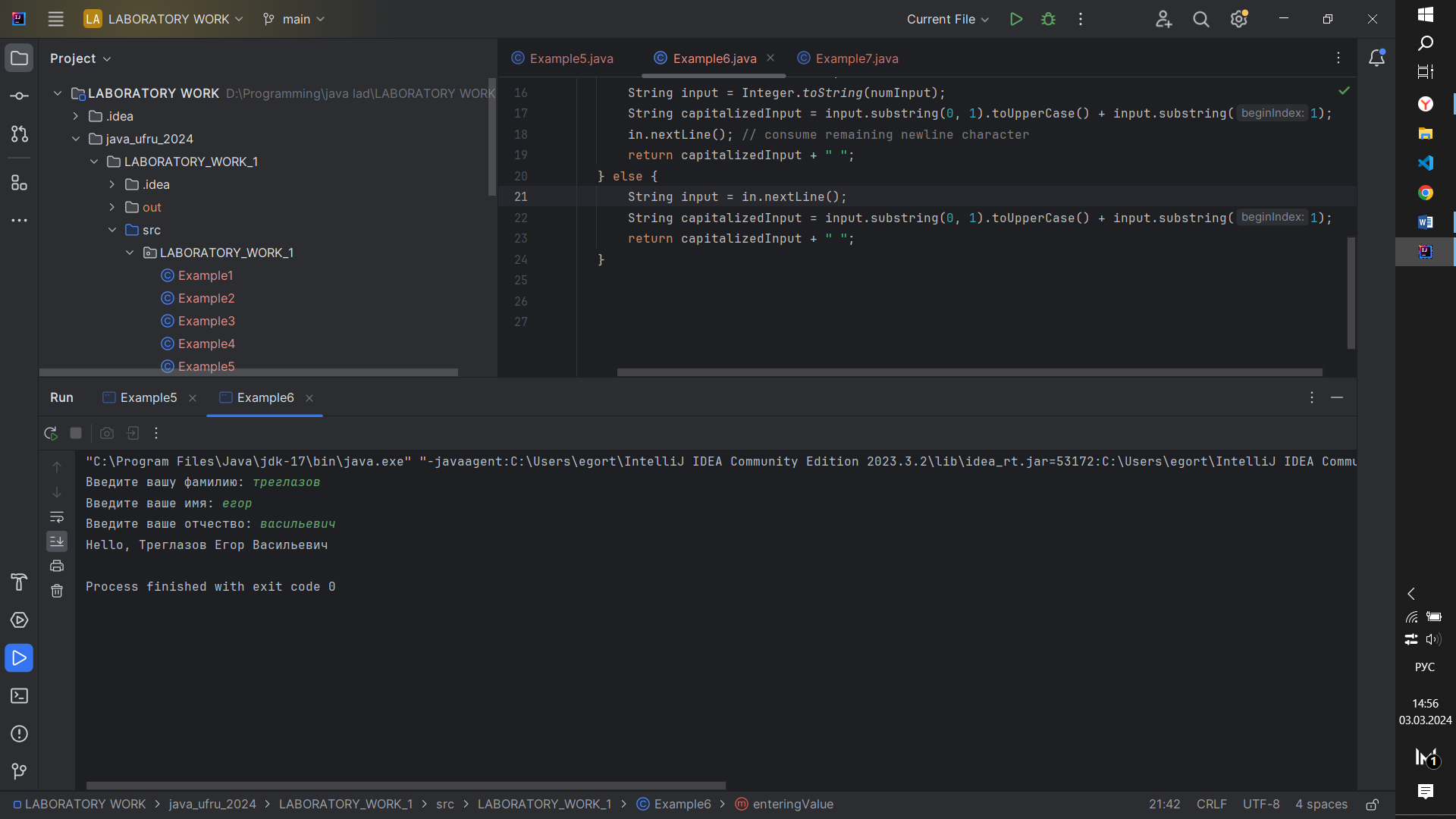
import java.util.Scanner;  
public class Example5 {  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.println("Input name: ");  
 String name = in.nextLine();  
 System.*out*.println("Input age: ");  
 int age = in.nextInt();  
 System.*out*.println("Input height: ");  
 float height = in.nextFloat();  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Height: " + height);  
 in.close();  
 }  
}

**Программа** позволяет пользователю ввести свое имя, возраст и рост, а затем выводит эту информацию на экран. Программа состоит из одного класса Example5, который содержит метод main(). В теле метода создается объект класса Scanner для чтения данных с консоли. Затем выводится на экран сообщение "Input name:", и пользователь вводит свое имя. Далее считывается имя и сохраняется в переменной name. Затем выводится на экран сообщение "Input age:", и пользователь вводит свой возраст. Далее считывается возраст и сохраняется в переменной age. Затем выводится на экран сообщение "Input height:", и пользователь вводит свой рост. Далее считывается рост и сохраняется в переменной height. Затем выводится на экран сообщение "Name: " + name, "Age: " + age и "Height: " + height. В конце метода вызывается метод close() для закрытия потока ввода.

Example6

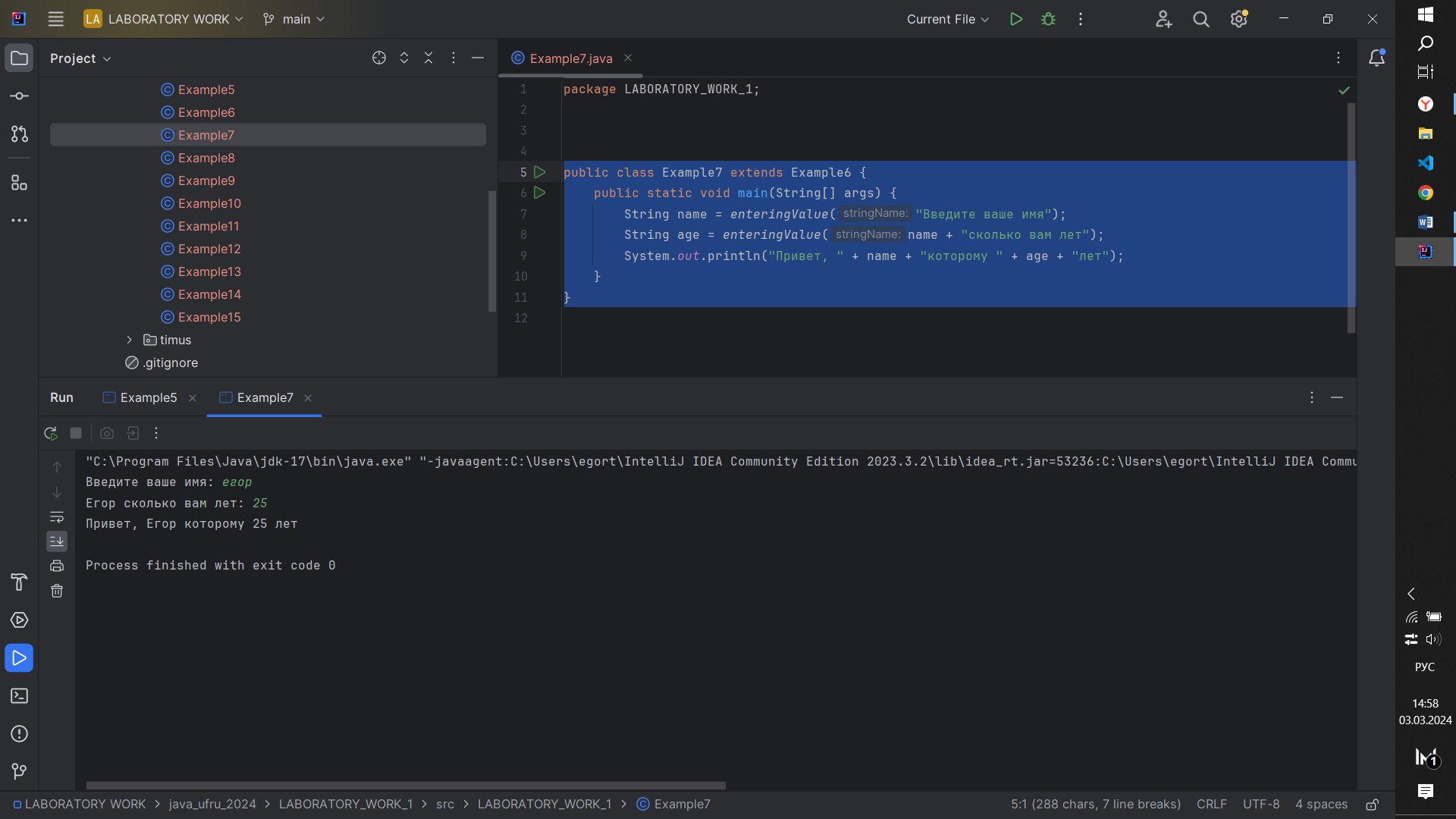
import java.util.Scanner;  
public class Example6 {  
 public static void main(String[] args) {  
 String lastName = *enteringValue*("Введите вашу фамилию");  
 String firstName = *enteringValue*("Введите ваше имя");  
 String middleName = *enteringValue*("Введите ваше отчество");  
 System.*out*.println("Hello, " + lastName + firstName + middleName);  
 }  
 public static String enteringValue(String stringName) {  
 Scanner in = new Scanner(System.*in*);  
 System.*out*.print(stringName + ": ");  
 if (in.hasNextInt()) {  
 int numInput = in.nextInt();  
 String input = Integer.*toString*(numInput);  
 String capitalizedInput = input.substring(0, 1).toUpperCase() + input.substring(1);  
 in.nextLine(); // consume remaining newline character  
 return capitalizedInput + " ";  
 } else {  
 String input = in.nextLine();  
 String capitalizedInput = input.substring(0, 1).toUpperCase() + input.substring(1);  
 return capitalizedInput + " ";  
 }  
 }  
}

**Программа** позволяет пользователю ввести свою фамилию, имя и отчество, а затем выводит приветствие на экран. Программа состоит из одного класса Example6, который содержит метод main() и статический метод enteringValue(). В теле метода main() вызывается метод enteringValue() для ввода фамилии, имени и отчества пользователя. Затем выводится на экран приветствие "Hello, " + фамилия + имя + отчество.

Данный метод будет использовать в последующих классов, как говориться *Don't Repeat Yourself.* Метод enteringValue() использует класс Scanner для чтения данных с консоли. Если пользователь вводит число, то оно преобразуется. Если пользователь вводит строку, то первая буква в строке преобразуется в верхний регистр. Затем возвращается результат в виде строки с пробелом в конце.

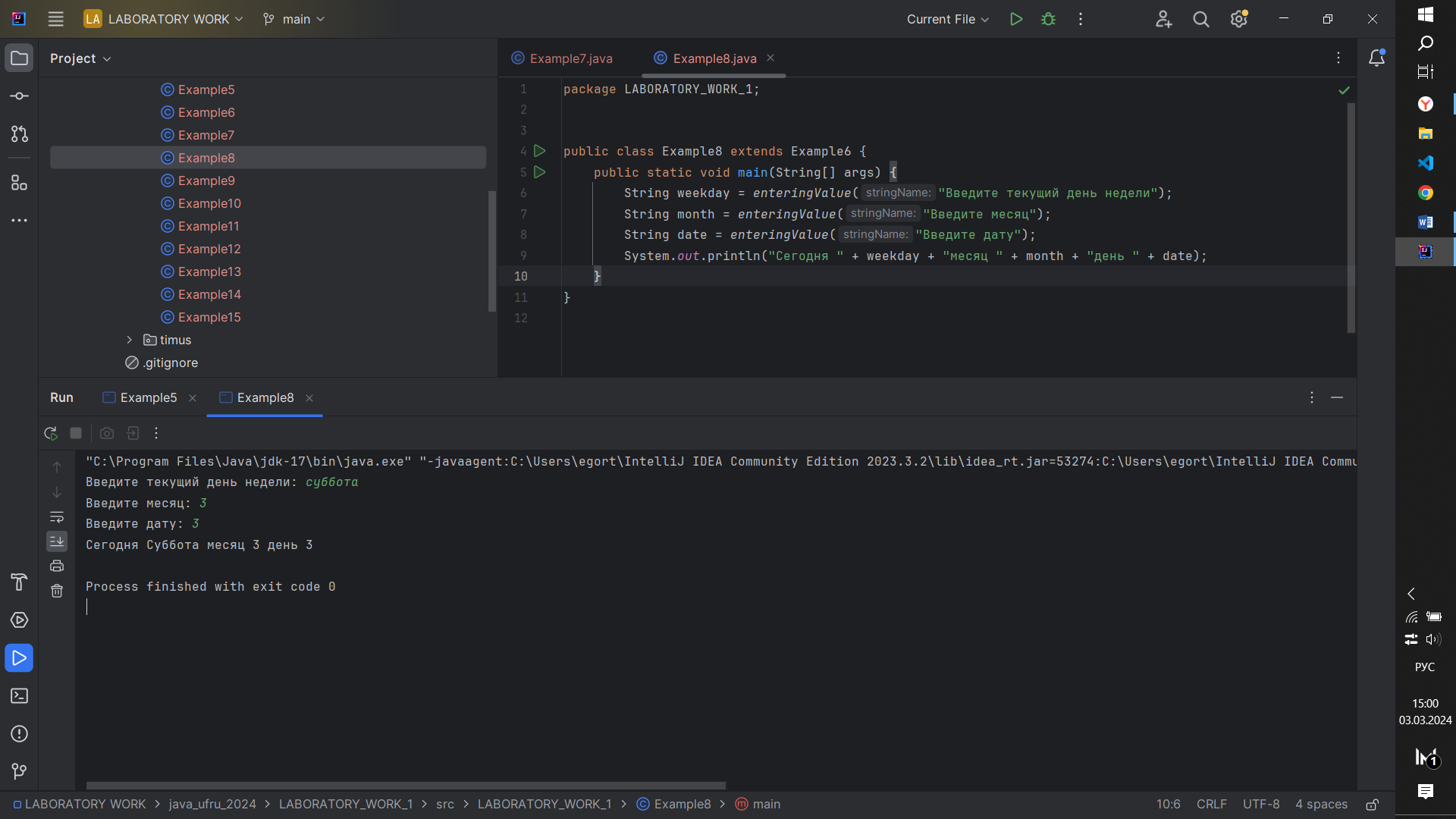
Example7

public class Example7 extends Example6 {  
 public static void main(String[] args) {  
 String name = *enteringValue*("Введите ваше имя");  
 String age = *enteringValue*(name + "сколько вам лет");  
 System.*out*.println("Привет, " + name + "которому " + age + "лет");  
 }  
}

Программа позволяет пользователю ввести свое имя и возраст, а затем выводит приветствие на экран. Программа состоит из одного класса Example7, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода имени и возраста пользователя. Затем выводится на экран приветствие "Привет, " + имя + "которому " + возраст + "лет".

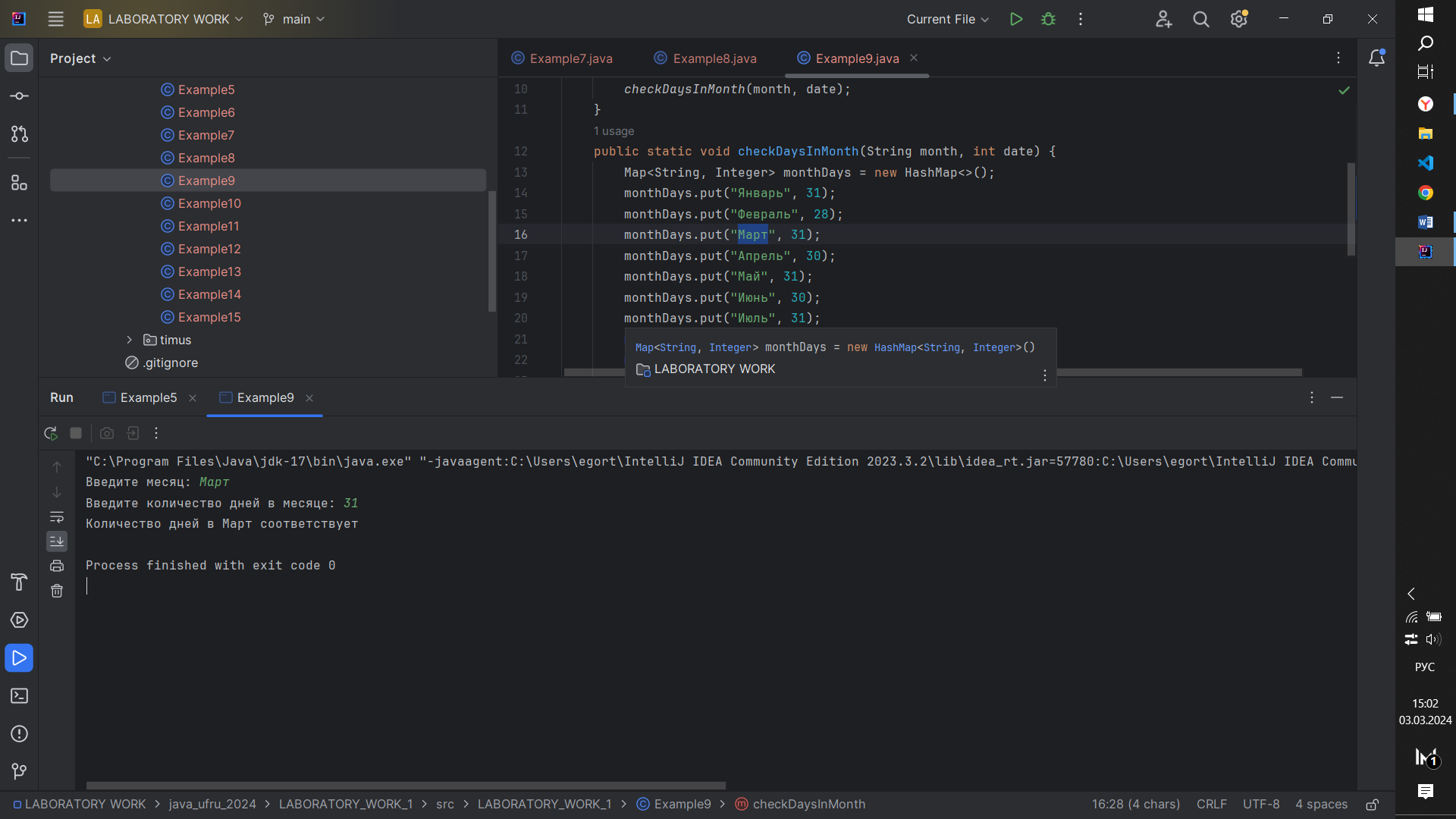
Example8

public class Example8 extends Example6 {  
 public static void main(String[] args) {  
 String weekday = *enteringValue*("Введите текущий день недели");  
 String month = *enteringValue*("Введите месяц");  
 String date = *enteringValue*("Введите дату");  
 System.*out*.println("Сегодня " + weekday + "месяц " + month + "день " + date);  
 }  
}

Программа позволяет пользователю ввести текущий день недели, месяц и дату, а затем выводит информацию о сегодняшнем дне на экран. Программа состоит из одного класса Example8, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода дня недели, месяца и даты. Затем выводится на экран сообщение "Сегодня " + день недели + "месяц " + месяц + "день " + дата.

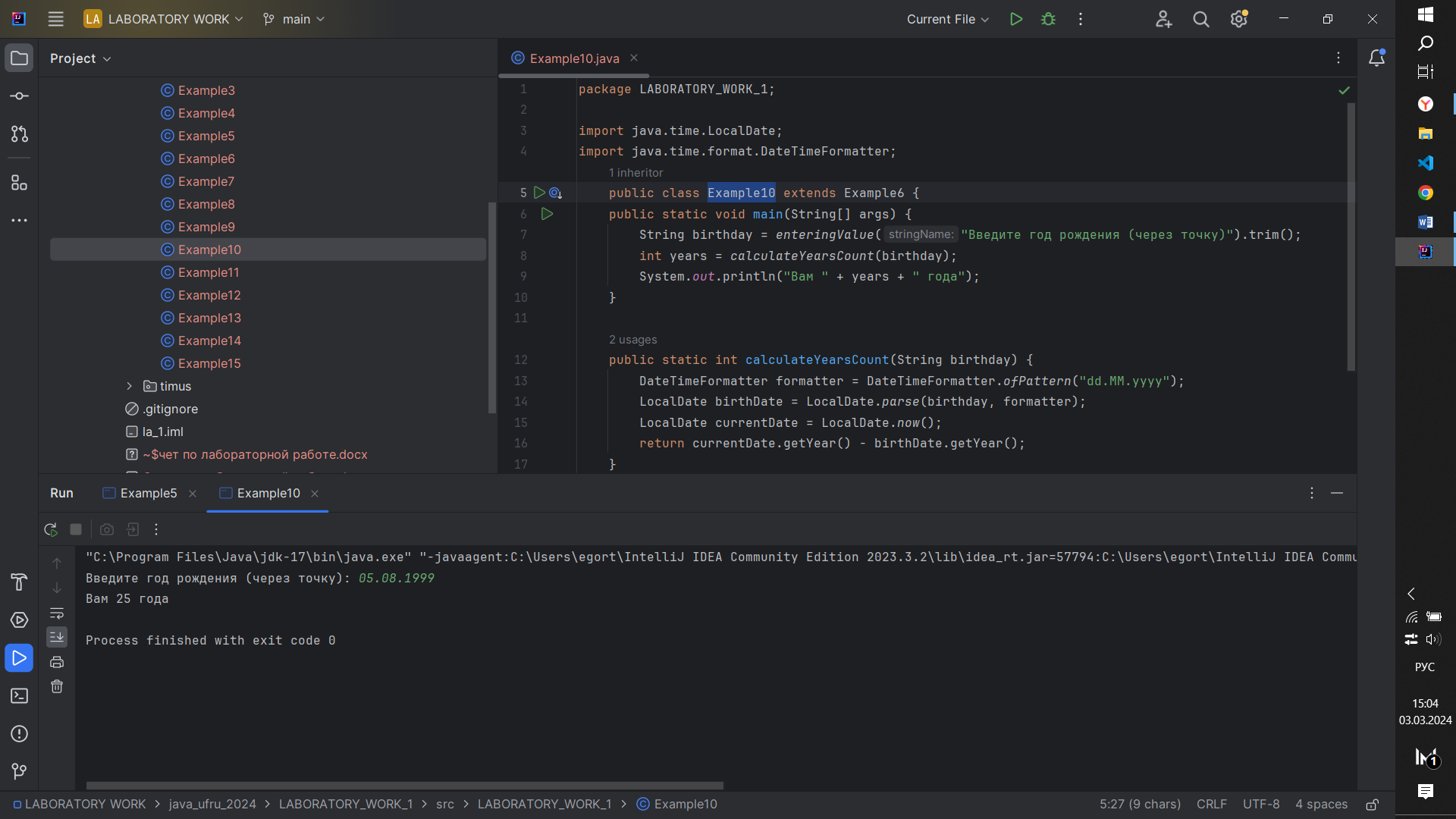
Example9

public class Example9 extends Example6 {  
 public static void main(String[] args) {  
 String month = *enteringValue*("Введите месяц").trim();  
 int date = Integer.*parseInt*(*enteringValue*("Введите количество дней в месяце").trim());  
 *checkDaysInMonth*(month, date);  
 }  
 public static void checkDaysInMonth(String month, int date) {  
 Map<String, Integer> monthDays = new HashMap<>();  
 monthDays.put("Январь", 31);  
 monthDays.put("Февраль", 28);  
 monthDays.put("Март", 31);  
 monthDays.put("Апрель", 30);  
 monthDays.put("Май", 31);  
 monthDays.put("Июнь", 30);  
 monthDays.put("Июль", 31);  
 monthDays.put("Август", 31);  
 monthDays.put("Сентябрь", 30);  
 monthDays.put("Октябрь", 31);  
 monthDays.put("Ноябрь", 30);  
 monthDays.put("Декабрь", 31);  
  
 if (monthDays.containsKey(month) && monthDays.get(month) == date) {  
 System.*out*.println("Количество дней в "+ month + " соответствует");  
 } else {  
 System.*out*.println("Количество дней в "+ month + " НЕ соответствует");  
 }  
  
 }  
}

Программа позволяет пользователю ввести название месяца и количество дней в нем, а затем проверяет, соответствует ли количество дней в месяце действительности. Программа состоит из одного класса Example9, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода названия месяца и количества дней в нем. Затем вызывается метод checkDaysInMonth() для проверки соответствия количества дней в месяце действительности.

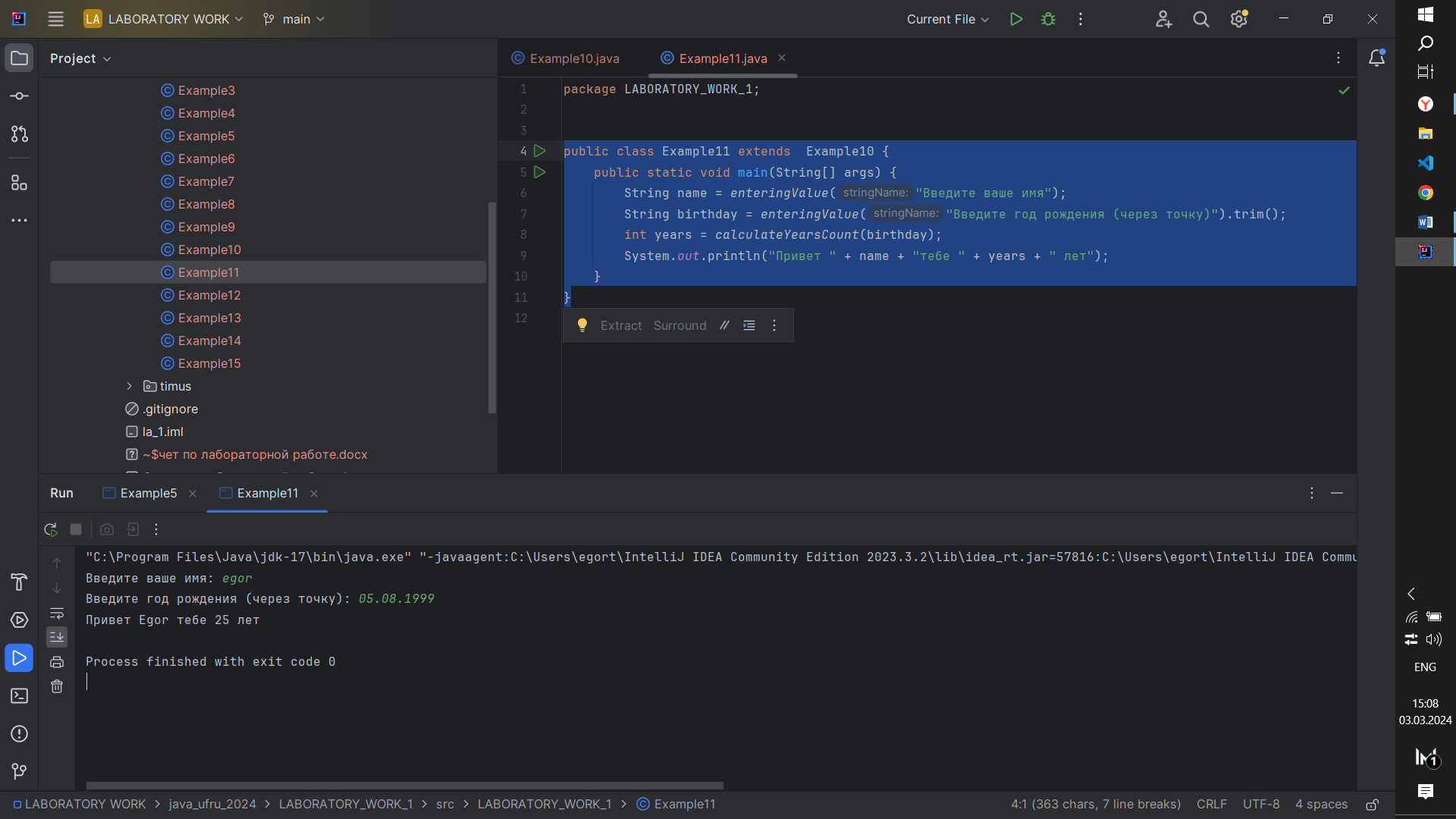
Example10

public class Example10 extends Example6 {  
 public static void main(String[] args) {  
 String birthday = *enteringValue*("Введите год рождения (через точку)").trim();  
 int years = *calculateYearsCount*(birthday);  
 System.*out*.println("Вам " + years + " года");  
 }  
  
 public static int calculateYearsCount(String birthday) {  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("dd.MM.yyyy");  
 LocalDate birthDate = LocalDate.*parse*(birthday, formatter);  
 LocalDate currentDate = LocalDate.*now*();  
 return currentDate.getYear() - birthDate.getYear();  
 }  
}

Программа позволяет пользователю ввести год своего рождения в формате "день.месяц.год", а затем вычисляет количество полных лет, прожитых пользователем. Программа состоит из одного класса Example10, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода года рождения пользователя. Затем вызывается метод calculateYearsCount() для вычисления количества полных лет, прожитых пользователем. Затем выводится на экран сообщение "Вам " + количество полных лет + " лет".

Example11

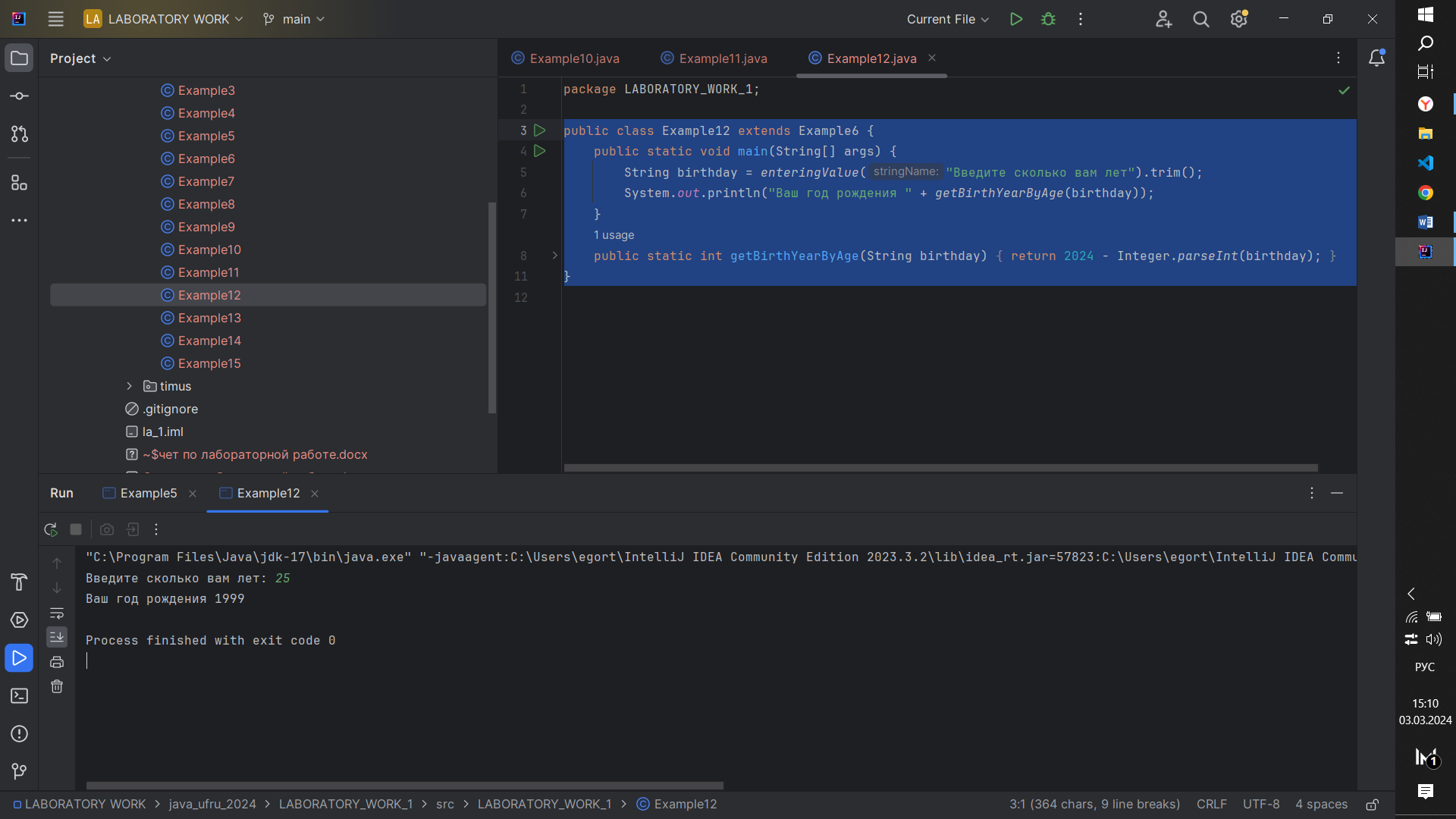
public class Example11 extends Example10 {  
 public static void main(String[] args) {  
 String name = *enteringValue*("Введите ваше имя");  
 String birthday = *enteringValue*("Введите год рождения (через точку)").trim();  
 int years = *calculateYearsCount*(birthday);  
 System.*out*.println("Привет " + name + "тебе " + years + " лет");  
 }  
}

Программа позволяет пользователю ввести свое имя и год рождения в формате "день.месяц.год", а затем вычисляет количество полных лет, прожитых пользователем, и выводит приветствие на экран. Программа состоит из одного класса Example11, который наследуется от класса Example10> Example6. В теле метода main() вызывается метод enteringValue() для ввода имени и года рождения пользователя. Затем вызывается метод calculateYearsCount() для вычисления количества полных лет, прожитых пользователем. Затем выводится на экран сообщение "Привет " + имя + "тебе " + количество полных лет + " лет".

Example12

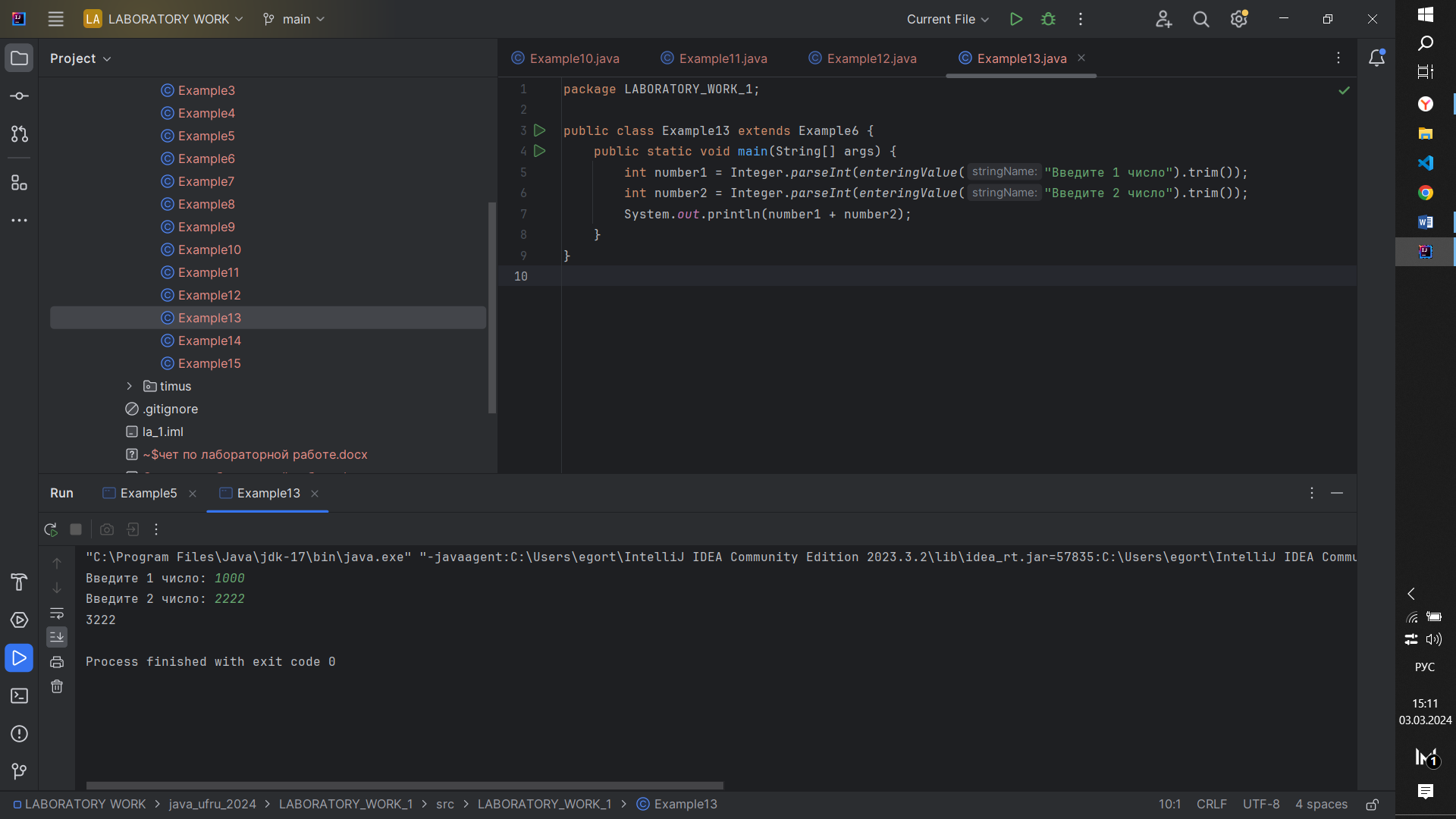
public class Example12 extends Example6 {  
 public static void main(String[] args) {  
 String birthday = *enteringValue*("Введите сколько вам лет").trim();  
 System.*out*.println("Ваш год рождения " + *getBirthYearByAge*(birthday));  
 }  
 public static int getBirthYearByAge(String birthday) {  
 return 2024 - Integer.*parseInt*(birthday);  
 }  
}

Программа позволяет пользователю ввести свой возраст, а затем вычисляет год рождения пользователя и выводит его на экран. Программа состоит из одного класса Example12, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода возраста пользователя. Затем вызывается метод getBirthYearByAge() для вычисления года рождения пользователя. Затем выводится на экран сообщение "Ваш год рождения " + год рождения.



Example13

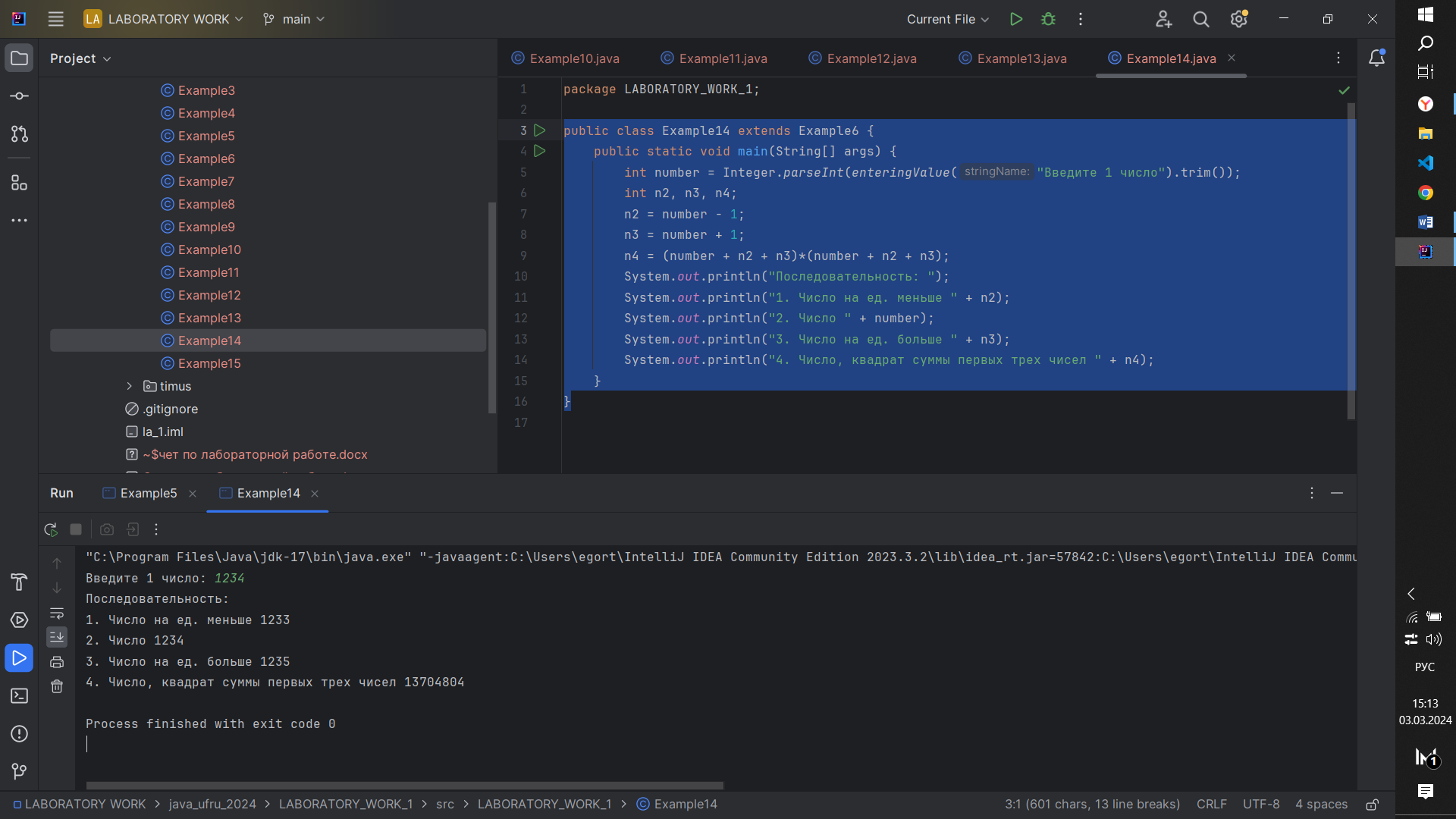
public class Example13 extends Example6 {  
 public static void main(String[] args) {  
 int number1 = Integer.*parseInt*(*enteringValue*("Введите 1 число").trim());  
 int number2 = Integer.*parseInt*(*enteringValue*("Введите 2 число").trim());  
 System.*out*.println(number1 + number2);  
 }  
}

Программа позволяет пользователю ввести два числа, а затем выводит их сумму на экран. Программа состоит из одного класса Example13, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода двух чисел. Затем вызывается метод Integer.parseInt() для преобразования строки в число. Затем выводится на экран сумма двух чисел.

Example14

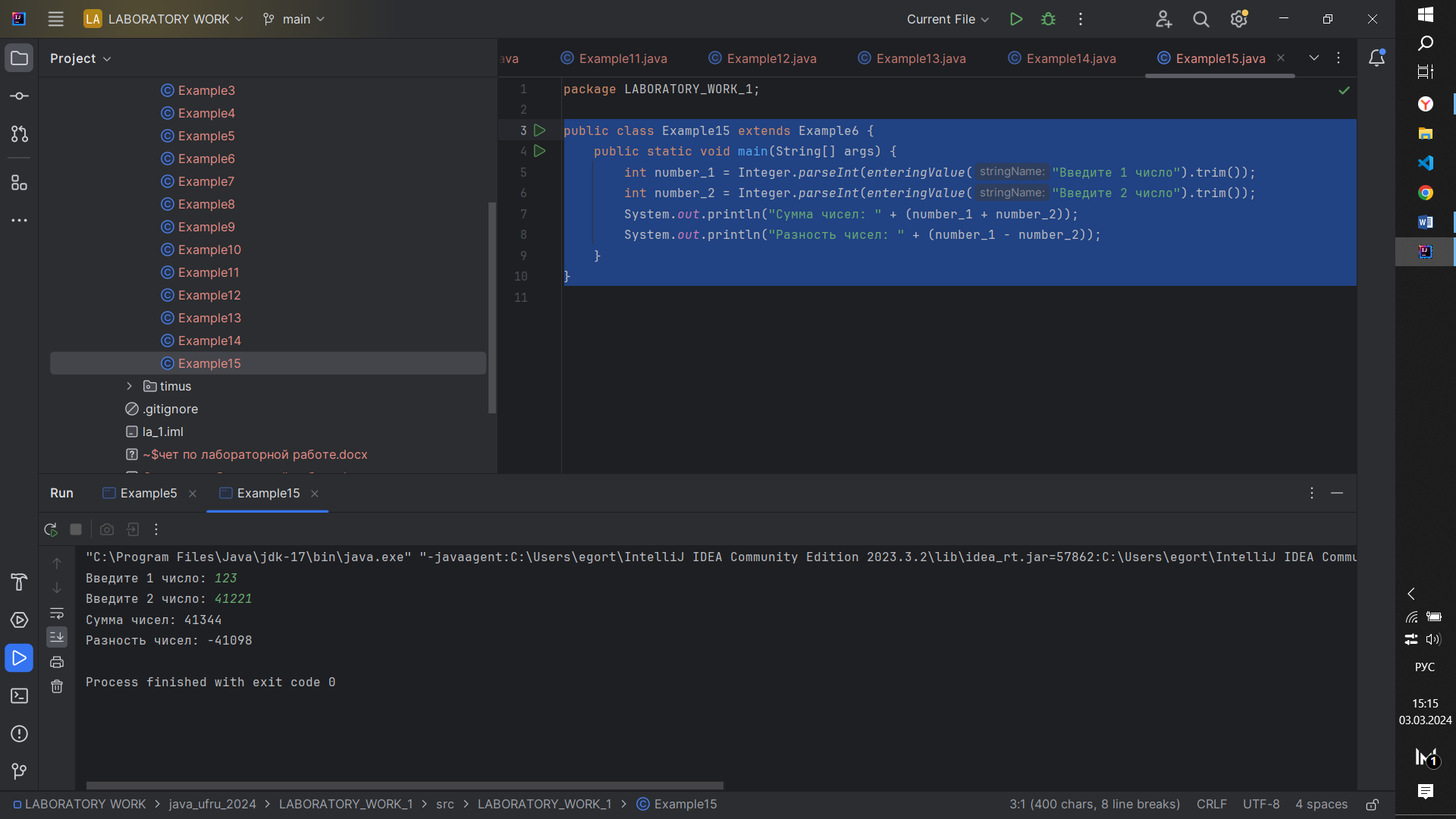
public class Example14 extends Example6 {  
 public static void main(String[] args) {  
 int number = Integer.*parseInt*(*enteringValue*("Введите 1 число").trim());  
 int n2, n3, n4;  
 n2 = number - 1;  
 n3 = number + 1;  
 n4 = (number + n2 + n3)\*(number + n2 + n3);  
 System.*out*.println("Последовательность: ");  
 System.*out*.println("1. Число на ед. меньше " + n2);  
 System.*out*.println("2. Число " + number);  
 System.*out*.println("3. Число на ед. больше " + n3);  
 System.*out*.println("4. Число, квадрат суммы первых трех чисел " + n4);  
 }  
}

Программа позволяет пользователю ввести одно число, а затем вычисляет последовательность чисел и выводит ее на экран. Программа состоит из одного класса Example14, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода числа. Затем вычисляются числа, которые составляют последовательность. Затем выводится на экран последовательность чисел.



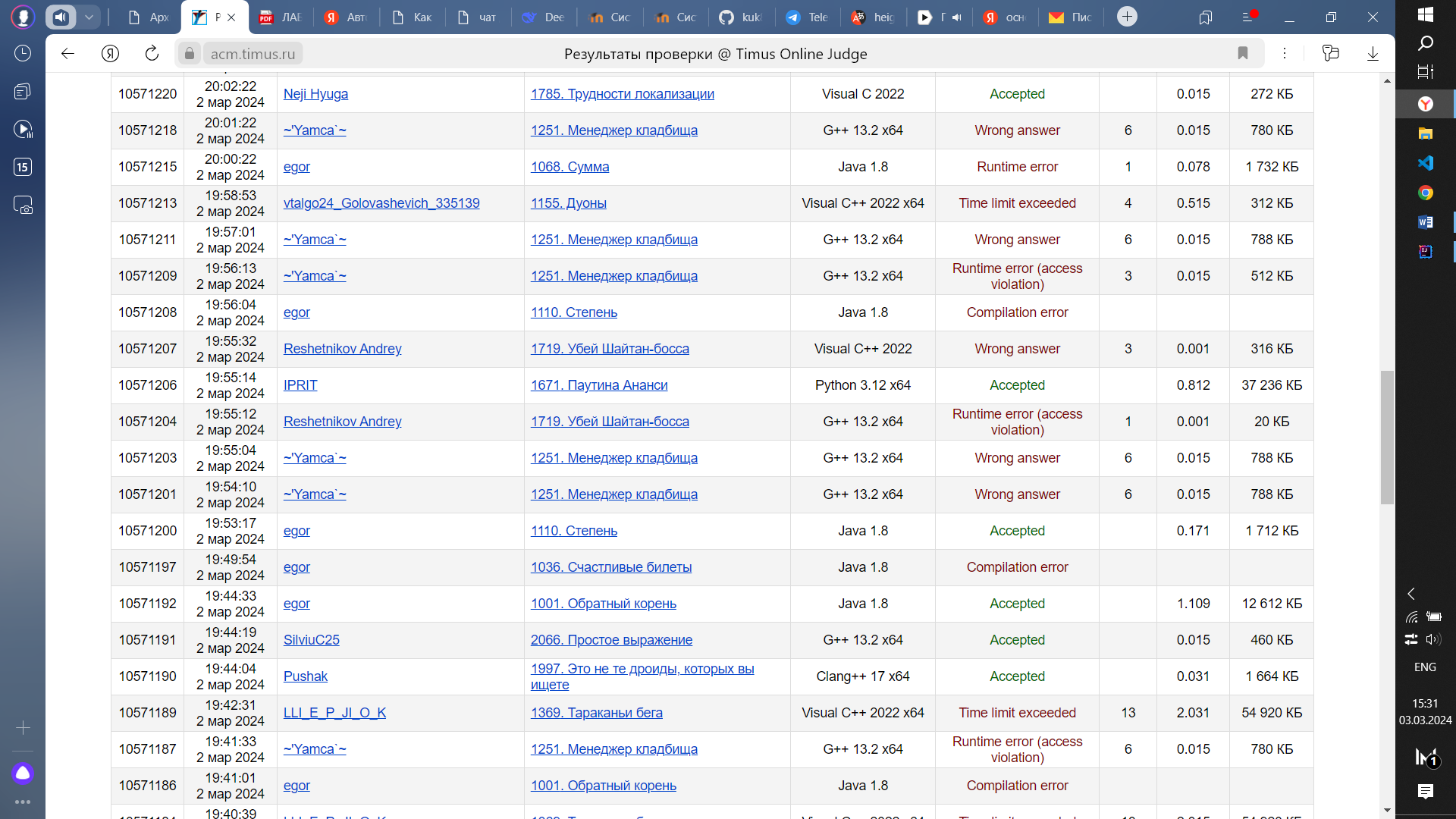
Example15

public class Example15 extends Example6 {  
 public static void main(String[] args) {  
 int number\_1 = Integer.*parseInt*(*enteringValue*("Введите 1 число").trim());  
 int number\_2 = Integer.*parseInt*(*enteringValue*("Введите 2 число").trim());  
 System.*out*.println("Сумма чисел: " + (number\_1 + number\_2));  
 System.*out*.println("Разность чисел: " + (number\_1 - number\_2));  
 }  
}

Программа позволяет пользователю ввести два числа, а затем вычисляет их сумму и разность и выводит результаты на экран. Программа состоит из одного класса Example15, который наследуется от класса Example6. В теле метода main() вызывается метод enteringValue() для ввода двух чисел. Затем вызывается метод Integer.parseInt() для преобразования строки в число. Затем вычисляются сумма и разность двух чисел. Затем выводится на экран результаты вычислений.

1110. Степень

Ограничение времени: 0.5 секунды  
Ограничение памяти: 64 МБ



Даны целые числа *N*, *M* и *Y*. Напишите программу, которая найдёт все целые числа *X* в диапазоне [0, *M* – 1], такие что *XN* mod *M* = *Y*.

**Исходные данные**

Ввод содержит единственную строку с числами *N*, *M* и *Y* (0 < *N* < 999, 1 < *M* < 999, 0 < *Y* < 999), записанными через пробел.

**Результат**

Выведите все числа *X* через пробел в одной строке. Числа должны идти в порядке возрастания. Если искомых чисел нет, выведите −1.

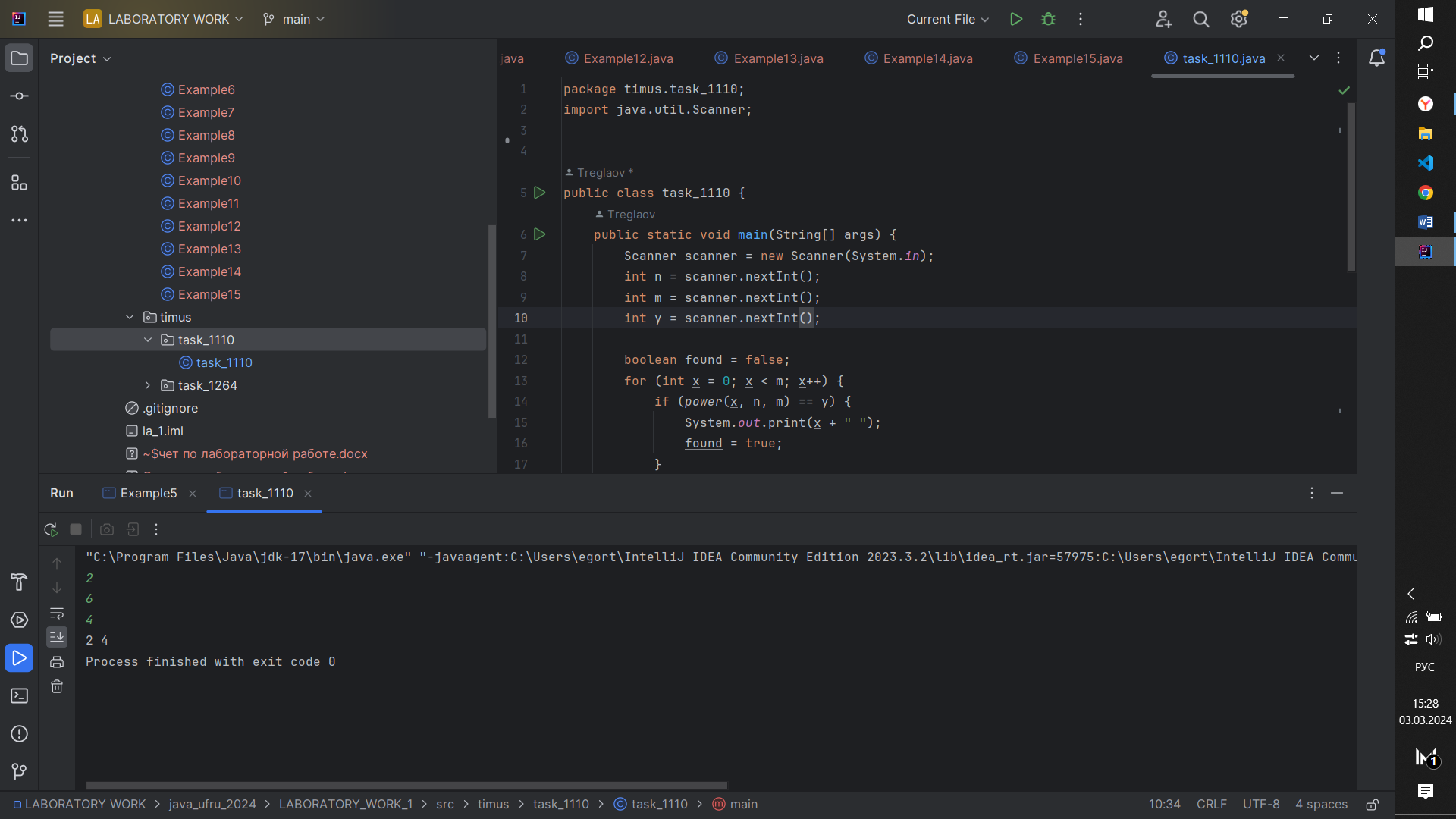
**Пример**

|  |  |
| --- | --- |
| **исходные данные** | **результат** |
| 2 6 4 | 2 4 |

public class task\_1110 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int n = scanner.nextInt();  
 int m = scanner.nextInt();  
 int y = scanner.nextInt();  
  
 boolean found = false;  
 for (int x = 0; x < m; x++) {  
 if (*power*(x, n, m) == y) {  
 System.*out*.print(x + " ");  
 found = true;  
 }  
 }  
  
 if (!found) {  
 System.*out*.println(-1);  
 }  
 }  
  
 private static int power(int x, int n, int m) {  
 int result = 1;  
 for (int i = 0; i < n; i++) {  
 result = (result \* x) % m;  
 }  
 return result;  
 }  
}

Программа ищет все числа x, для которых выполняется равенство power(x, n, m) = y, где power(x, n, m) - это остаток от деления x в степени n на m. Программа состоит из одного класса task\_1110. В теле метода main() вызывается класс Scanner для чтения данных с консоли. Затем вызывается метод power() для вычисления остатка от деления x в степени n на m. Затем выводится на экран результаты вычислений.

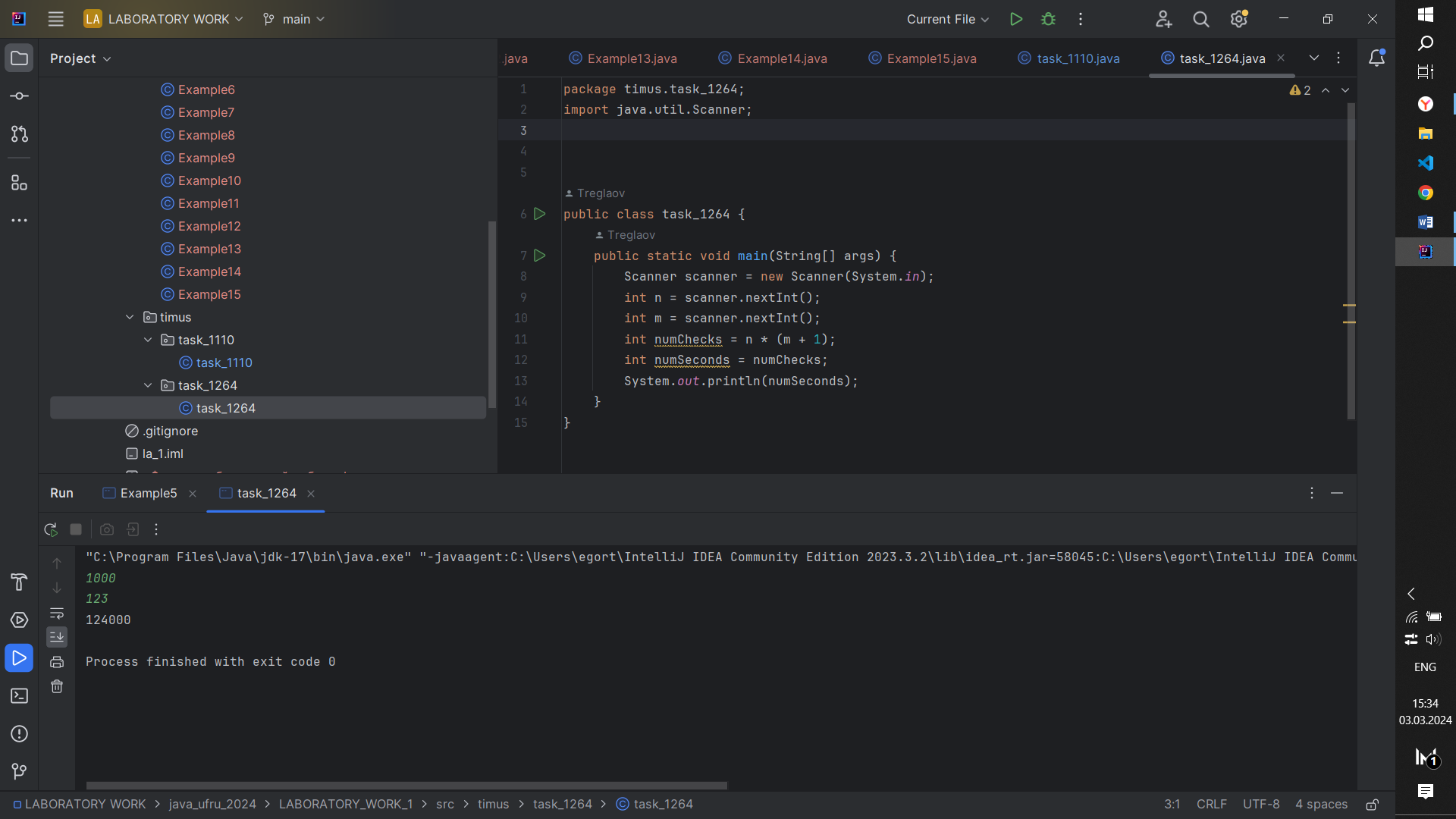
Метод power() использует цикл for для вычисления остатка от деления x в степени n на m. Метод main() использует цикл for для проверки всех возможных значений x. Если найдено значение x, для которого выполняется равенство power(x, n, m) = y, то выводится на экран это значение x. Если такого значения x не найдено, то выводится на экран -1.



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1264. Трудовые будни  Ограничение времени: 1.0 секунды Ограничение памяти: 64 МБ    После успеха предыдущей программы Васечкина, позволившей подвести итоги выборов всего за два дня, Васечкин был назначен начальником отдела. Не правда ли, успех? В данный момент Артемий Сидорович готовит техническое задание для своего подчиненного — программиста Петечкина. Задание заключается в написании крайне полезной функции, которая намного облегчит жизнь всем программистам отдела. Для каждого числа от 0 до *M* функция будет подсчитывать число раз, которое число встречается в *N*-элементном массиве. Артемий Сидорович полагает, что функция должна работать следующим образом (пример кода для *N* = 3, *M* = 1):   |  |  | | --- | --- | | **C** | **Pascal** | | if (arr[0]==0) ++count[0];  if (arr[0]==1) ++count[1];  if (arr[1]==0) ++count[0];  if (arr[1]==1) ++count[1];  if (arr[2]==0) ++count[0];  if (arr[2]==1) ++count[1]; | if arr[0]=0 then count[0] := count[0] + 1;  if arr[0]=1 then count[1] := count[1] + 1;  if arr[1]=0 then count[0] := count[0] + 1;  if arr[1]=1 then count[1] := count[1] + 1;  if arr[2]=0 then count[0] := count[0] + 1;  if arr[2]=1 then count[1] := count[1] + 1; |   Артемий Сидорович хочет оценить время, за которое Петечкин справится с заданием. Известно, что Петечкин пишет одну строчку кода за одну секунду (не правда ли, очень быстро?). Артемий Сидорович точно не знает, какими *M* и *N* можно ограничиться. Ваша цель — написать программу, которая подсчитает число секунд, которые потребуется Петечкину на написание кода.  **Исходные данные**  Единственная строка содержит целые числа *N* (0 ≤ *N* ≤ 40000) и *M* (0 ≤ *M* ≤ 40000).  **Результат**  Выведите число секунд, требуемых Петечкину на написание функции.  **Пример**   |  |  | | --- | --- | | **исходные данные** | **результат** | | 3 1 | 6 | |

public class task\_1264 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int n = scanner.nextInt();  
 int m = scanner.nextInt();  
 int numChecks = n \* (m + 1);  
 int numSeconds = numChecks;  
 System.*out*.println(numSeconds);  
 }  
}

Программа вычисляет количество секунд, которое потребуется для проверки n чисел на делимость на m чисел. Программа состоит из одного класса task\_1264. В теле метода main() вызывается класс Scanner для чтения данных с консоли. Затем вычисляется количество проверок, которое равно n \* (m + 1). Затем вычисляется количество секунд, которое равно количеству проверок. Затем выводится на экран количество секунд.



Ответы на вопросы из 5 раздела.

1. Будет ошибка. Реременная teenyBit объявлена внутри блока кода, который находится внутри блока кода, где объявлена переменная bitesOfCheese. Поэтому она недоступна вне этого блока кода.

public class SomeScopeExample {  
 public void eatIfHungry(boolean hungry) {  
 if (hungry) {  
 int bitesOfCheese = 1;  
 {  
 boolean teenyBit = true;  
 System.*out*.println(bitesOfCheese);  
 }  
 // Переменная teenyBit недоступна здесь  
 }  
 // Переменная teenyBit недоступна здесь  
 }  
}

Можно исправить код, переменная teenyBit объявлена перед блоком кода, где объявлена переменная bitesOfCheese. Поэтому она доступна для использования внутри этого блока кода и вне его.

public class SomeScopeExample {  
 public void eatIfHungry(boolean hungry) {  
 boolean teenyBit;  
 if (hungry) {  
 int bitesOfCheese = 1;  
 teenyBit = true;  
 System.*out*.println(bitesOfCheese);  
 }  
 // Переменная teenyBit доступна здесь  
 System.*out*.println(teenyBit);  
 }  
}

1. В первом куске кода ошибки нет, потому что переменная bird типа short инициализируется значением 'd', которое является символом, а не числом.

Второй кусок кода даст ошибку, потому что переменная fish типа char инициализируется значением (short)65535, которое является числом, а не символом. В Java символы хранятся в типе char, а не в типе short. Поэтому инициализация переменной fish типа char значением (short)65535 вызовет ошибку компиляции.

short reptile = 65535;  
char fish = (short)-1;  
char fish = -1;

1. Код не будет компилироваться из-за переполнения целочисленного типа long. Значение 3123456789 слишком велико для типа long, который может хранить значения в диапазоне от -2^63 до 2^63-1.

Чтобы исправить ошибку, необходимо использовать тип BigInteger, который может хранить произвольно большие целые числа:

BigInteger max = new BigInteger("3123456789");

1. **Будет компилироваться:**

List<Integer> heights = new ArrayList<>();  
heights.add(null);  
String s = null;

**Не будет компилироваться:**

int value = null;

**Объяснение:**

* int - это примитивный тип данных, который не может принимать значение null.
* List<Integer> и String - ссылочные типы данных, которые могут принимать значение null, указывающее на отсутствие объекта.
* heights.add(null) добавляет в список null, что допустимо для ссылочных типов данных

1. B. Переменная класса типа boolean по умолчанию имеет значение false.

E. Переменная класса типа long по умолчанию имеет значение 0L

F. Классовая переменная типа long имеет значение по умолчанию 0

1. A. Локальная переменная типа boolean по умолчанию не имеет значения, поэтому ее нельзя использовать до инициализации.

E. Локальная переменная типа boolean по умолчанию имеет значение false.

1. E. Переменная экземпляра типа String имеет значение по умолчанию null.

F. Переменная экземпляра типа String по умолчанию принимает значение пустой строки "".

1. …
2. B. \_helloWorld$

C. true

E. Public

F. \_1980\_s

G. \_Q2\_

1. true

true

true

false

true

true

1. Ошибки в коде:

Ошибка в объявлении массива floatArray. Не нужно указывать тип данных float перед фигурными скобками.

Ошибка в инициализации массива charArray. Необходимо указать размер массива в фигурных скобках.

Исправленный код:

int[] intArray = {3, 4, 5}; float[] floatArray = {7.2f, 1.8f, 9.3f}; char[] charArray = new char[2];

Цикл, который будет итерироваться по массиву с помощью цикла for:

for (int i = 0; i < intArray.length; i++) { System.out.println(intArray[i]); }

Цикл, который будет итерироваться по массиву с помощью метода forEach (для массива intArray):

Arrays.stream(intArray).forEach(System.out::println);