Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 7.3**»

Выполнил: Сергеева Агата Андреевна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2024 г.

**Задание 7(1):**

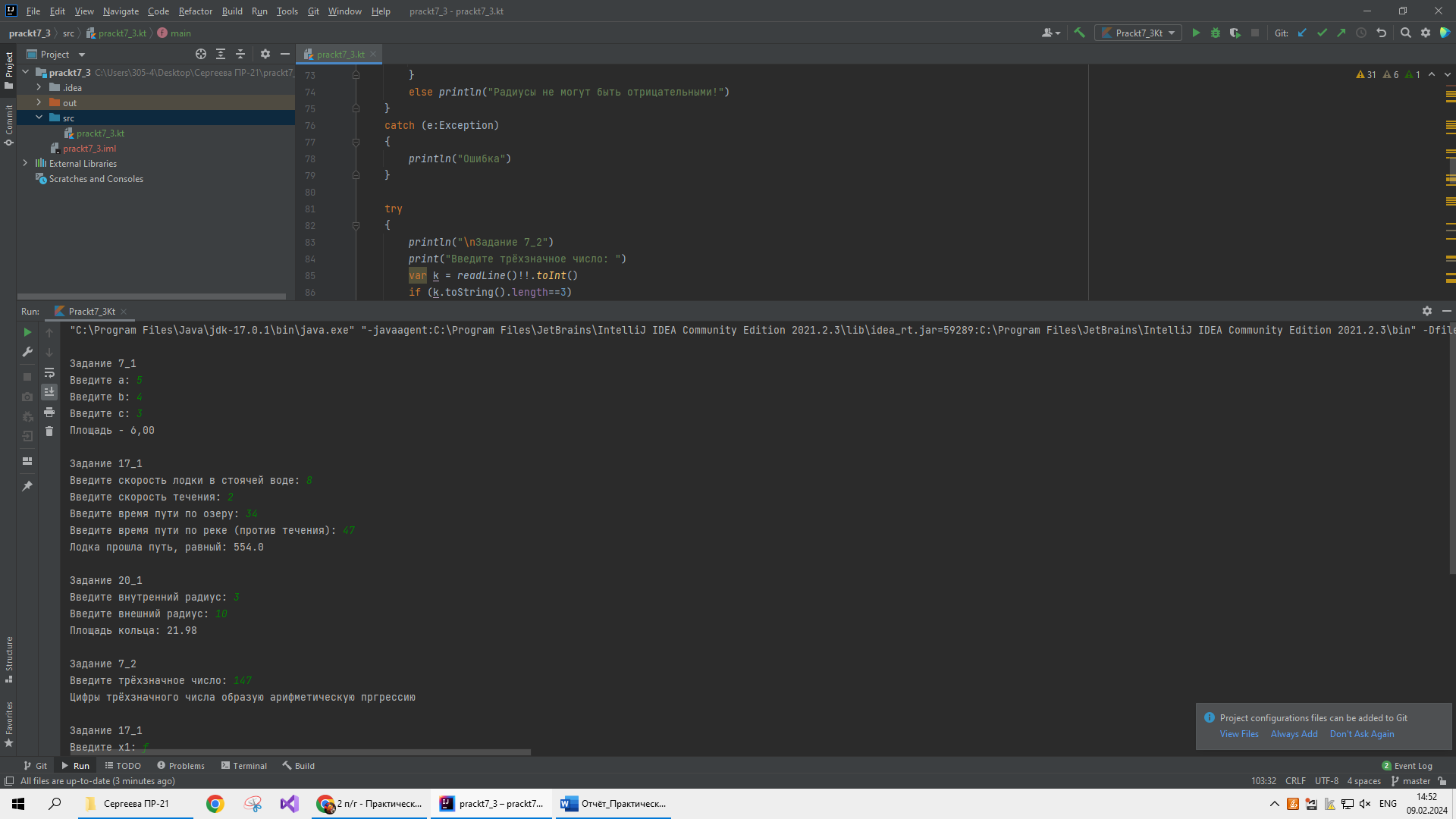
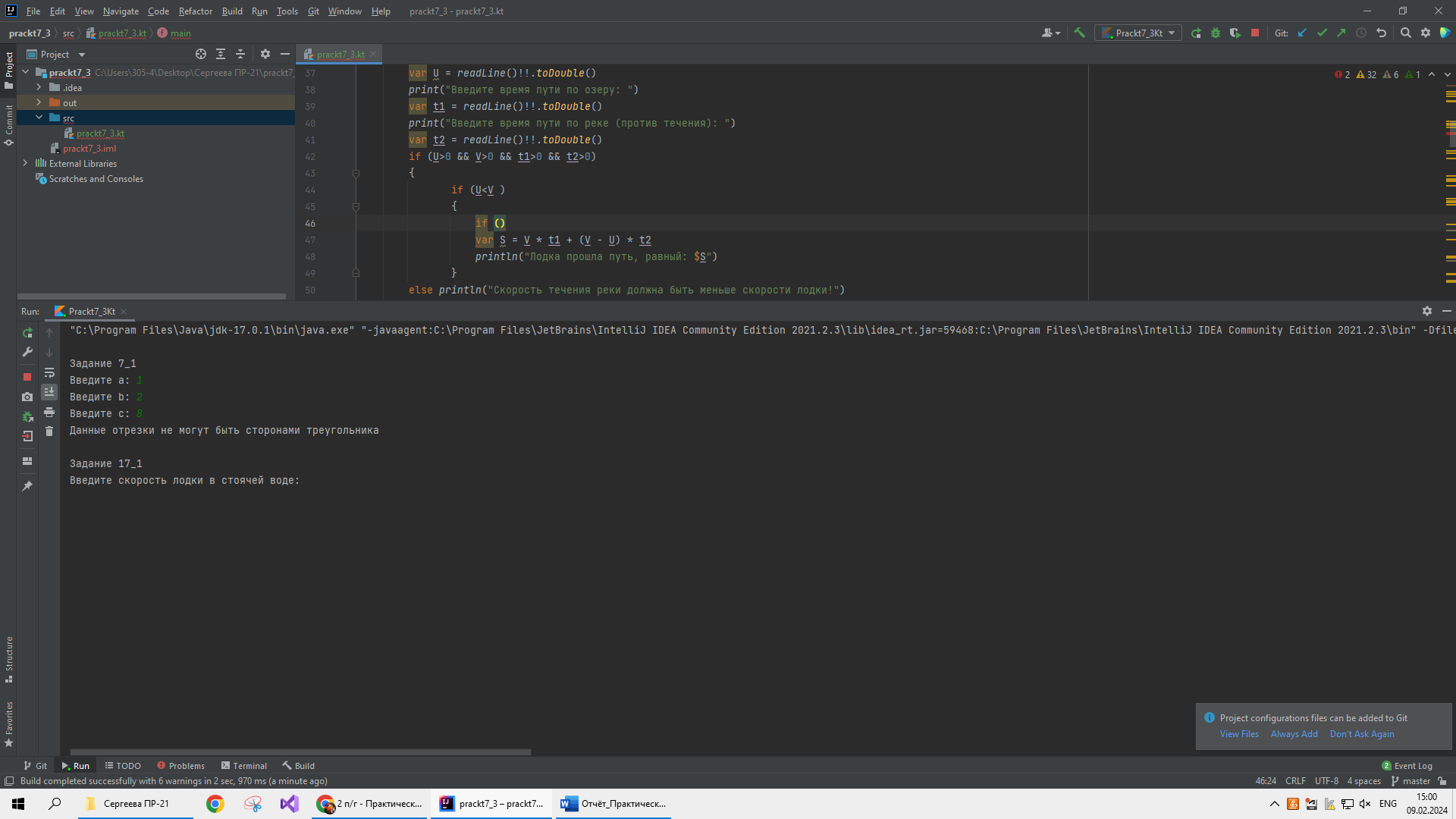
**Входные данные:**

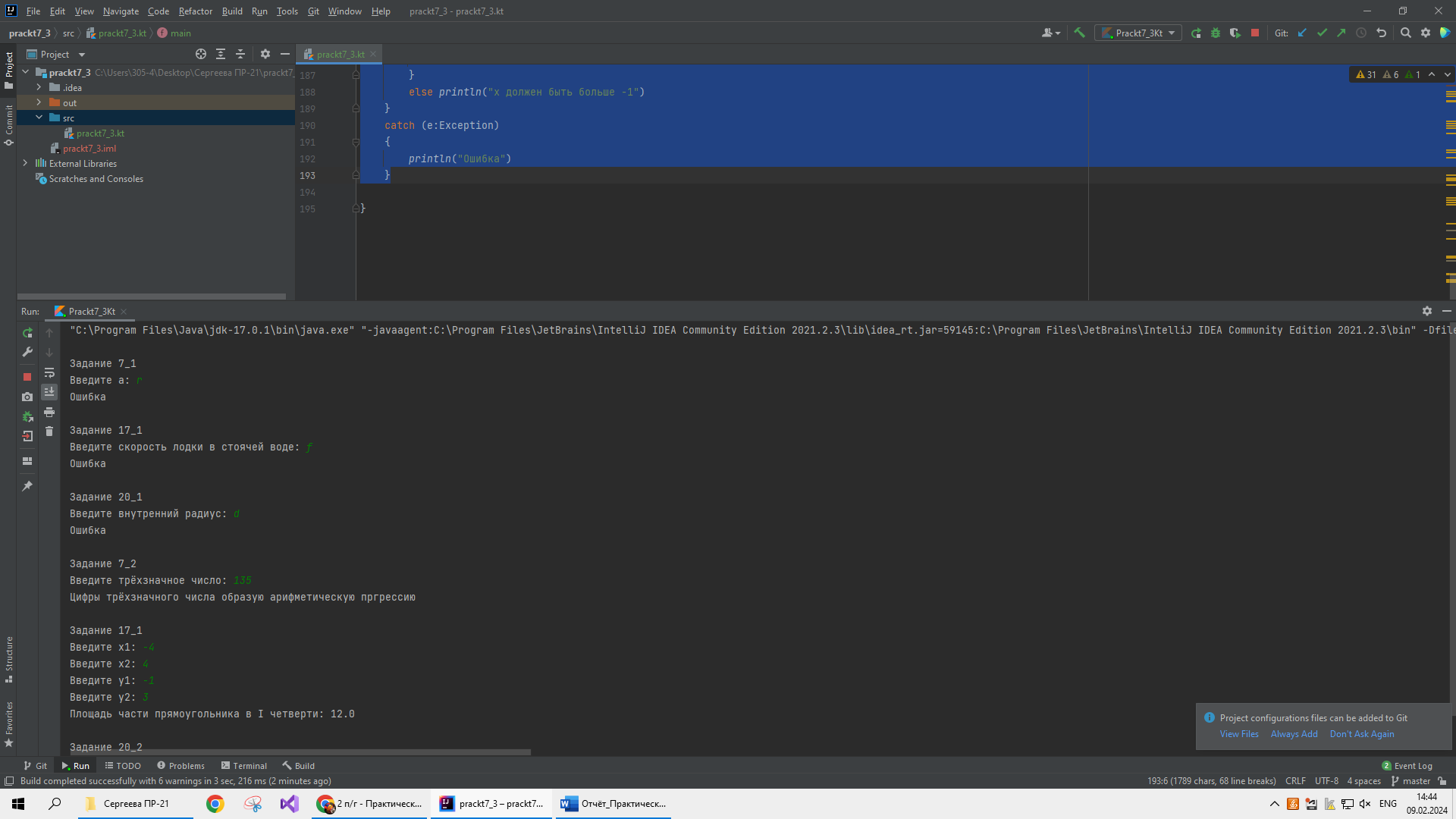
**Выходные данные:**

**Блок-схема:**

**Листинг программы:**

try  
{  
 *println*("\nЗадание 7\_1")  
 *print*("Введите a: ")  
 var a = *readLine*()!!.*toDouble*()  
 *print*("Введите b: ")  
 var b = *readLine*()!!.*toDouble*()  
 *print*("Введите c: ")  
 var c = *readLine*()!!.*toDouble*()  
 if (a>0&&b>0&&c>0) {  
 when {  
 (a + b) > c && (a + c) > b && (b + c) > a -> {  
 var pp = (a+b+c) / 2  
 var s = String.*format*("%.2f", (Math.sqrt(pp \* (pp - a) \* (pp - b) \* (pp - c))))  
 *println*("Площадь - $s")  
 }  
  
 else -> *println*("Данные отрезки не могут быть сторонами треугольника")  
 }  
 }  
 else *println*("Проверьте данные, которые вы ввели. Среди них есть отрицательное")  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**



**Задание 17(1):**

**Входные данные:**

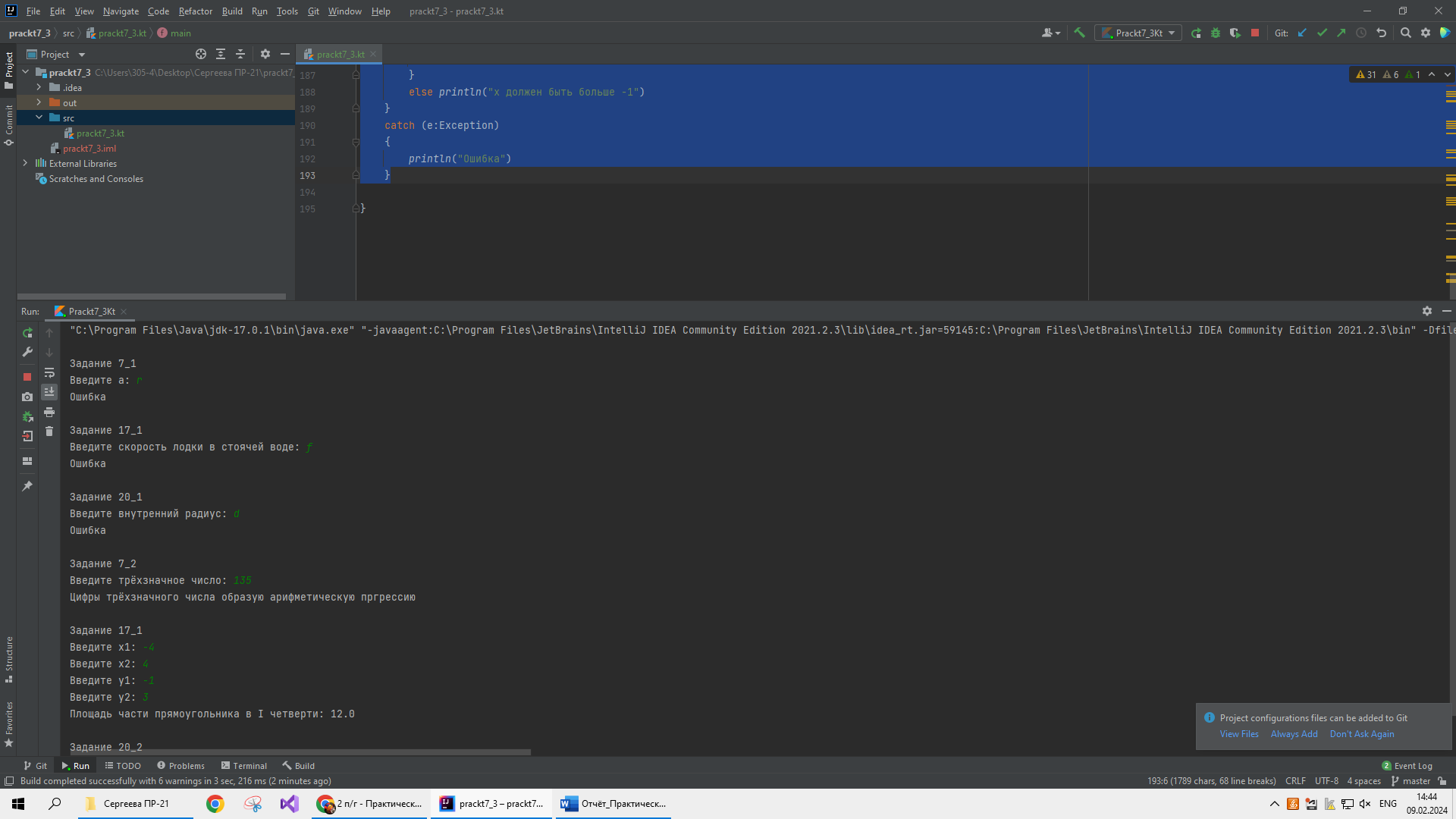
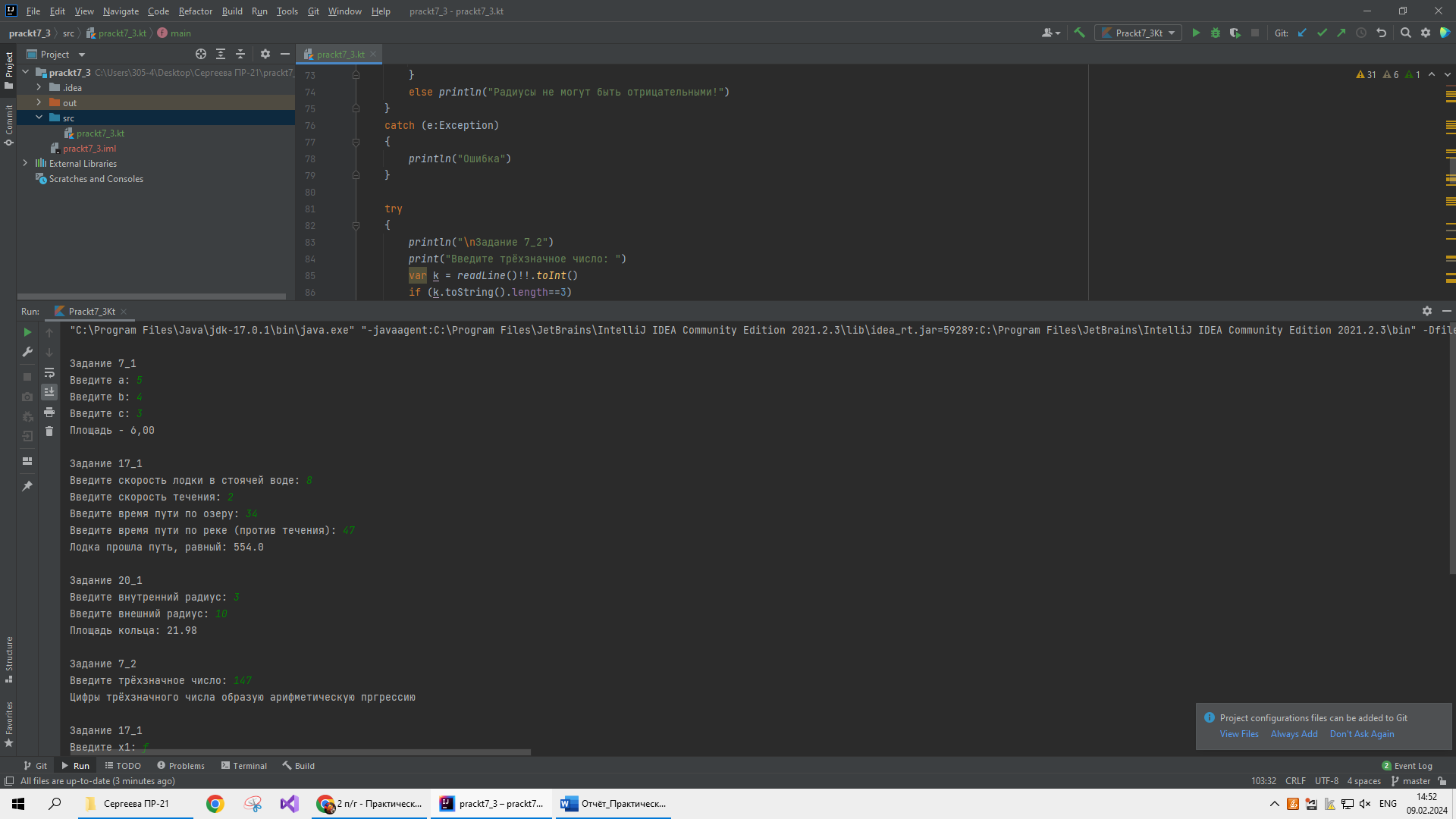
**Выходные данные:**

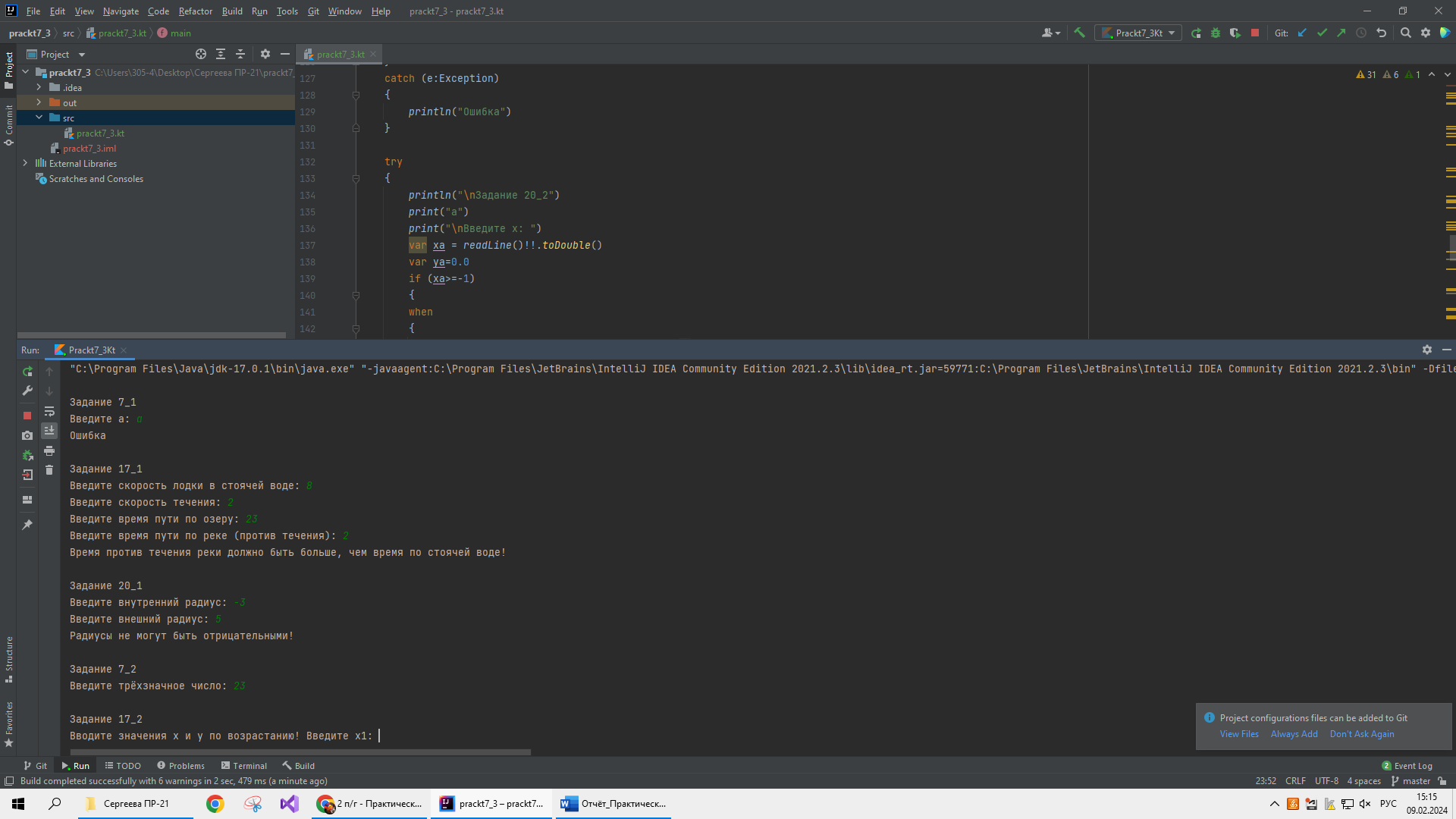
**Блок-схема:**

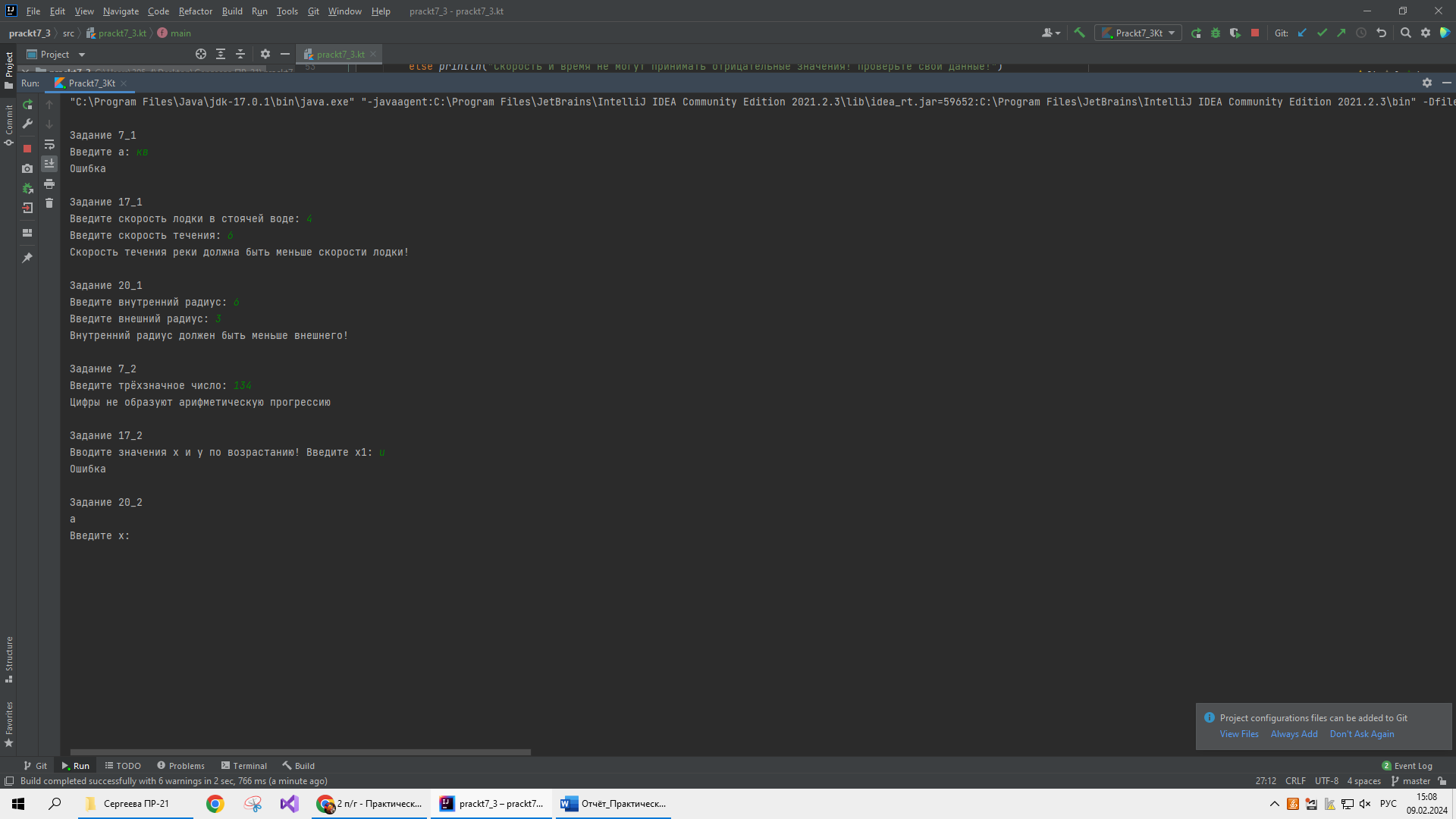
**Листинг программы:**

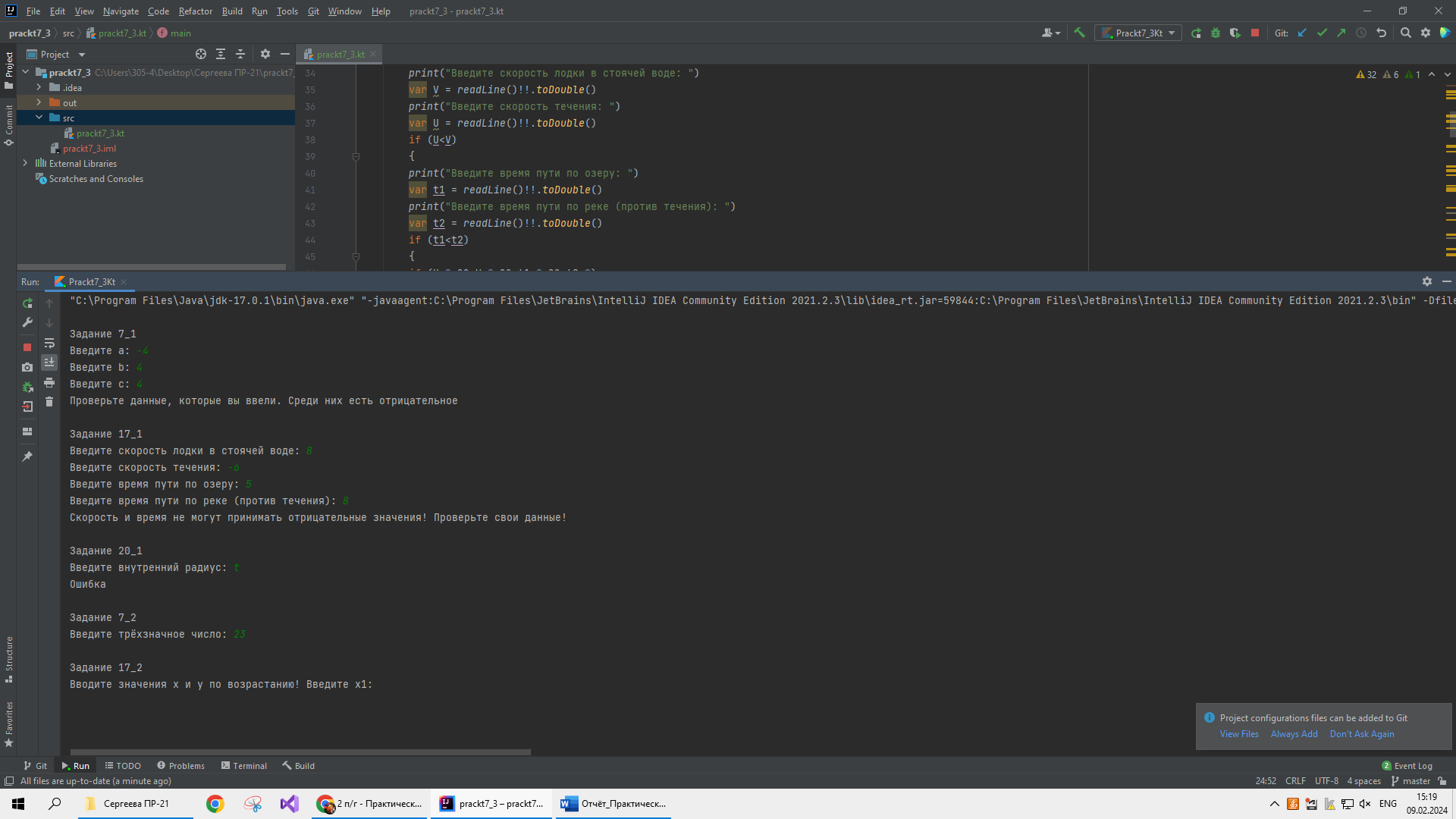
try  
{  
 *println*("\nЗадание 17\_1")  
 *print*("Введите скорость лодки в стоячей воде: ")  
 var V = *readLine*()!!.*toDouble*()  
 *print*("Введите скорость течения: ")  
 var U = *readLine*()!!.*toDouble*()  
 if (U<V)  
 {  
 *print*("Введите время пути по озеру: ")  
 var t1 = *readLine*()!!.*toDouble*()  
 *print*("Введите время пути по реке (против течения): ")  
 var t2 = *readLine*()!!.*toDouble*()  
 if (t1<t2)  
 {  
 if (U>0 && V>0 && t1>0 && t2>0)  
 {  
  
  
 var S = V \* t1 + (V - U) \* t2  
 *println*("Лодка прошла путь, равный: $S")  
 }  
 else *println*("Скорость и время не могут принимать отрицательные значения! Проверьте свои данные!")  
 }  
 else *println*("Время против течения реки должно быть больше, чем время по стоячей воде!")  
 }  
  
 else *println*("Скорость течения реки должна быть меньше скорости лодки!")  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**









**Задание 20(1):**

**Входные данные:**

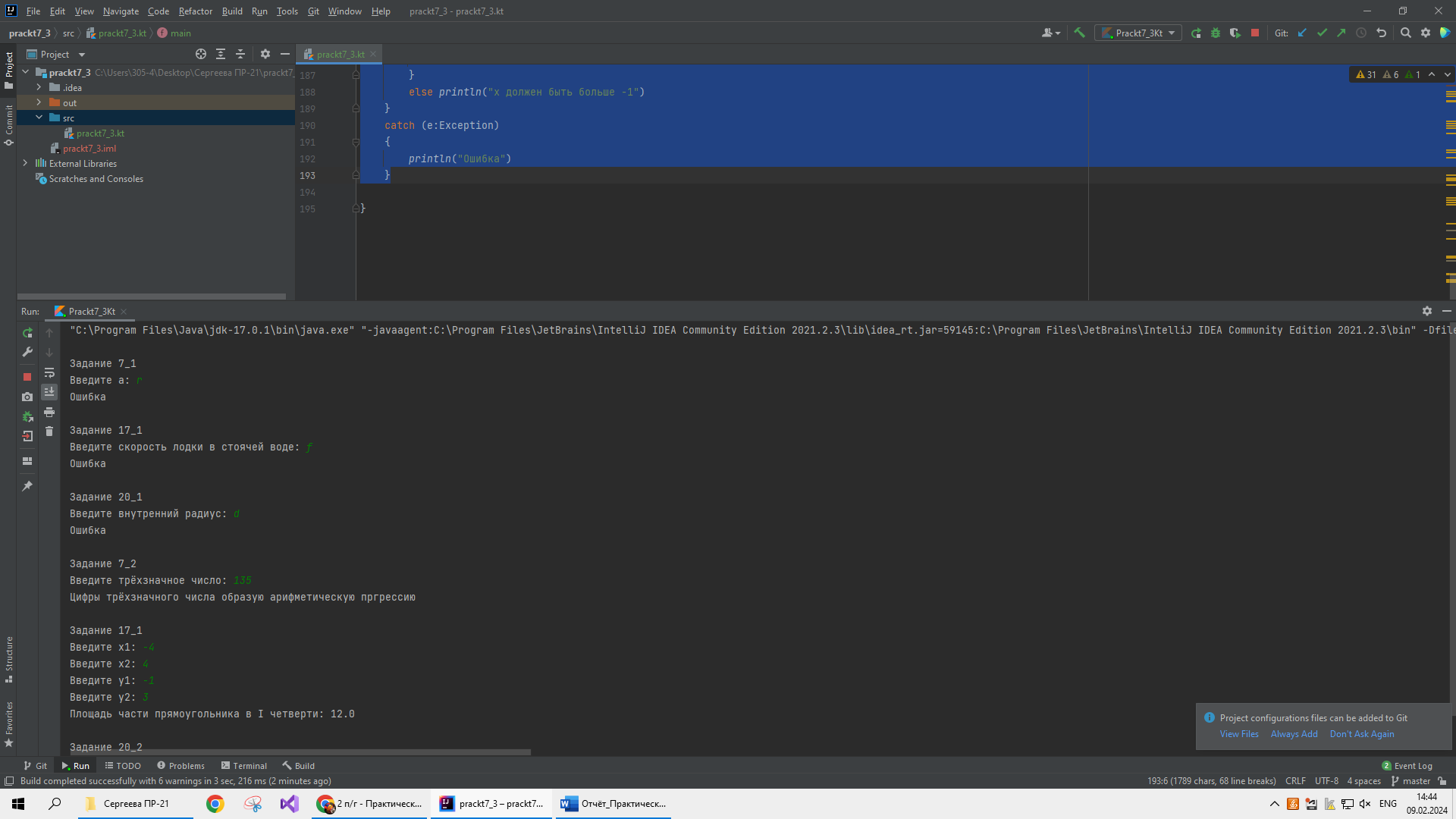
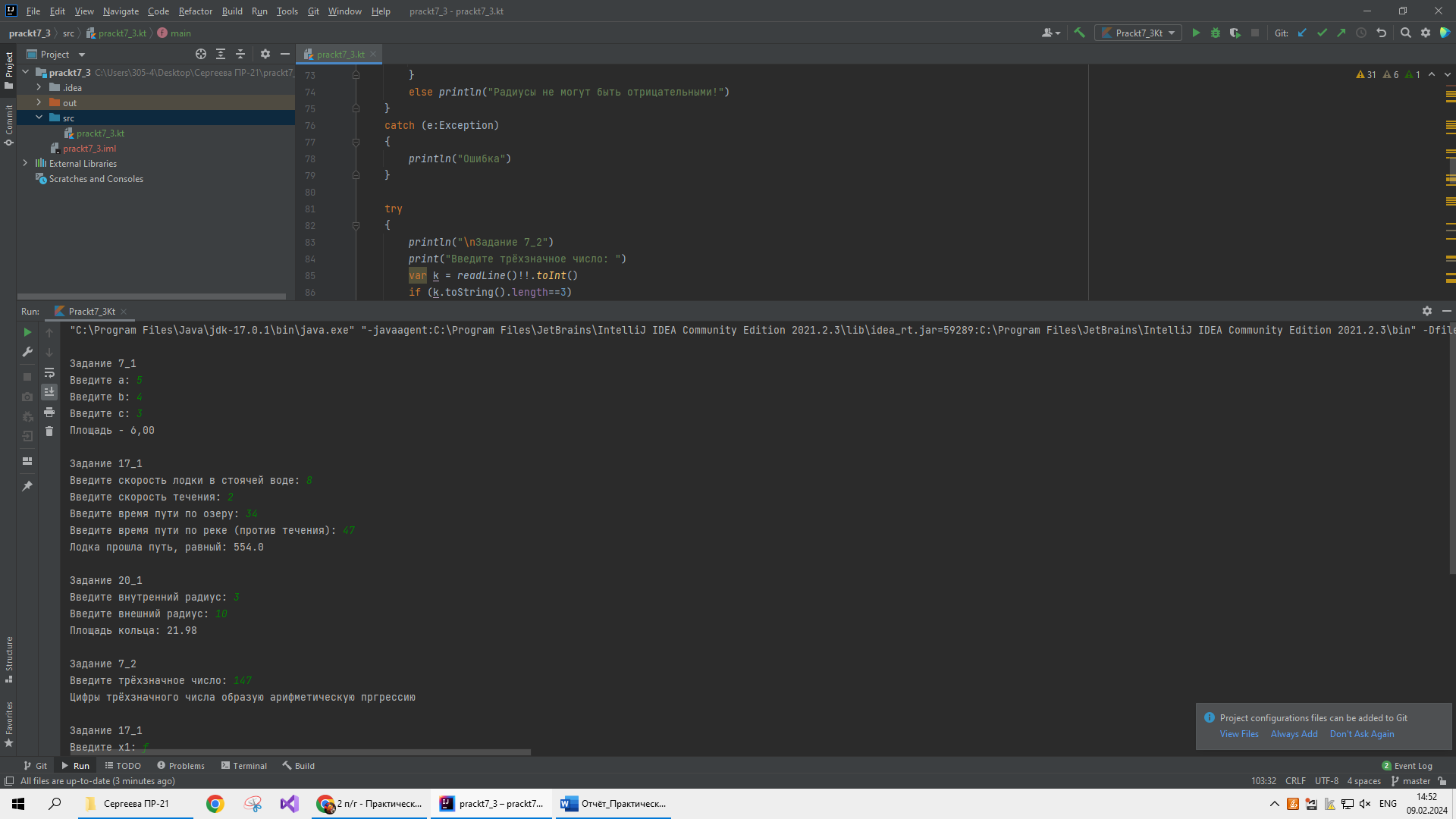
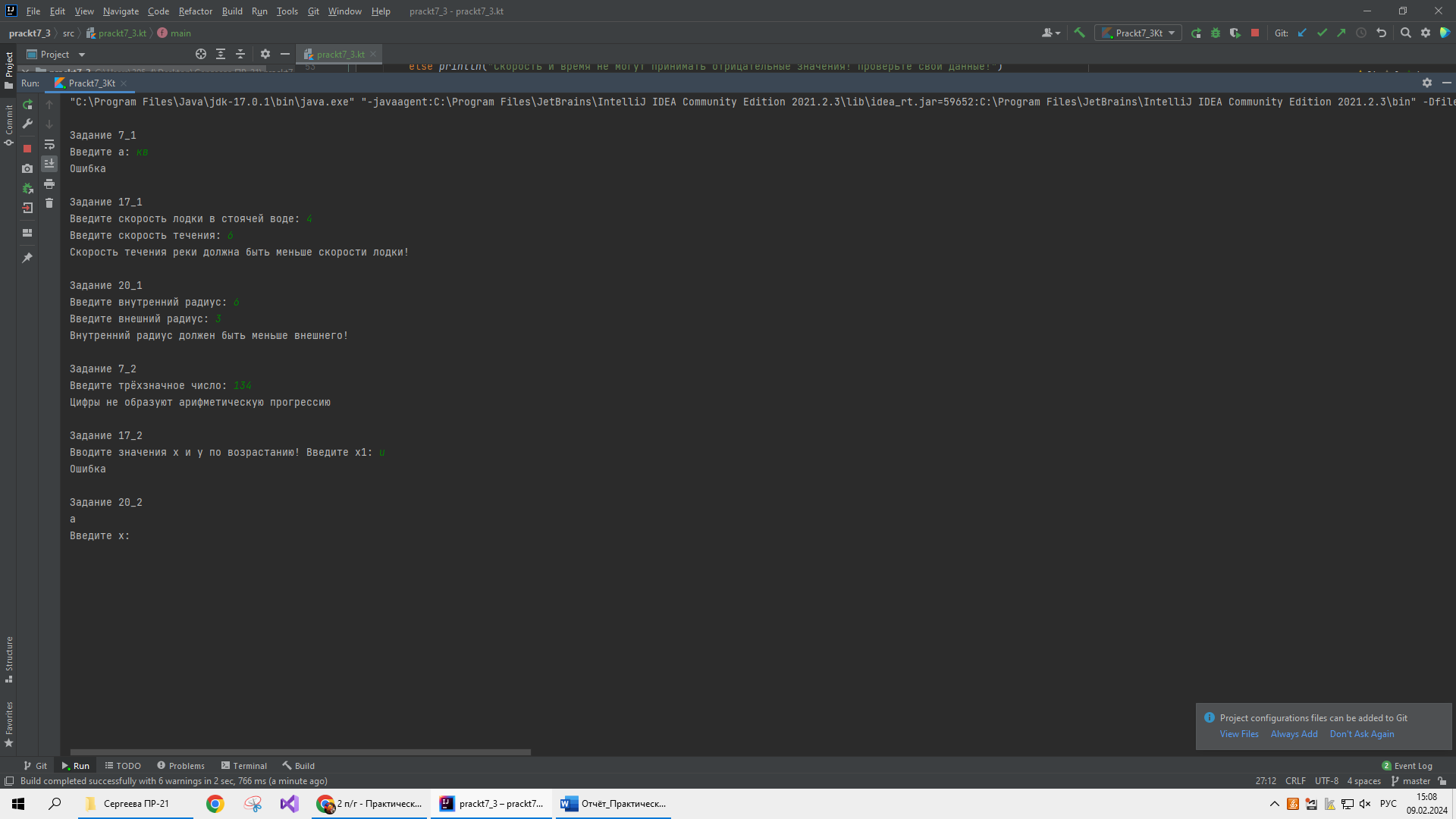
**Выходные данные:**

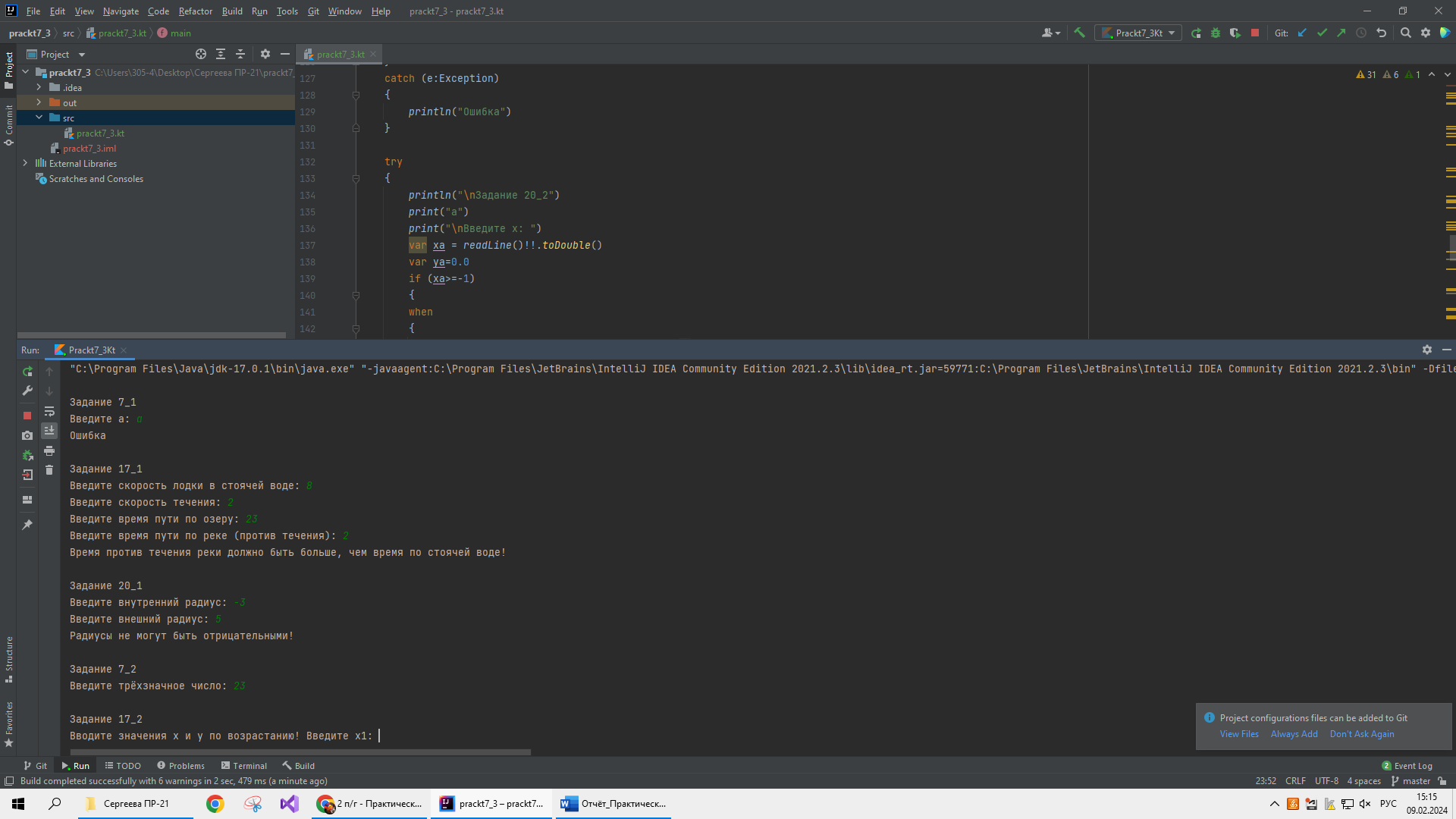
**Блок-схема:**

**Листинг программы:**

try  
{  
 *println*("\nЗадание 20\_1")  
 *print*("Введите внутренний радиус: ")  
 var r1 = *readLine*()!!.*toDouble*()  
 *print*("Введите внешний радиус: ")  
 var r2 = *readLine*()!!.*toDouble*()  
 if (r1>0 && r2>0)  
 {  
 if (r1<r2)  
 {  
 var s = 3.14 \* (r2 - r1)  
 *println*("Площадь кольца: $s")  
 }  
 else *println*("Внутренний радиус должен быть меньше внешнего!")  
 }  
 else *println*("Радиусы не могут быть отрицательными!")  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**





**Задание 7(2):**

**Входные данные:**

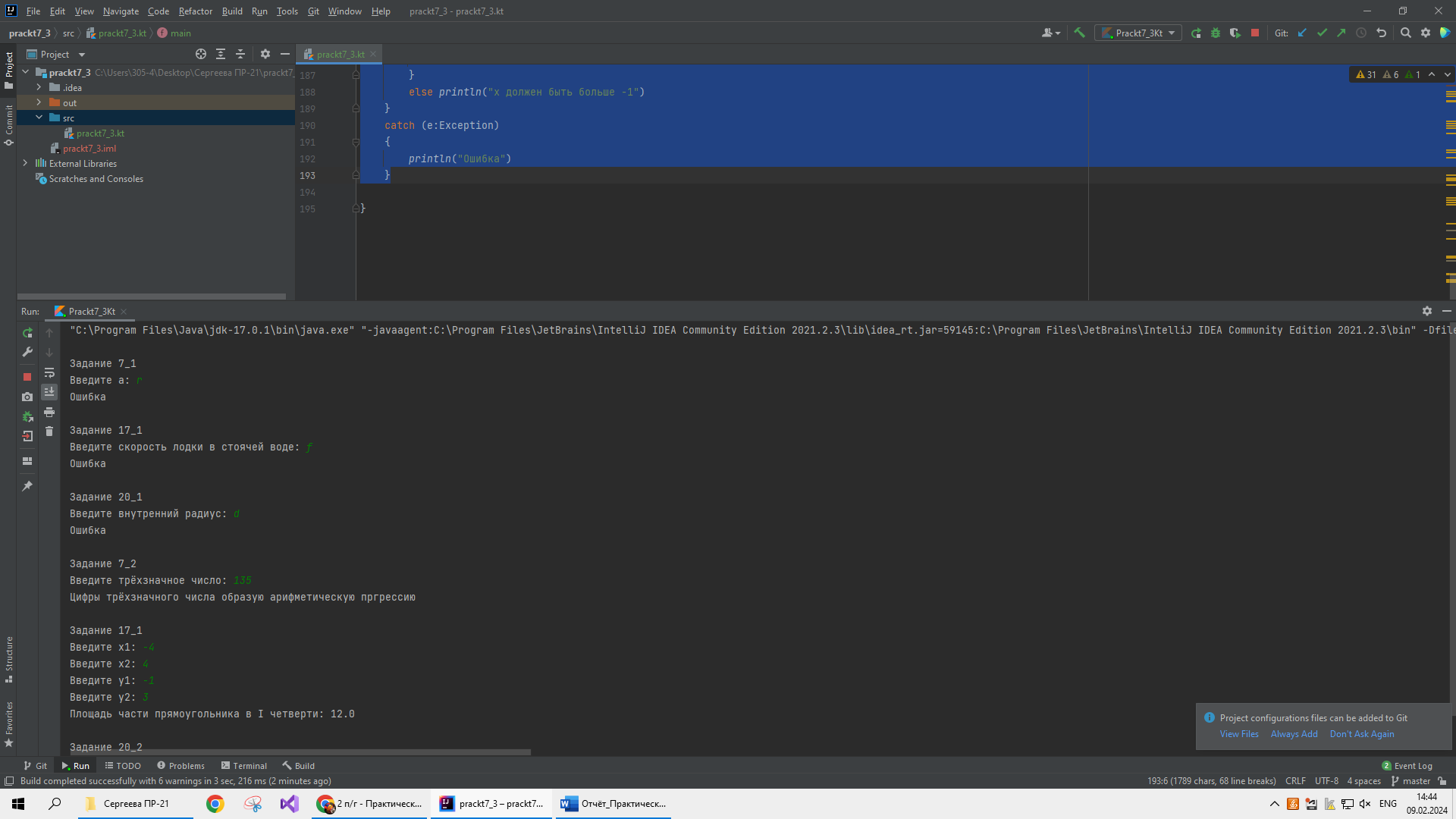
**Выходные данные:**

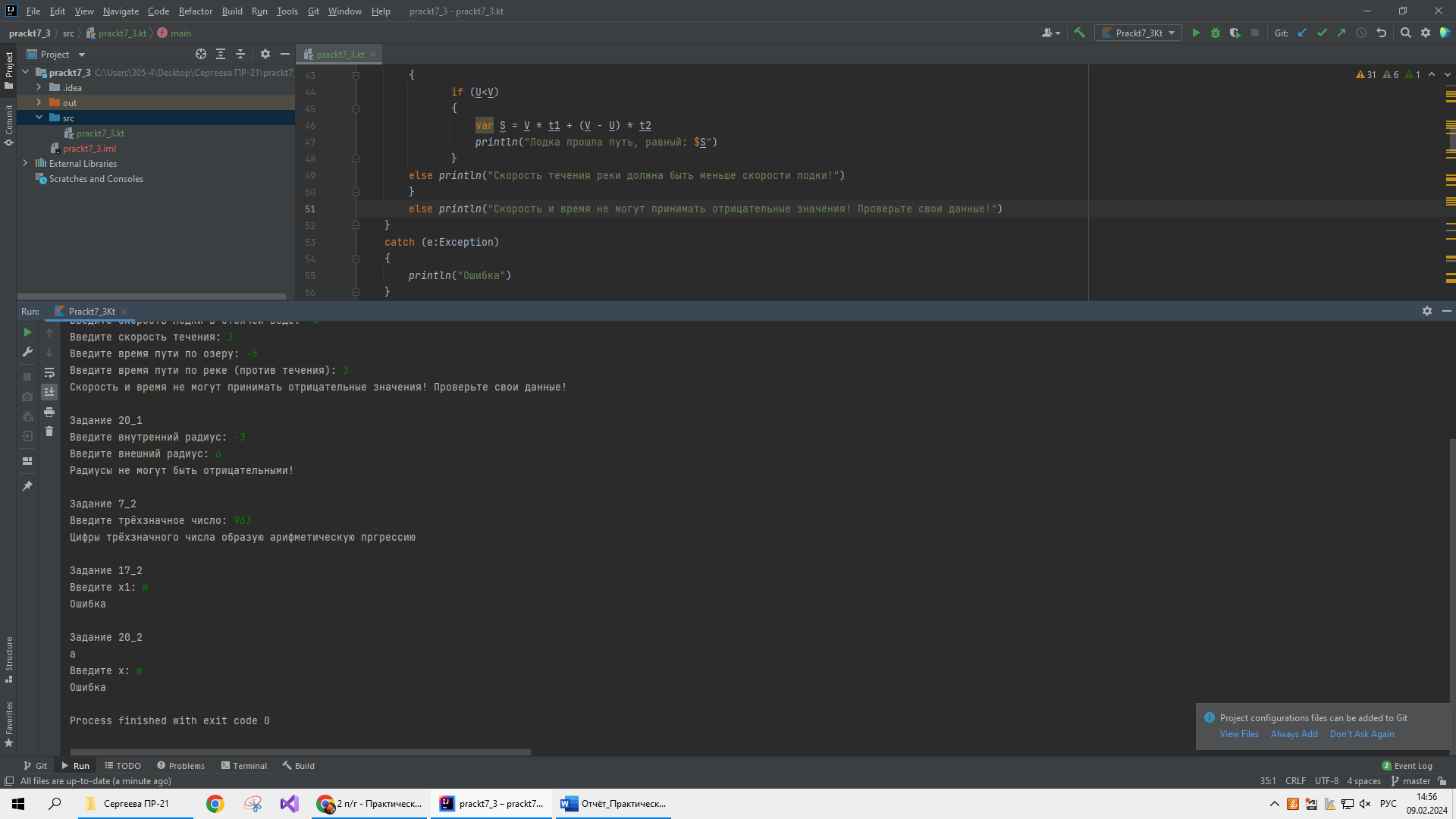
**Блок-схема:**

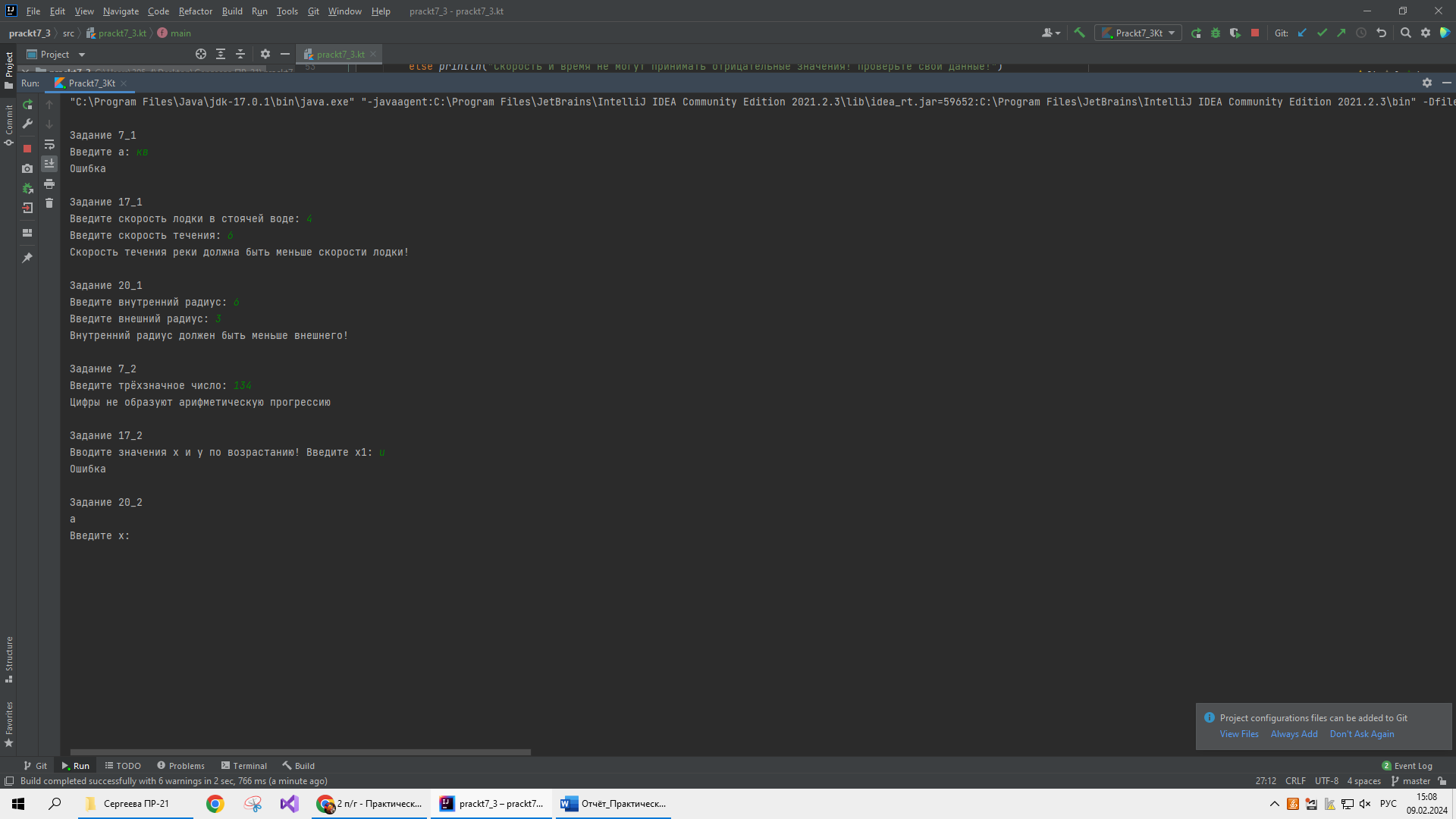
**Листинг программы:**

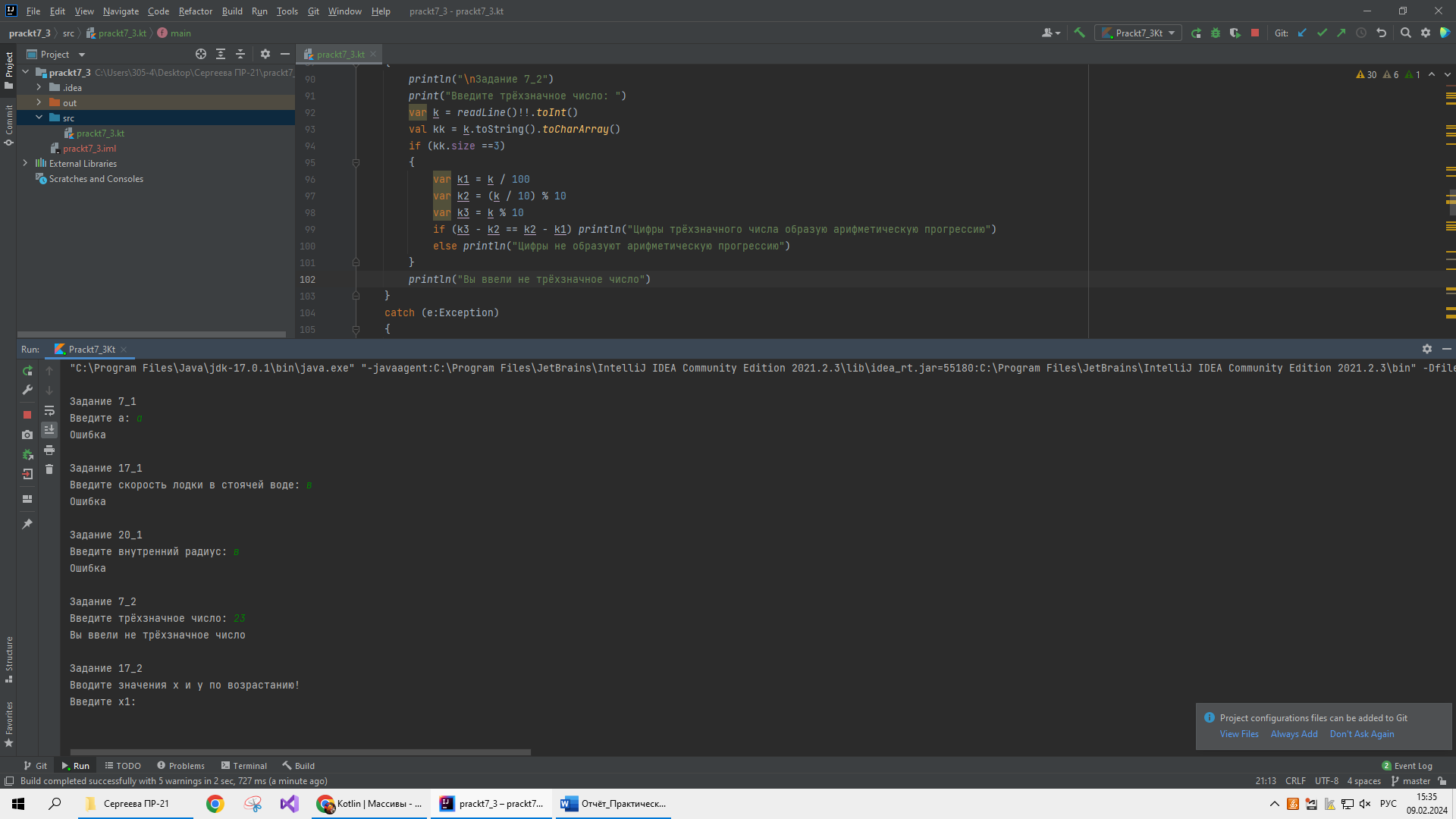
try  
{  
 *println*("\nЗадание 7\_2")  
 *print*("Введите трёхзначное число: ")  
 var k = *readLine*()!!.*toInt*()  
 val kk = k.toString().*toCharArray*()  
 if (kk.size ==3)  
 {  
 var k1 = k / 100  
 var k2 = (k / 10) % 10  
 var k3 = k % 10  
 if (k3 - k2 == k2 - k1) *println*("Цифры трёхзначного числа образую арифметическую прогрессию")  
 else *println*("Цифры не образуют арифметическую прогрессию")  
 }  
 *println*("Вы ввели не трёхзначное число")  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**









**Задание 17(2):**

**Входные данные:**

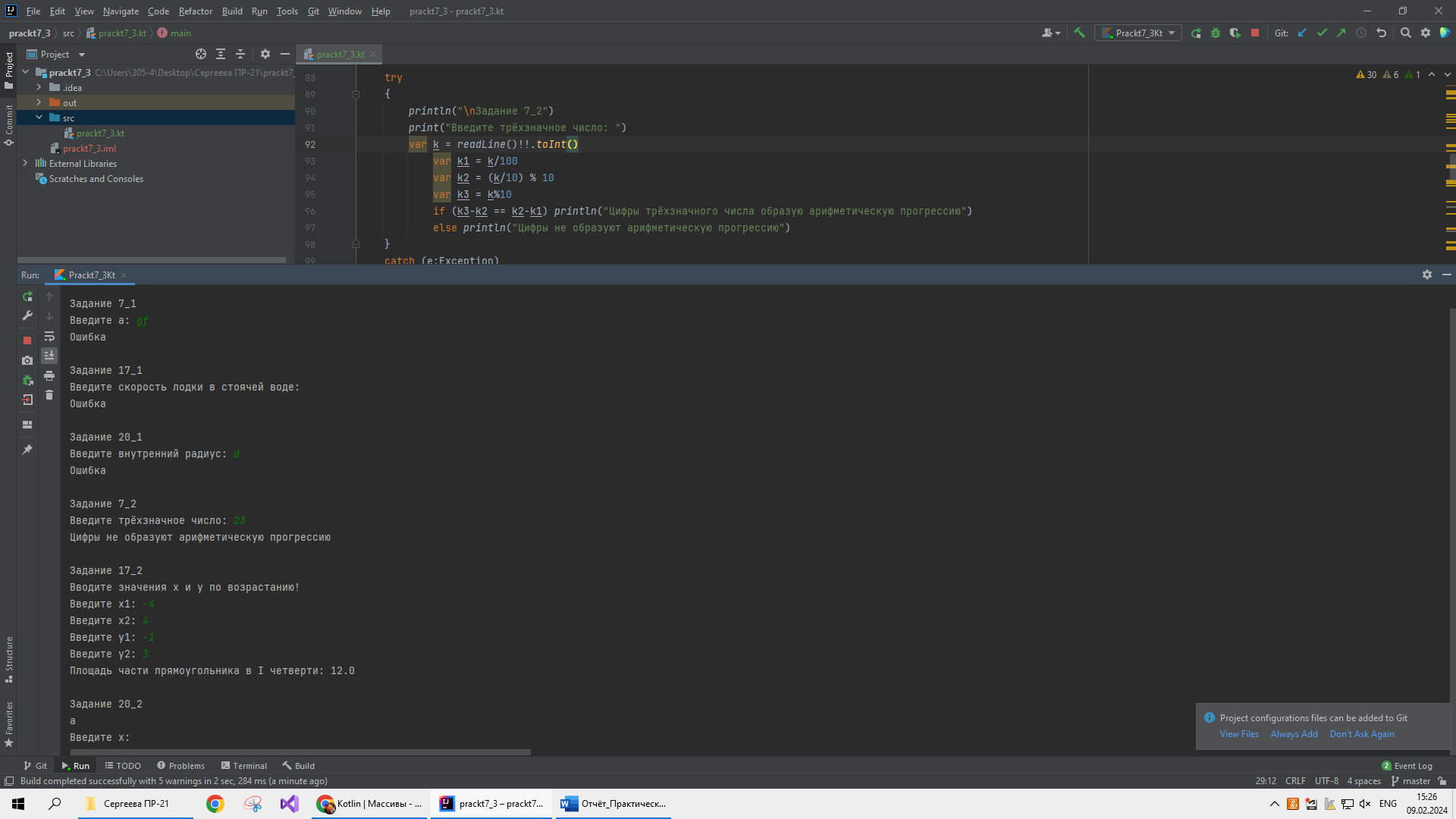
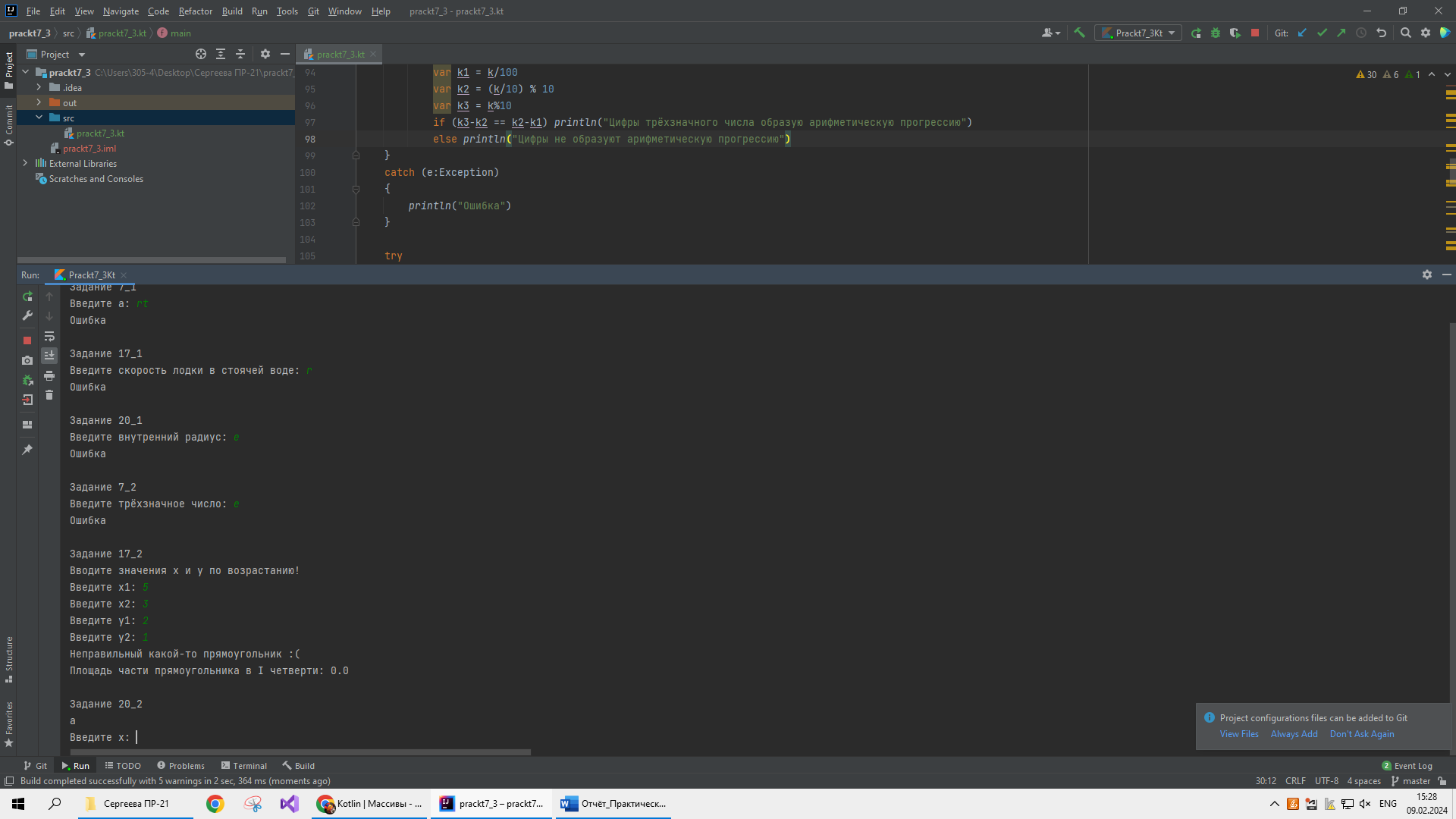
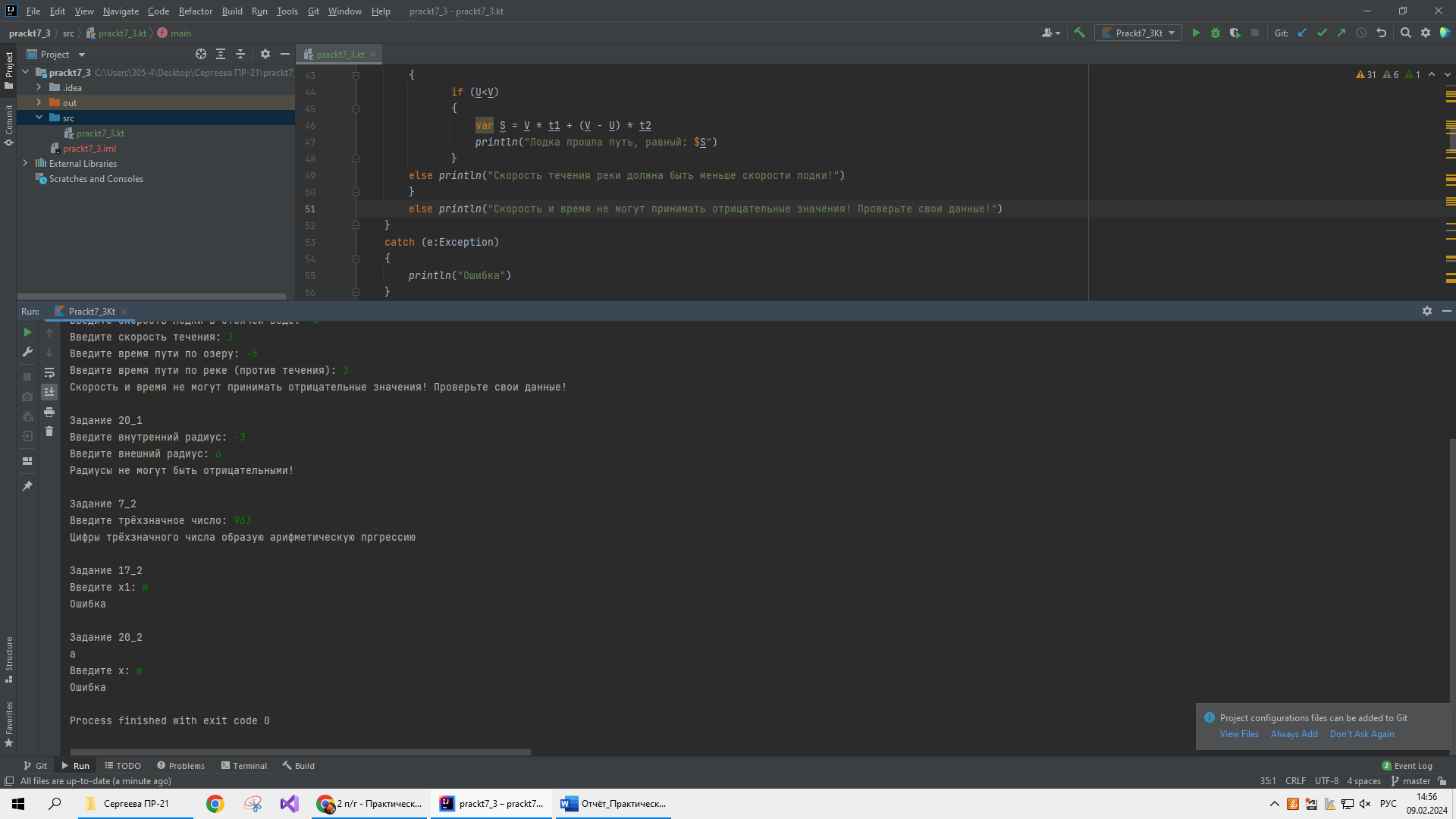
**Выходные данные:**

**Блок-схема:**

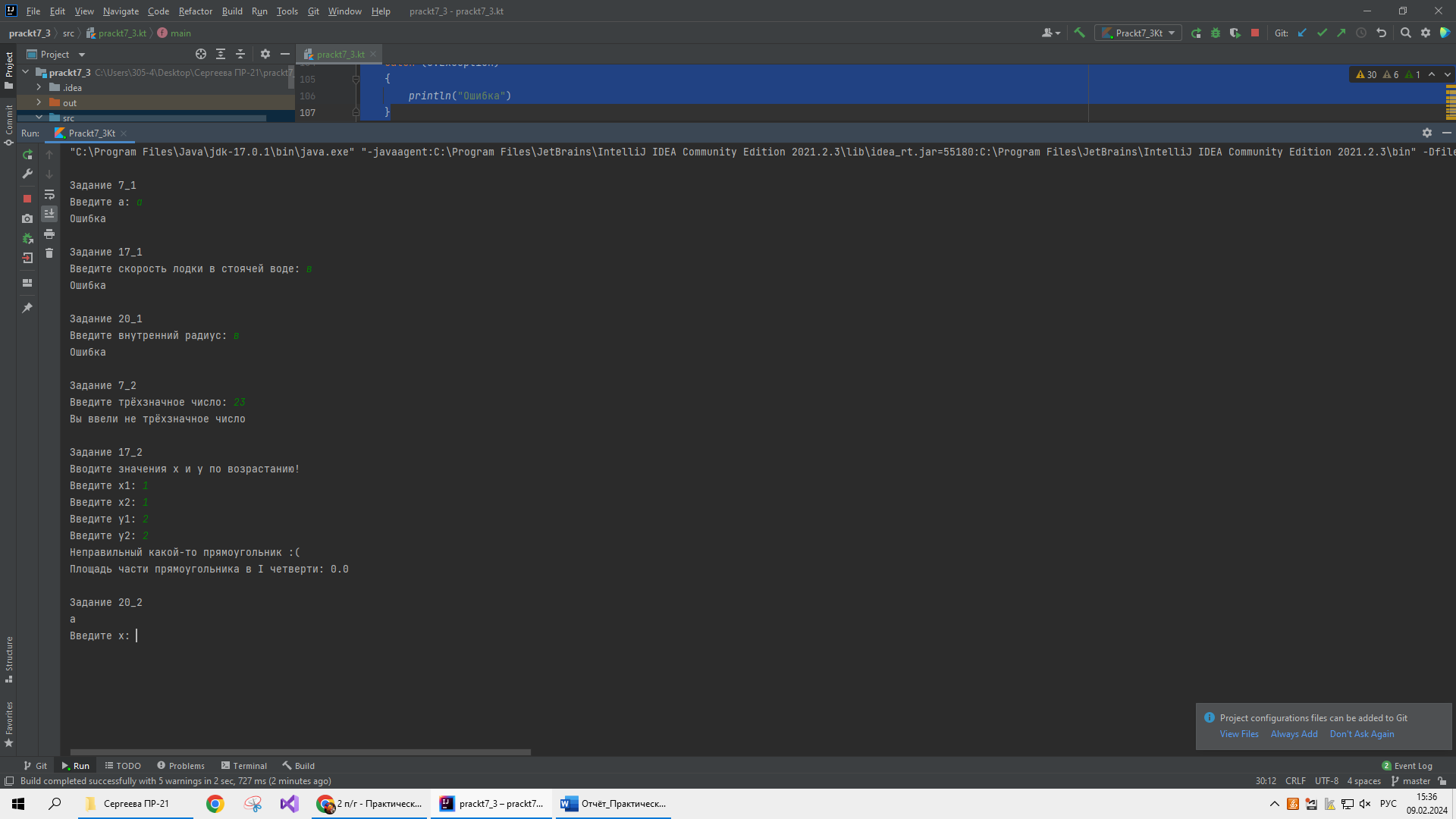
**Листинг программы:**

try  
{  
 *println*("\nЗадание 17\_2")  
 *print*("Вводите значения х и у по возрастанию! ")  
 *print*("Введите x1: ")  
 var x1 = *readLine*()!!.*toDouble*()  
 *print*("Введите x2: ")  
 var x2 = *readLine*()!!.*toDouble*()  
 *print*("Введите y1: ")  
 var y1 = *readLine*()!!.*toDouble*()  
 *print*("Введите y2: ")  
 var y2 = *readLine*()!!.*toDouble*()  
 var st = 0.0  
 if (x2>x1 && y2>y1) st = ((x2) \* (y2))  
 else *println*("Неправильный какой-то прямоугольник :(")  
 *println* ("Площадь части прямоугольника в I четверти: $st")  
  
  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**

**Задание 20(2):**



**Входные данные:**

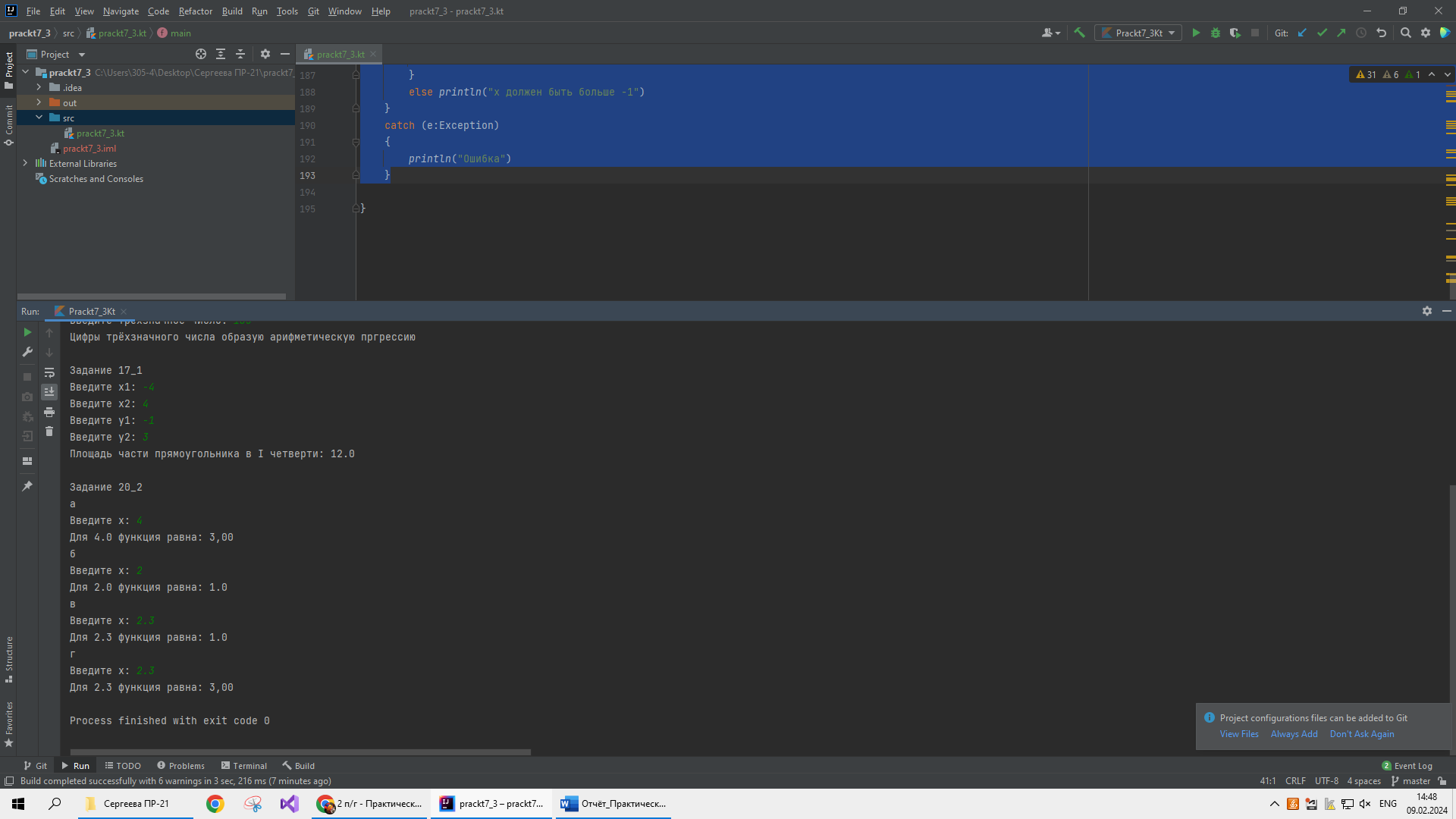
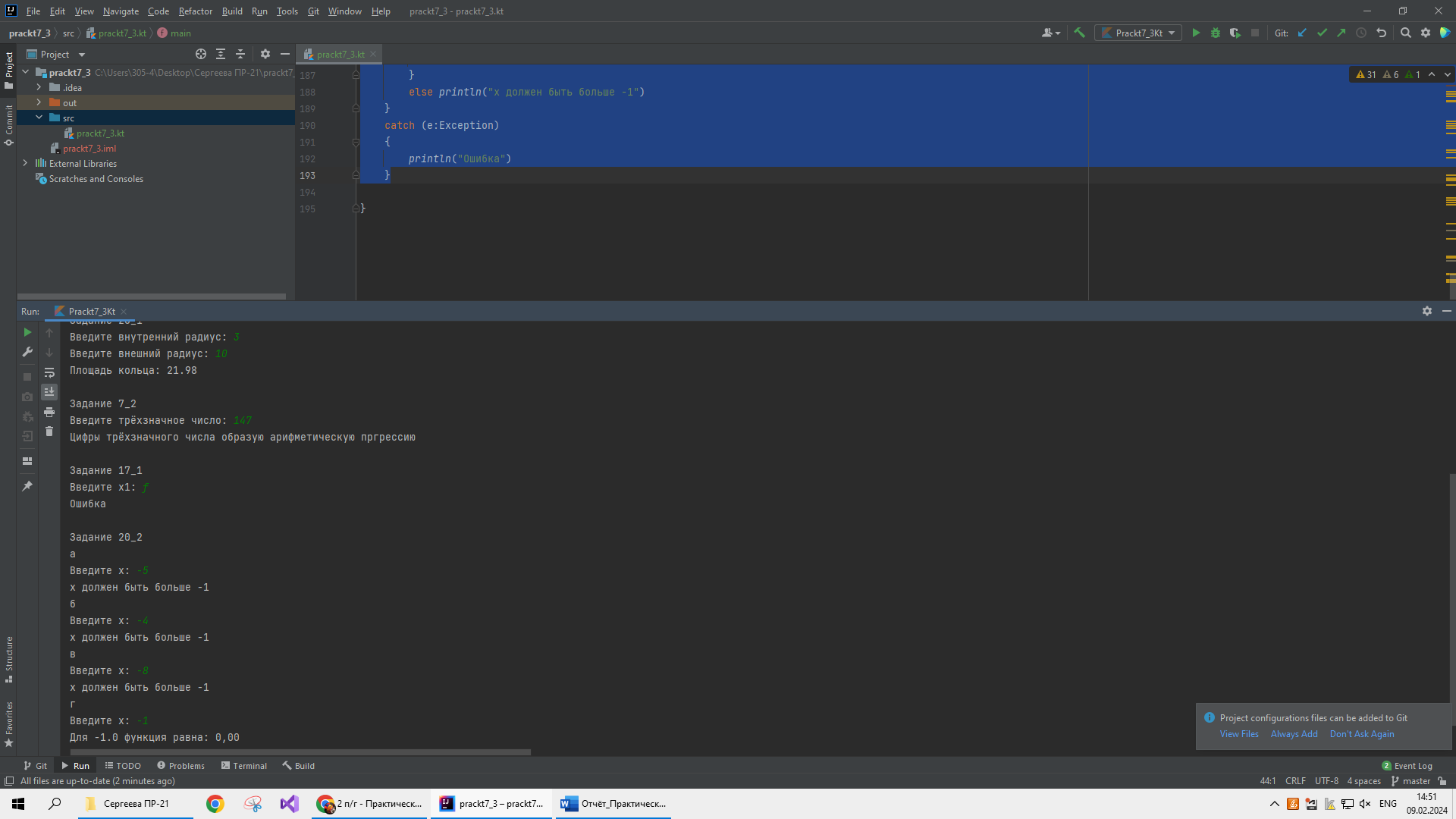
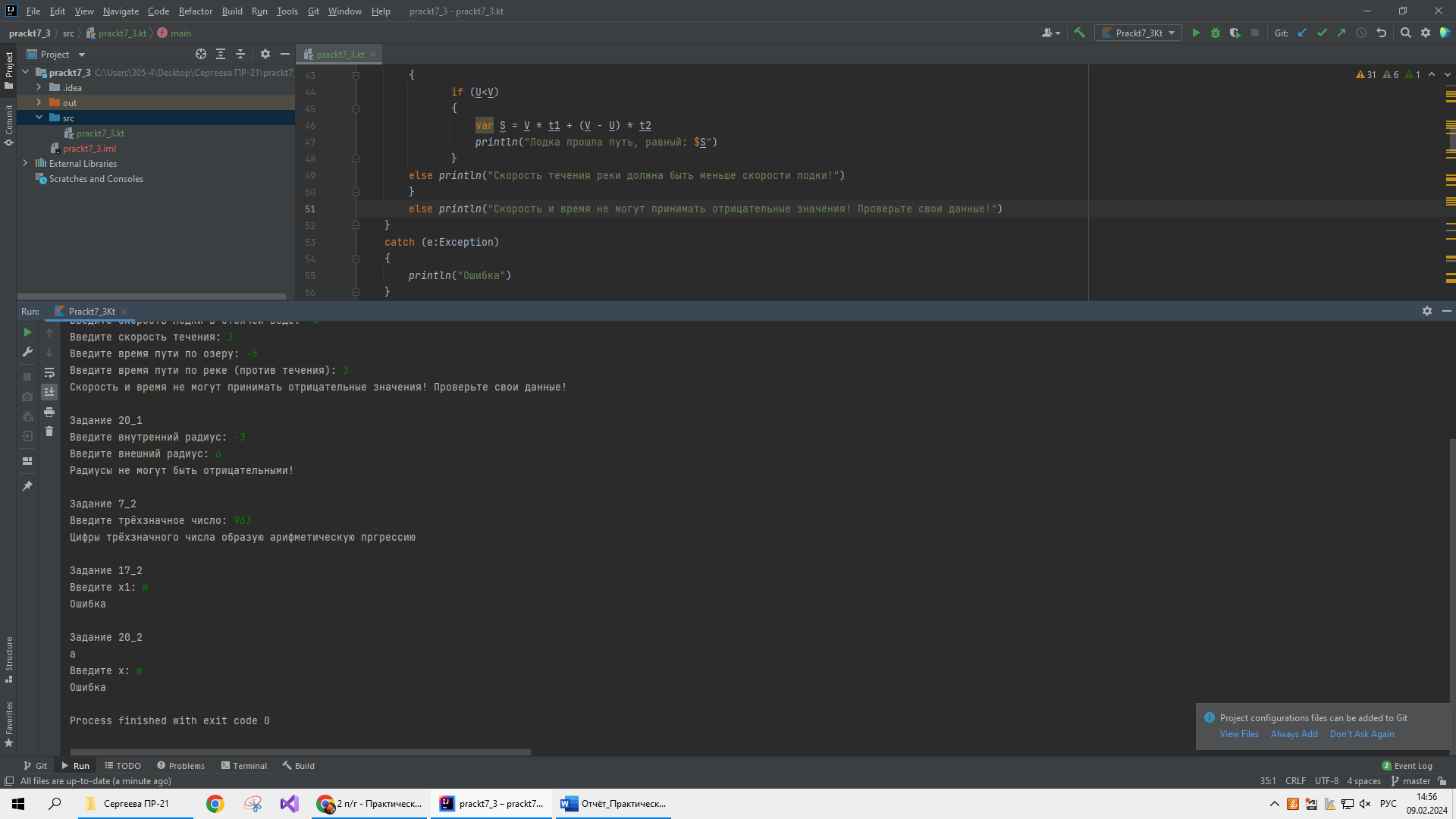
**Выходные данные:**

**Блок-схема:**

**Листинг программы:**

try  
{  
 *println*("\nЗадание 20\_2")  
 *print*("a")  
 *print*("\nВведите x: ")  
 var xa = *readLine*()!!.*toDouble*()  
 var ya=0.0  
 if (xa>=-1)  
 {  
 when  
 {  
 xa<=1 &&xa>=-1 -> ya = Math.sqrt(xa.*pow*(2) + 1)  
 xa>1 -> ya=xa-1  
 }  
 *println*("Для $xa функция равна: ${String.*format*("%.2f",ya)}")  
 }  
 else *println*("х должен быть больше -1")  
 *print*("б")  
 *print*("\nВведите x: ")  
 var xb = *readLine*()!!.*toDouble*()  
 var yb=2.0  
 if (xb>=-1)  
 {  
 when  
 {  
 xb<=0 &&xb>=-1 -> yb = xb+1  
 xb>0&&xb<=3 -> yb=1.0  
 xb>3&&xb<=4 -> yb--  
 else -> *println* ("х не входит в границы")  
 }  
 *println*("Для $xb функция равна: $yb")  
 }  
 else *println*("х должен быть больше -1")  
 *print*("в")  
 *print*("\nВведите x: ")  
 var xc = *readLine*()!!.*toDouble*()  
 var yc=0.0  
 if (xc>=-1)  
 {  
 when  
 {  
 xc<=0 &&xc>=-1 -> yc = xc+1  
 xc>0&&xc<=1 -> yc=xc  
 xc>3&&xc<=4 -> yc=xc\*0.5-0.5  
 else -> yc = 1.0  
 }  
 *println*("Для $xc функция равна: $yc")  
 }  
 else *println*("х должен быть больше -1")  
 *print*("г")  
 *print*("\nВведите x: ")  
 var xd = *readLine*()!!.*toDouble*()  
 var yd=0.0  
 if (xd>=-1)  
 {  
 when  
 {  
 xd<=0 &&xd>=-1 -> yd = Math.sqrt(xa.*pow*(2) + 1)  
 xd>1 && xd<=2-> yd=1.0  
 xd>2 -> yd = xd-1.0  
 }  
 *println*("Для $xd функция равна: ${String.*format*("%.2f",ya)}")  
 }  
 else *println*("х должен быть больше -1")  
}  
catch (e:Exception)  
{  
 *println*("Ошибка")  
}

**Тестовые ситуации:**



**Вывод:** научились работать с классом Math в Kotlin