Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГТУ»)**

**Факультет информационных технологий и кибербезопасности**

**Кафедра информационных систем и программирования**

**ЛАБОРАТОРНАЯ РАБОТА №1**

**Дисциплина: Тестирование и отладка программного обеспечения**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Д. Юданов

Направление подготовки: 09.03.04 Программная инженерия

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Г. Волик

**Наименование:** Отладка программ в интегрированных средах разработки.

**Цель работы:** изучить инструментальные средства и возможности отладки программ в интегрированной среде Microsoft Visual Studio или JetBrains Rider.

**Краткая теория:** Интегрированные интерактивные среды разработки программ (IDE) включают в себя ряд средств, облегчающих процесс нахождения ошибок в программе, которые не позволяют ей корректно работать. Как Microsoft Visual Studio, так и JetBrains Rider имеют в своем составе схожий набор средств отладки.

**Задание:**

В отчете должен присутствовать текст программы с номерами строк. Последовательность выполняемых строк записывается в столбец, с указанием значений изменяющихся переменных через запятую. При выполнении программы до точки остановки, в последовательности выполнения указать строку с точкой остановки и значения переменных в данной точке.

**5.1 Часть первая**

1) Набрать программу примера 1 (раздел 2.4.1)

2) Выполнить программу по шагам, фиксируя в отчете, в строке с каким номером находится строка выполнения при каждом нажатии на F10.

3) Внести в программу изменения в соответствии с примером 2.

4) Выполнить задание пункта 2.

5) Выполнить трассировку исходной программы примера 3 (раздел 2.4.2), фиксируя в отчете, в строке с каким номером находится строка выполнения при каждом нажатии на F11.

**5.2 Часть вторая**

1) Набрать и откомпилировать следующую программу.

2) После каждой строки программы проставить номер. Например, //1, //2 и т.д.

3) Выполнить трассировку программы (без захода в функции стандартных библиотек), наблюдая за переменными в окне Locals.

4) Остановить отладку программы.

5) Установить точку остановки на операторе if (a[i] > z).

6) Выполнить программу до курсора на строке s = sum(a, N);

7) Продолжить выполнить программы до точки остановки. Далее продолжать пошаговое выполнение до строки Console.WriteLine(k).

8) Добавить в окно Watch 1 переменные x, y, s для наблюдения изменения их значений. Продолжать пошаговое выполнение.

9) Остановить отладку программы.

10) В операторе цикла while задать условную точку останова по числу проходов. Запустить программу для отладки.

11) Продолжать пошаговое выполнение до конца программы, наблюдая изменение значений x, y, s в окне Watch 1.

12) Записать полученные результаты.

13) Остановить отладку программы.

14) В операторе цикла while задать еще одну условную точку останова по логическому условию. Запустить программу для отладки.

15) Продолжать пошаговое выполнение до конца программы, наблюдая изменение значений x, y, s в окне Watch 1.

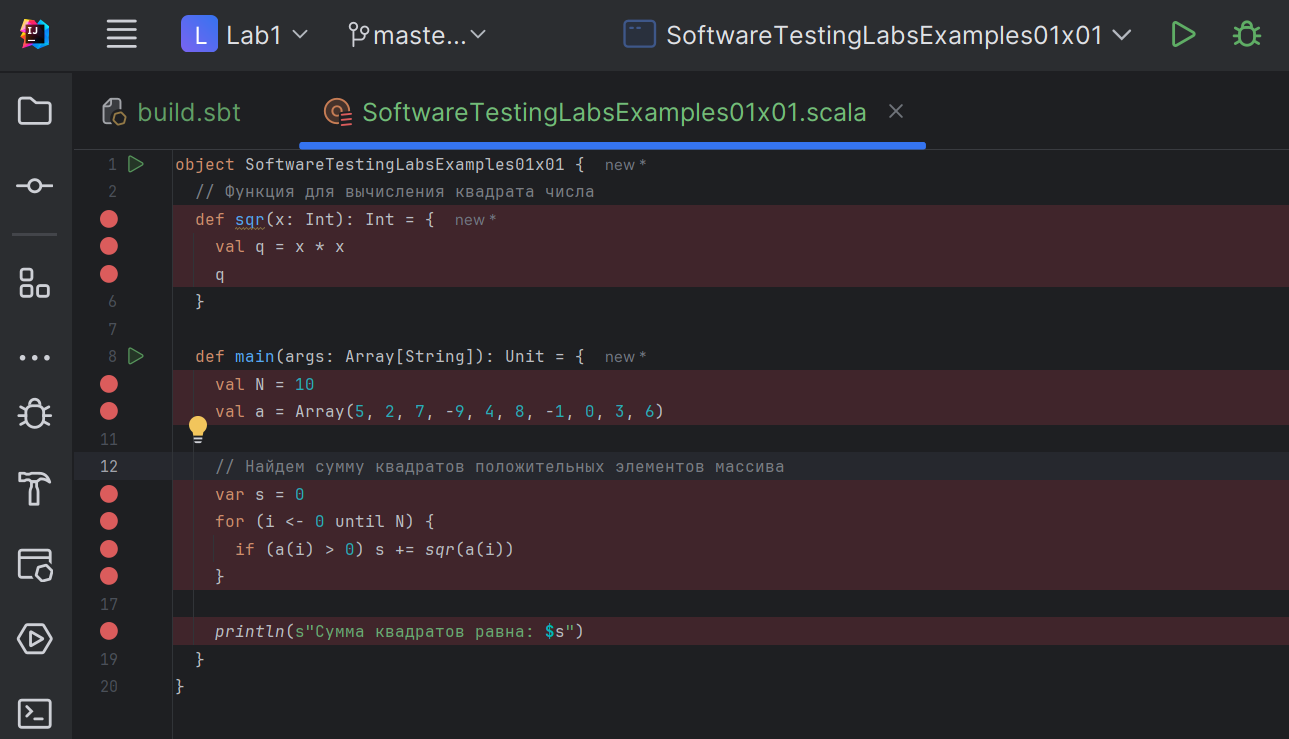
16) Записать полученные результаты.

**Ход работы:**

**Часть первая**

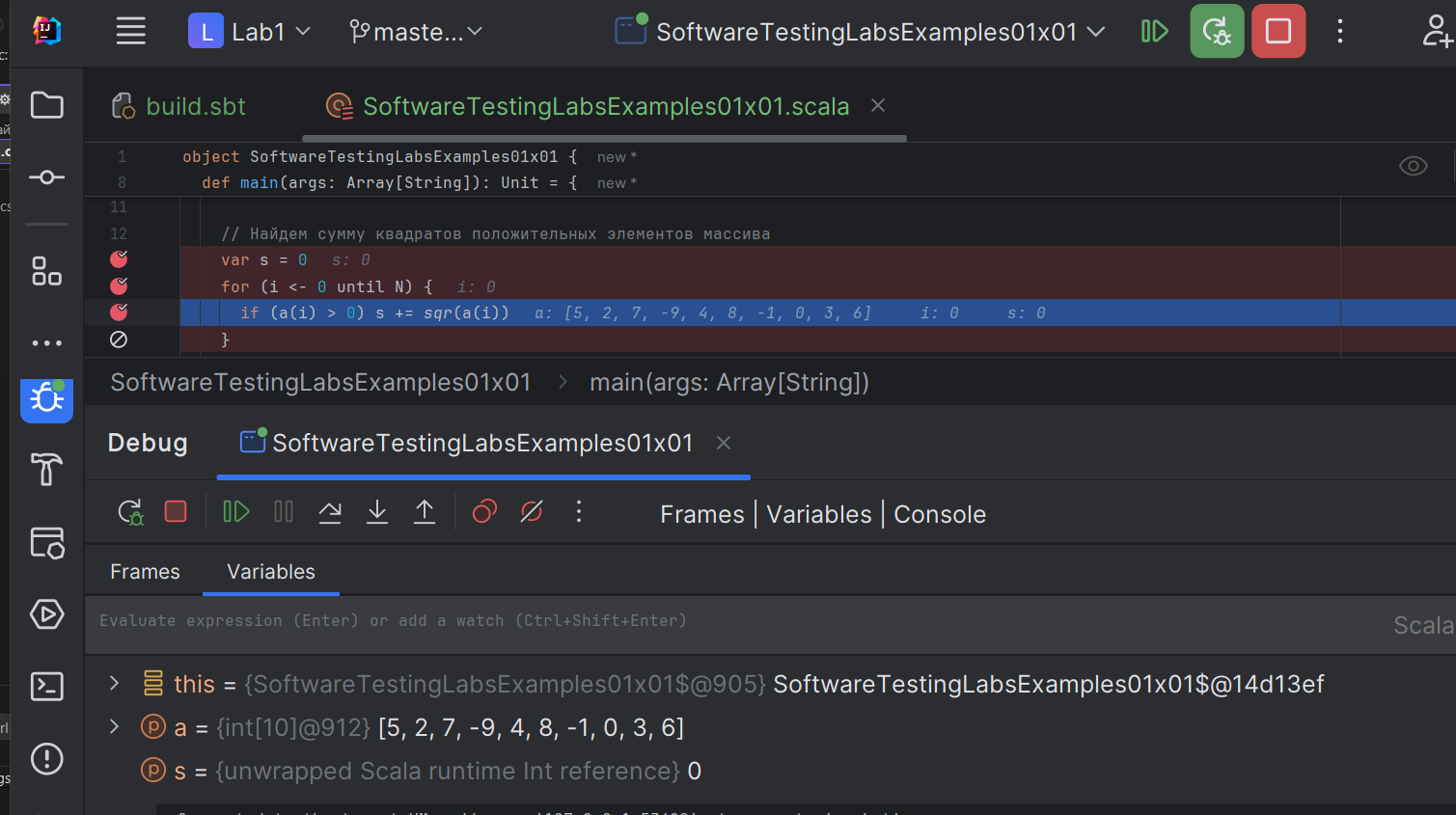
**2) Последовательность выполнения (F10) примера 1:**

0, 2, 5, 6, 6, 7, 7, 6, 6, 7, 7, 6, 6, 7, 7, 6, 6, 7, 6, 6, 7, 7, 6, 6, 7, 7, 6, 6, 7, 6, 6, 7, 6, 6, 7, 7, 6, 6, 7, 7, 6, 6, 8, 9



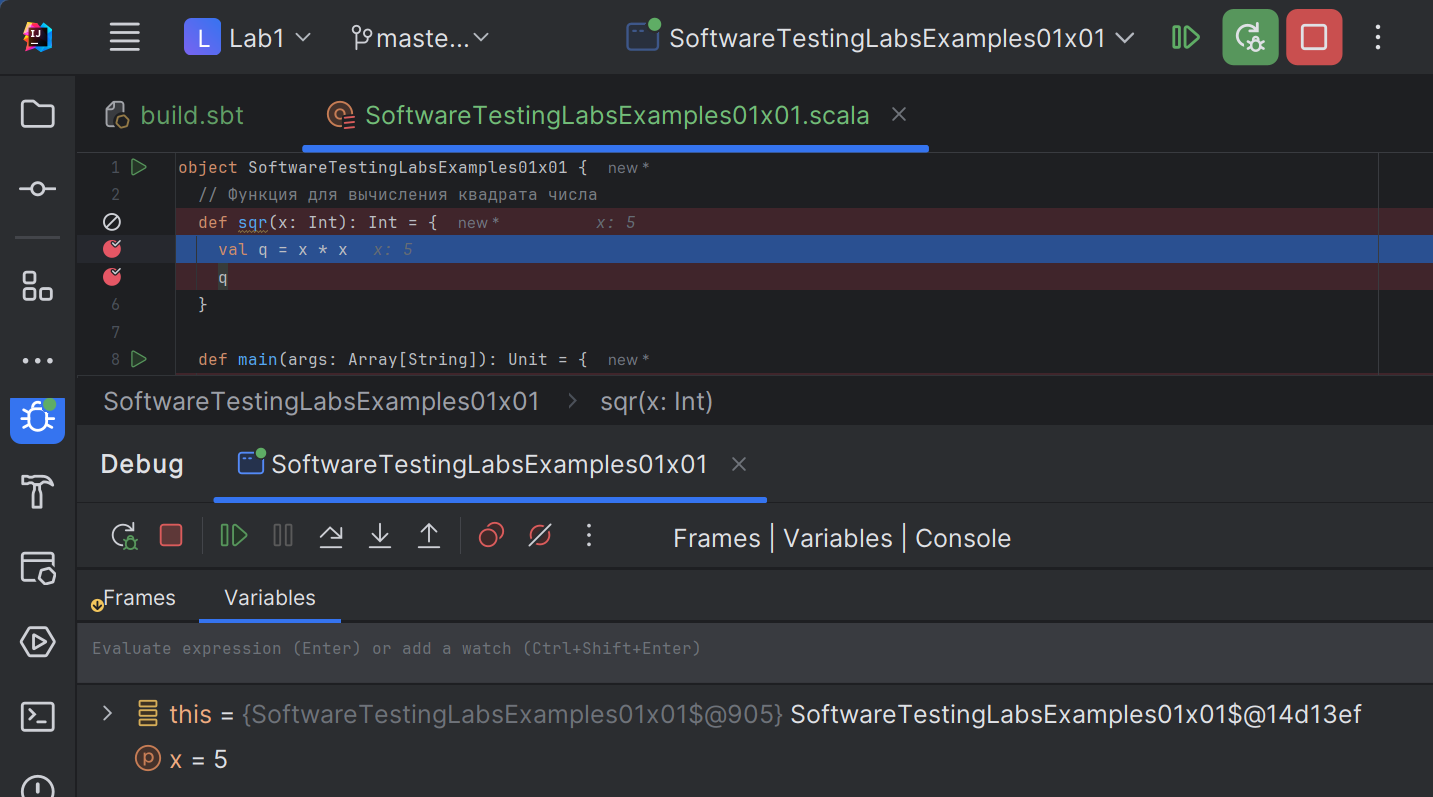
**4) Последовательность выполнения (F10) примера 2:**

0, 2, 5, 6, 6, 7.0, 7.1, 6, 6, 7.0, 7.1, 6, 6, 7.0, 7.1, 6, 6, 7.0, 6, 6, 7.0, 7.1, 6, 6, 7.0, 7.1, 6, 6, 7.0, 6, 6, 7.0, 6, 6, 7.0, 7.1, 6, 6, 7.0, 7.1, 6, 6, 8, 9



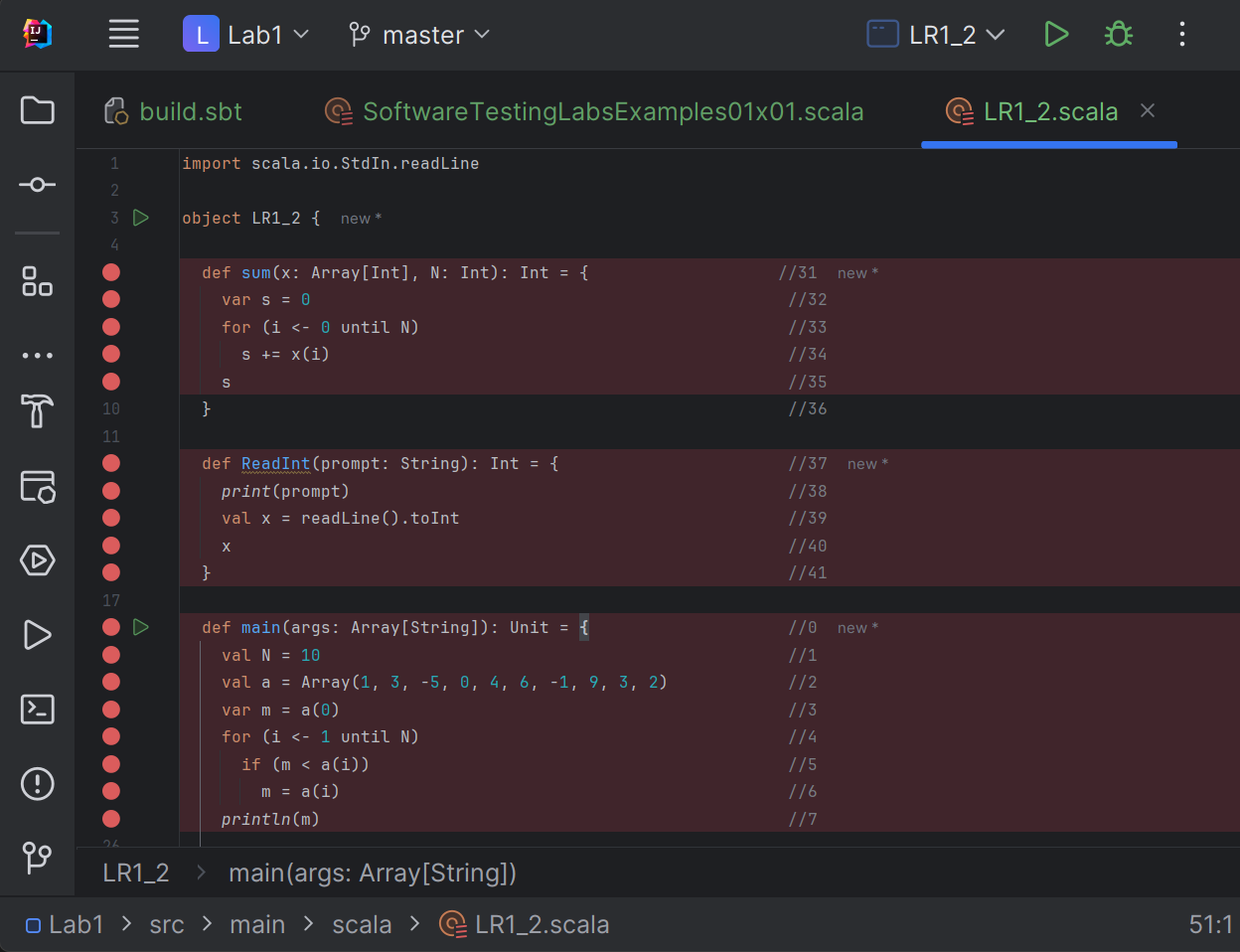
**5) Последовательность выполнения (F11) примера 2:**

0, 2, 5, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 6, 6, 7.0, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 7.0, 7.1, 10, 11, 12, 13, 7.1, 6, 6, 8, 9

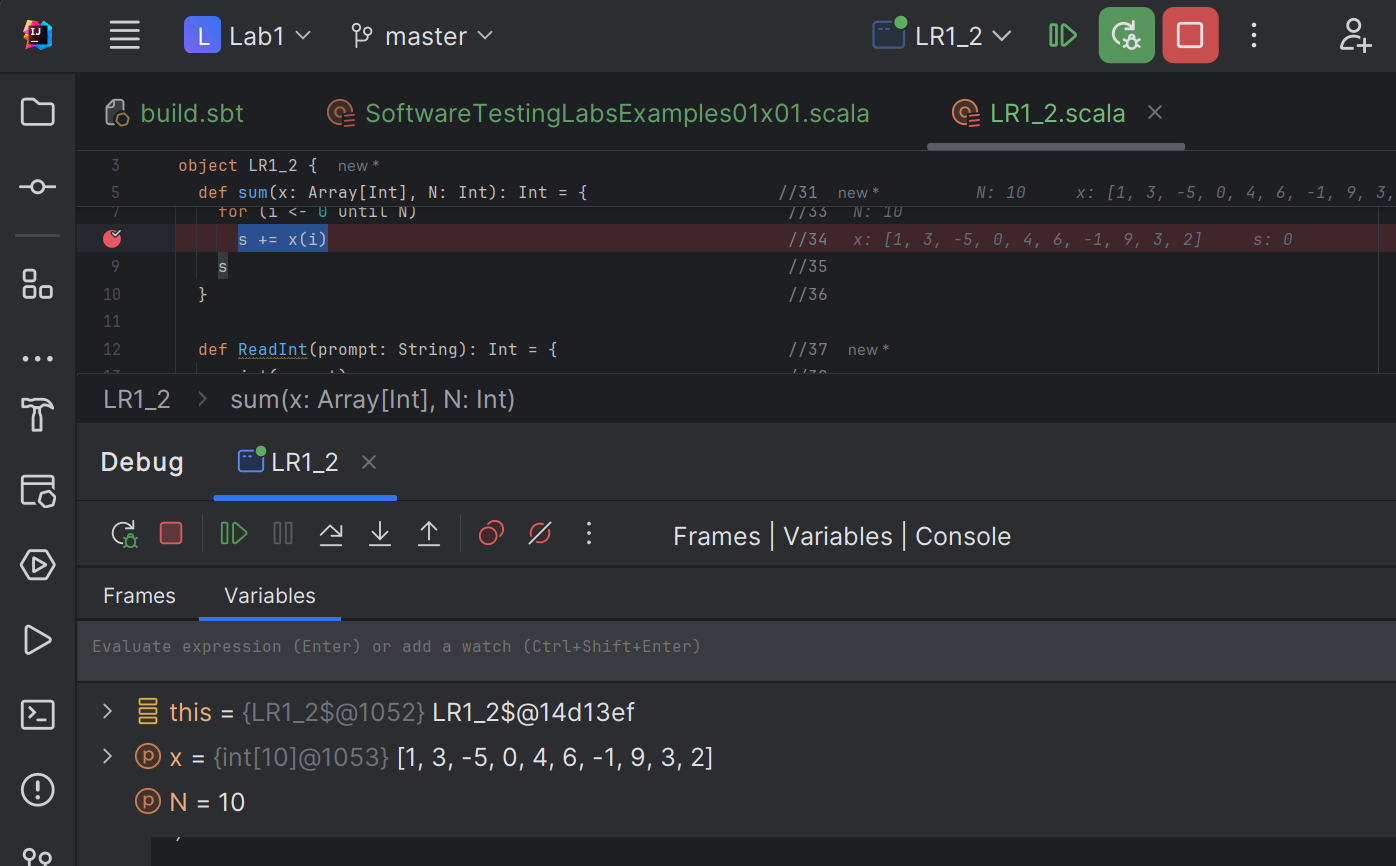


**Часть вторая**

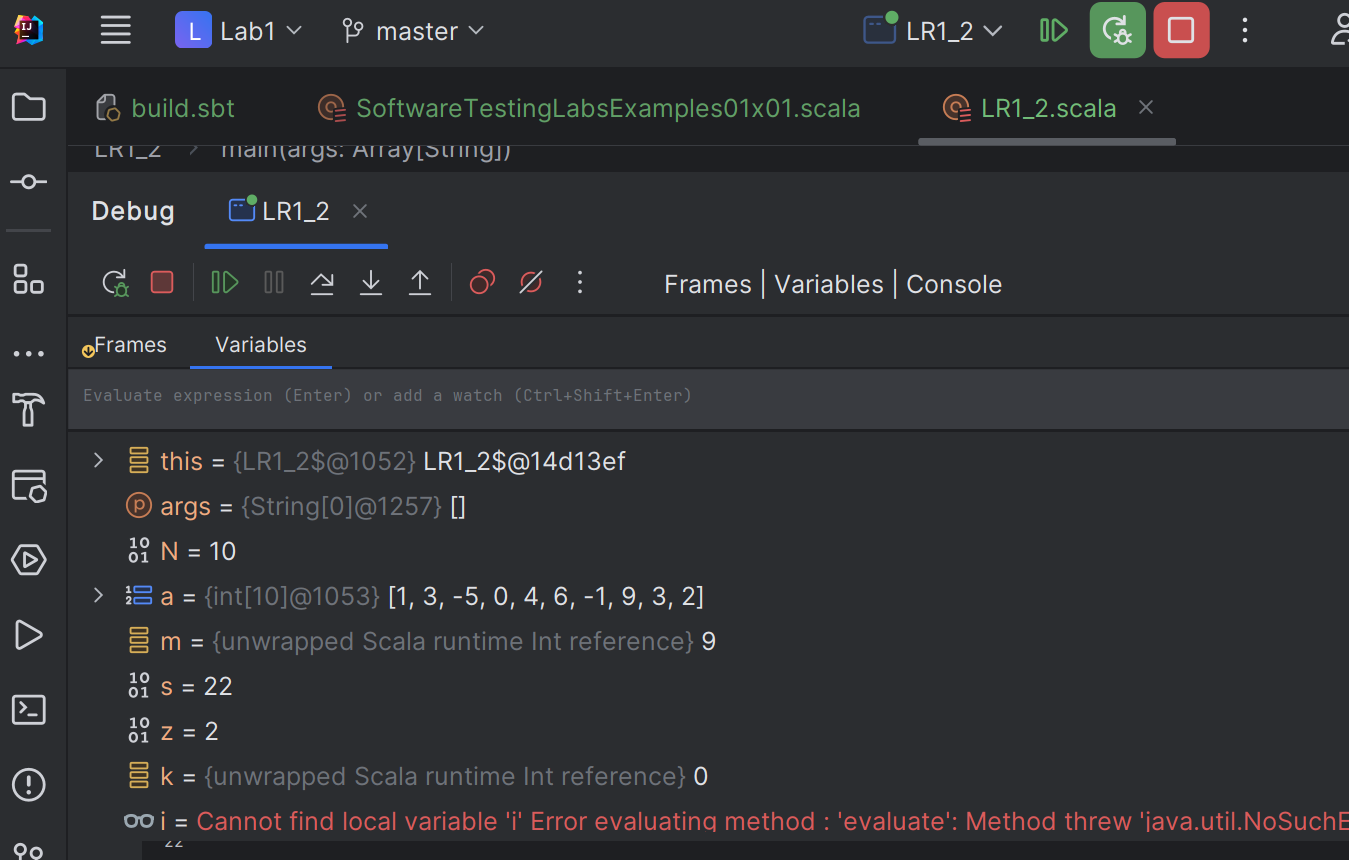
**1 – 2) Нумерация строк программы**



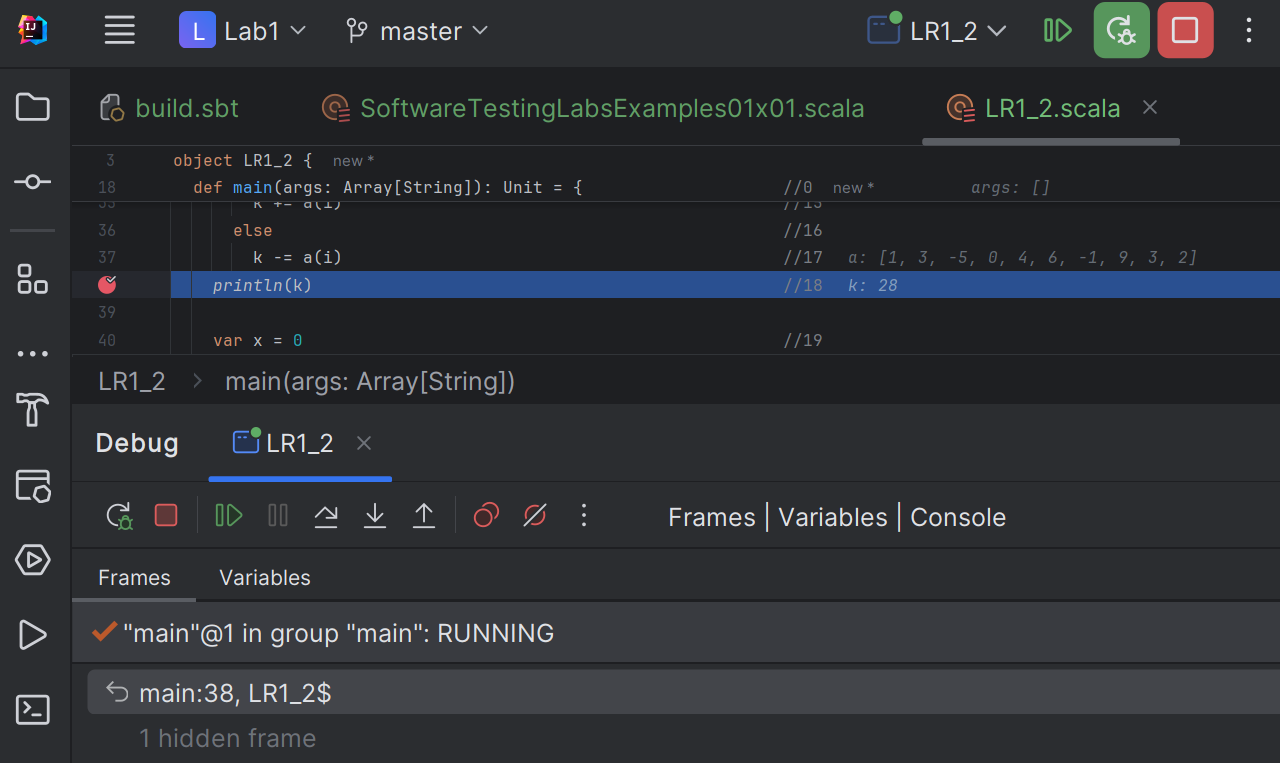
**3) Трассировка программы (F11)**



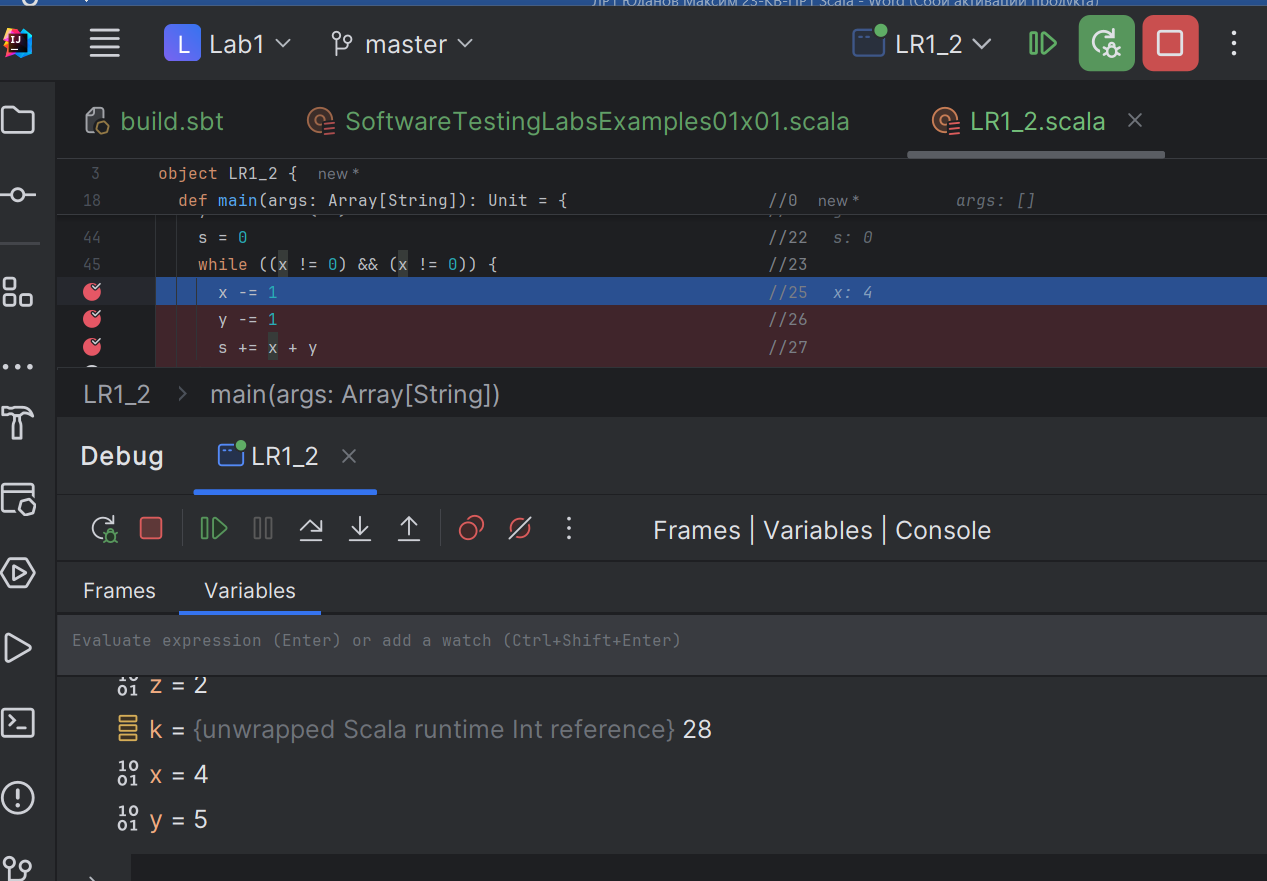
**4 – 6) Выполнение кода до курсора (Ctrl + F10)**



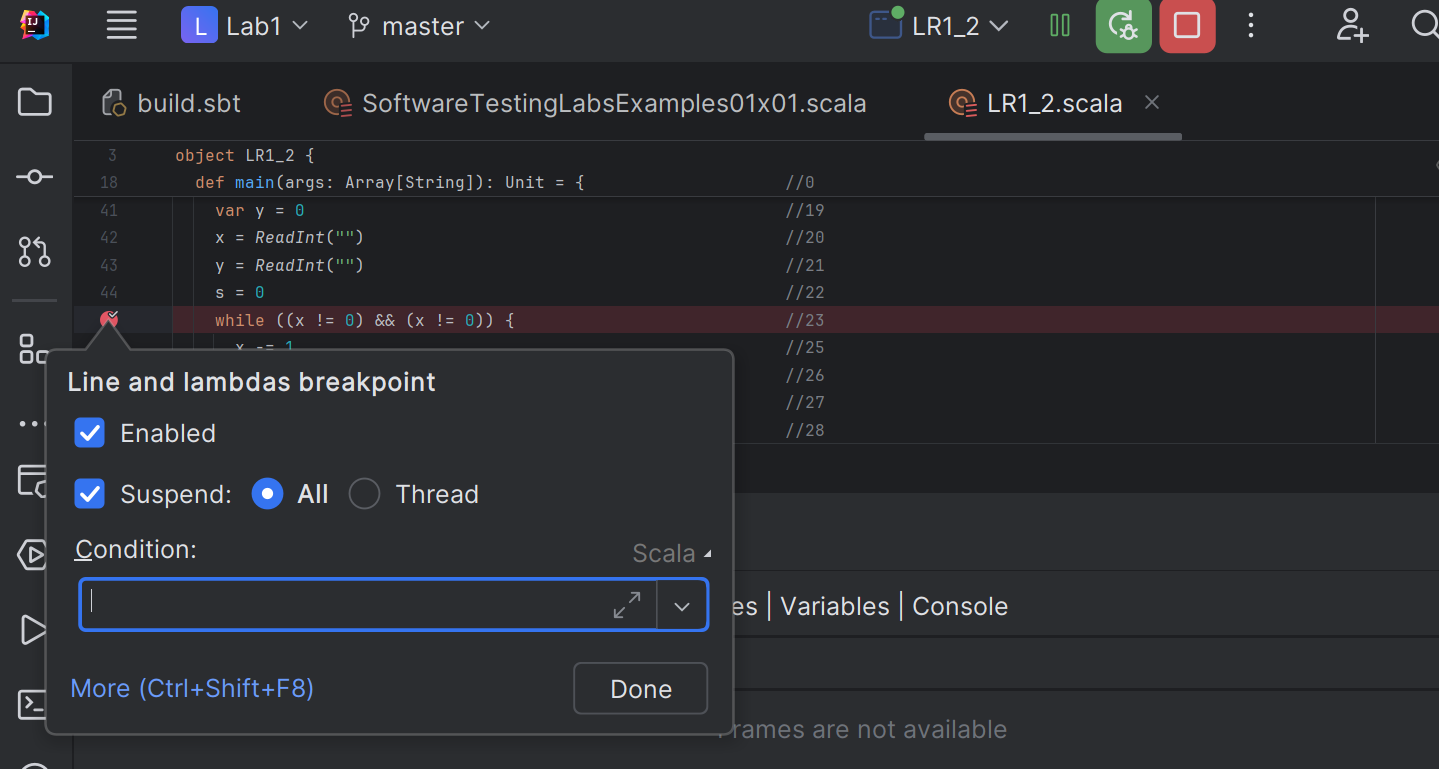
**7) Пошаговое выполнение до строки println(k)**



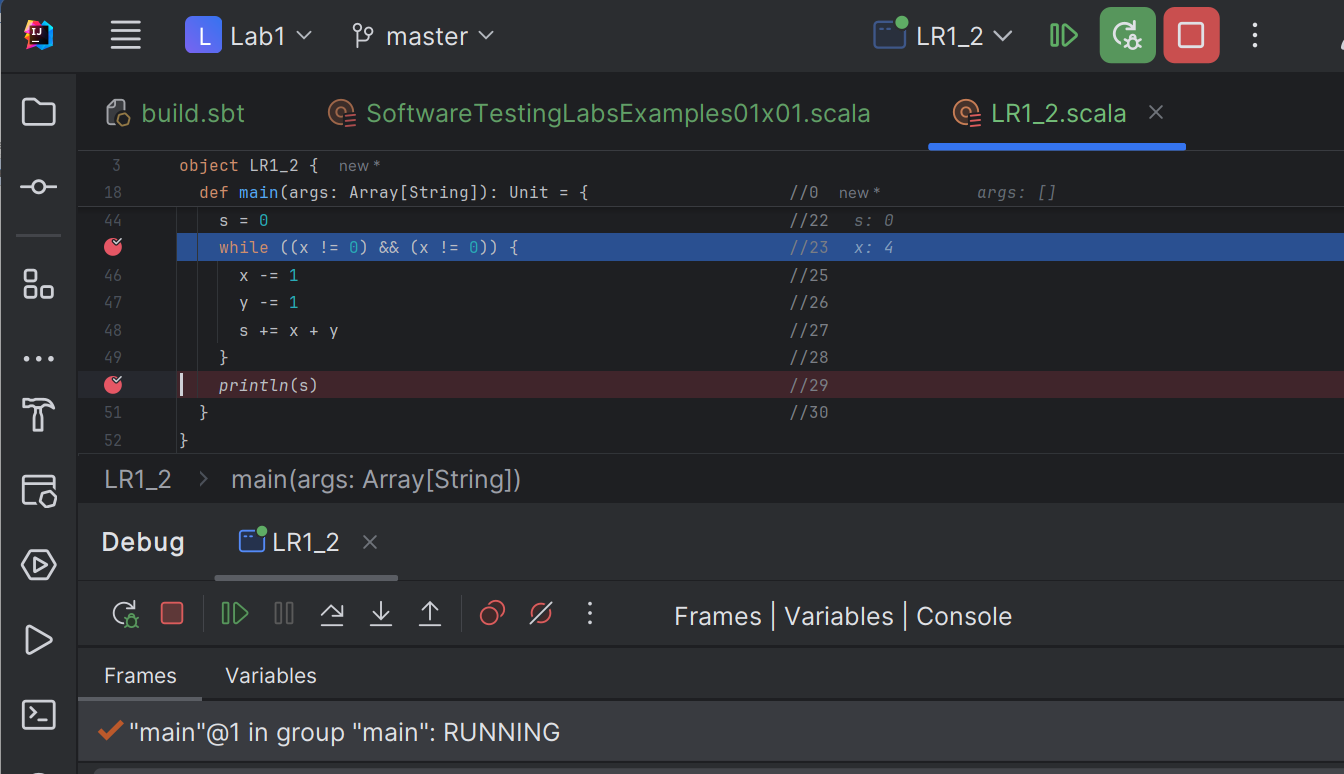
**8) Добавление в окно переменных x, y, s для наблюдения изменения их значений**



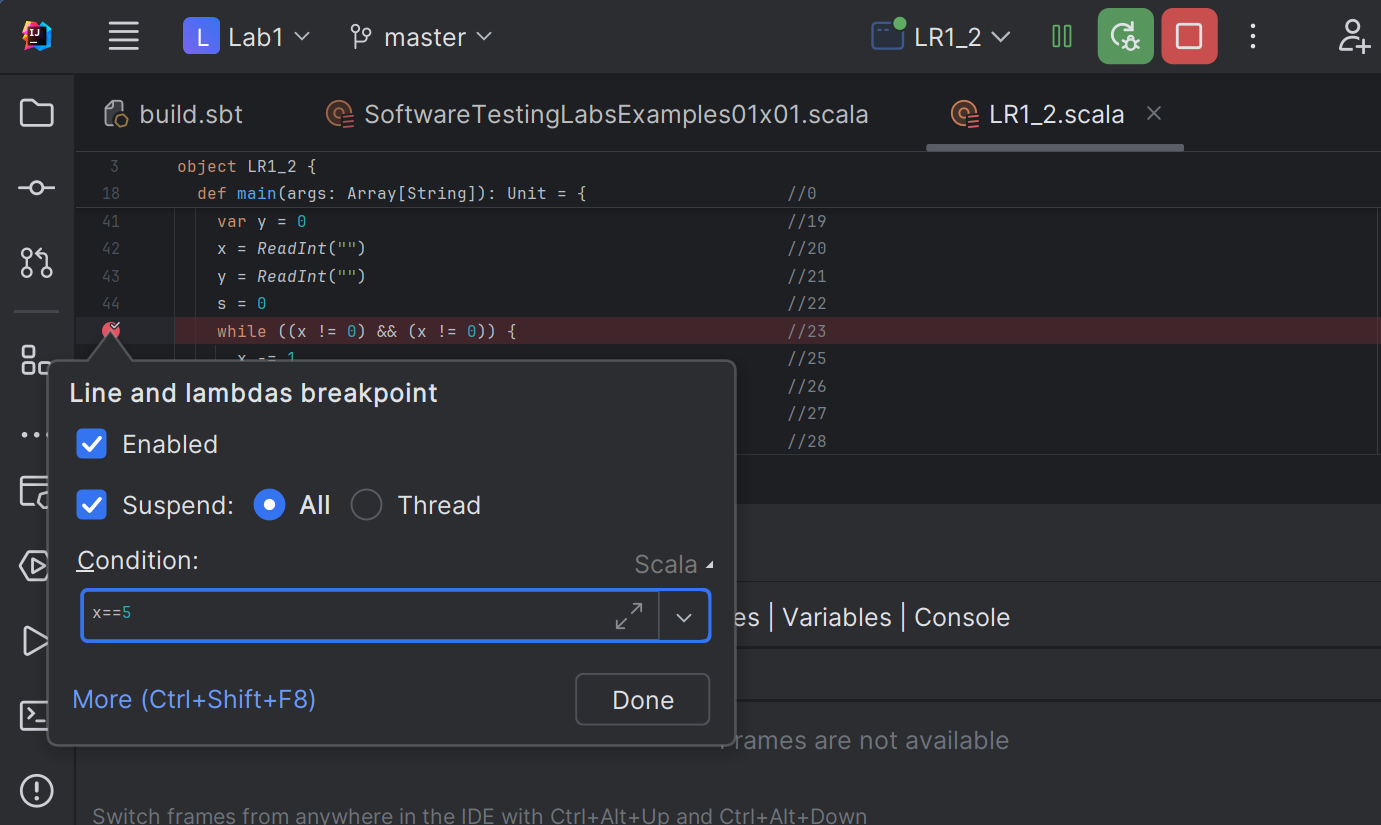
**9 – 10) Создание условной точки останова по числу проходов в while**



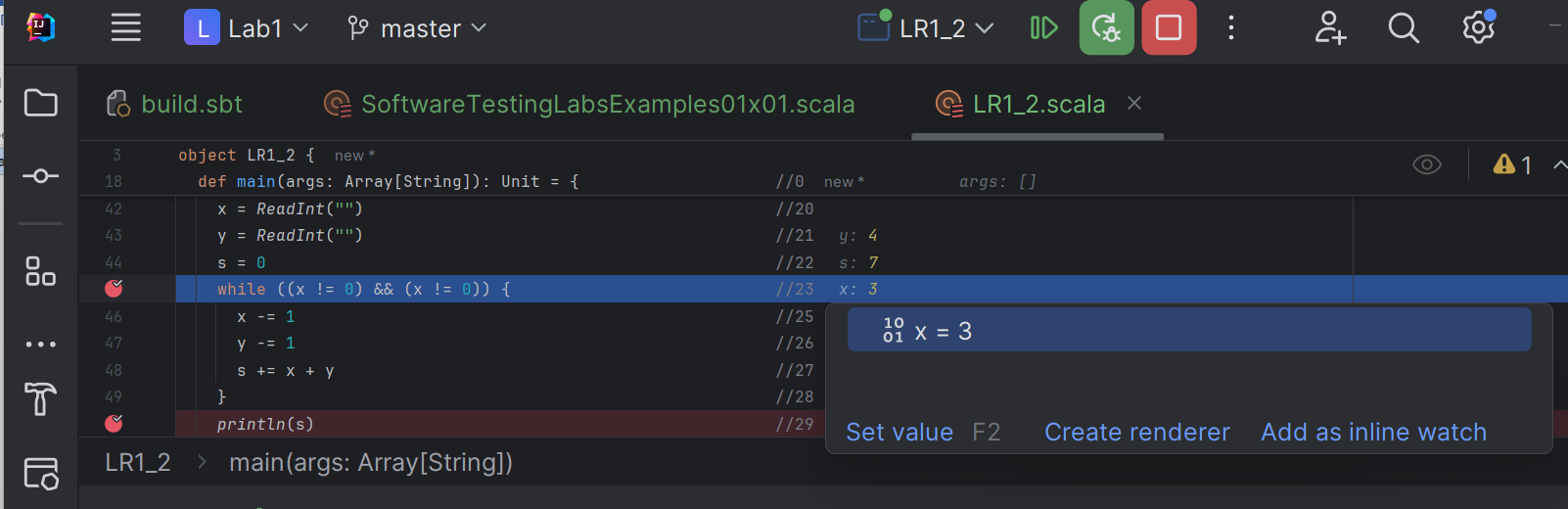
**11 – 13) Результаты x, y, s**



**14) Добавление логического условия в точку остановки в while**



**15 ­– 16)** **Результаты x, y, s**



**Вывод:** в ходе лабораторной работы №1 были изучены инструментальные средства и возможности отладки программ в интегрированной среде Microsoft Visual Studio.