

ЛАБОРАТОРНАЯ РАБОТА «ВВЕДЕНИЕ В СРЕДУ РАЗРАБОТКИ INTELLIJ IDEA. ТЕСТИРОВАНИЕ И ОТЛАДКА ПРОГРАММ НА JAVA»

Цель работы:

- Получить практические навыки установки среды разработки IntelliJ IDEA и комплекта разработчика JDK
- Изучить основные приемы работы в среде разработки IntelliJ IDEA
- Познакомиться со структурой программы на языке Java.
- Научиться составлять простейшие программы на языке Java.
- Получить практические навыки создания, редактирования, тестирования и отладки консольных приложений на языке Java
- Освоить организацию ввода / вывода в консольных приложениях на Java
- Получить практические навыки обработки математических выражений.

Краткие теоретические сведения

В лабораторных работах курса «Программирование на языках высокого уровня» будет использоваться тип приложений операционной системы Windows – приложения с интерфейсом командной строки, так называемые **консольные приложения**.

1. Установка и запуск JDK и IDE IntelliJ IDEA

Для написания и выполнения программ на языке Java необходимо скачать и установить:

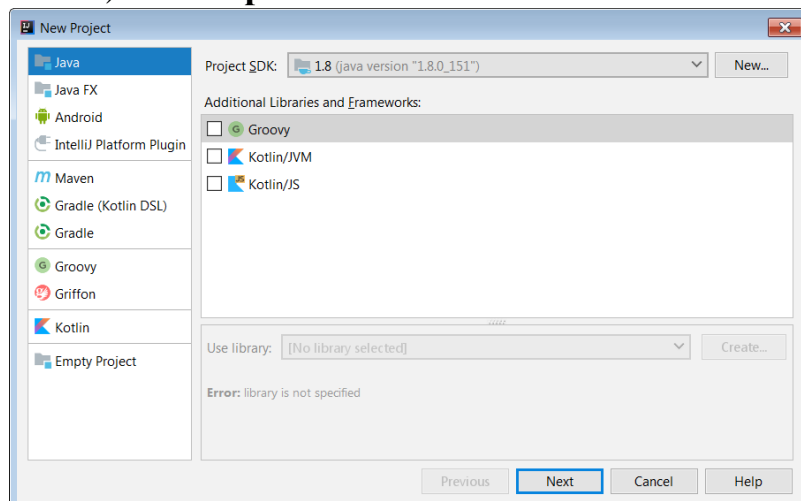
- комплект разработчика Java (JDK) - сайт «oracle.com» – в меню «Downloads» нужно выбрать пункт «Java for Developers» (раздел «Popular Downloads») (*Замечание:* JDK версий 9 и выше ориентированы только на 64-разрядные версии ОС, для 32-разрядных – предназначена версия JDK 8.3)
- среду разработки (IDE) – наиболее популярные:

- IntelliJ Idea – www.jetbrains.com/idea/. Community edition – бесплатная версия
- NetBeans – netbeans.org
- Eclipse – www.eclipse.org

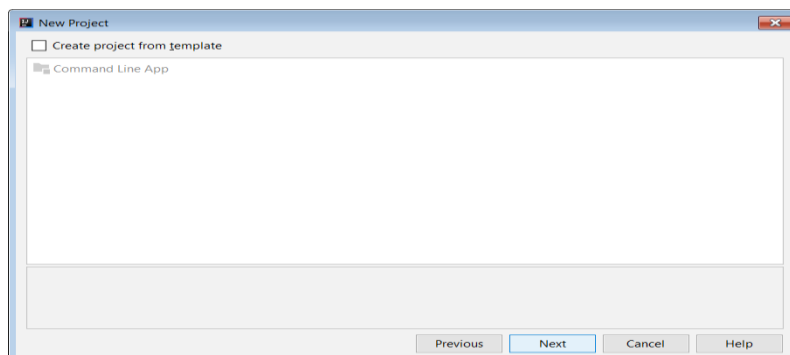
Для написания лабораторных работ на языке Java будем использовать IDE IntelliJ IDEA.

2. Создание нового проекта

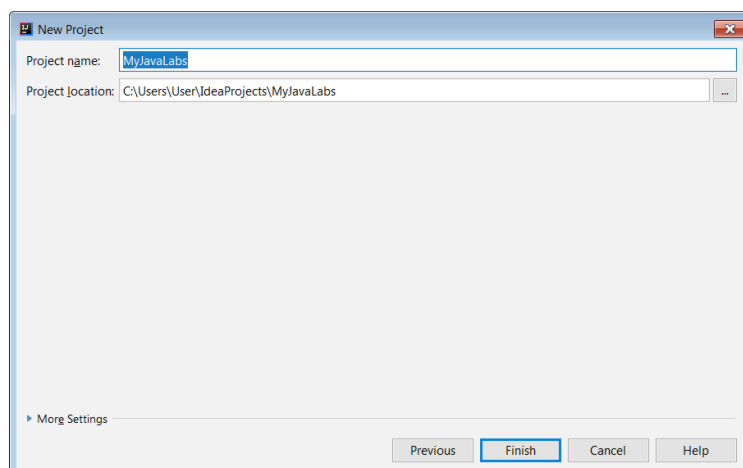
Если не существует проекта, открытого в IntelliJ IDEA, нажмите Create new Project в стартовом окне. В ином случае выберите new Project в меню File. Слева (см. рис. 1а) выбирается подключаемый модуль (который управляет вашим проектом) – **выбираем Java**.



a)



б)



В)

Рис. 1. – Создание нового приложения в IntelliJ IDEA

В выпадающем списке «Project SDK» необходимо выбрать версию JDK, используемую в проекте. Если в списке пусто – добавьте путь к JDK, который вы должны были установить заранее, с помощью кнопки «New...» (откроется обычный диалог открытия файла, в котором необходимо указать путь к папке с установленным JDK). Затем жмем на кнопочку Далее (Next).

Появится окно (рис. 1б), которое позволяет создать приложение по шаблону. Выбирать чекбокс «Create project from template» не нужно – то есть шаблон использовать не будем. Жмем далее.

Следующим шагом (рис. 1в) необходимо указать имя проекта (Project name) – назовите его «MyJavaLabs», и укажите его расположение (Project location) (абсолютный путь к одноименной директории). В зависимости от версии и выбранного модуля среда также попросит указать базовый пакет. Имя базового пакета запишите в формате «ваше имя».«ваша фамилия» (и имя и фамилию записывайте начиная со строчной буквы).

После нажатия кнопки «Finish» откроется созданный вами проект (рис. 2). Поскольку он пуст, никаких файлов открыто не будет (ибо их нет), и среда отобразит подсказки по открытию файлов.

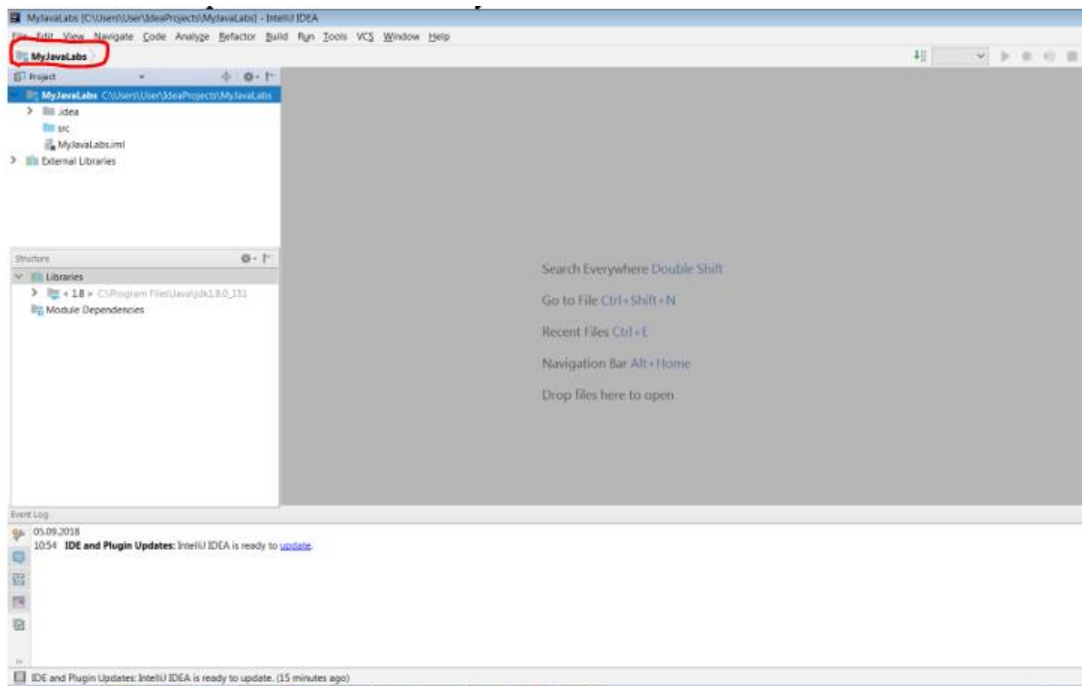


Рис.2 – Рабочее окно в IDE IntelliJ IDEA

Обратите внимание на две панели слева «Project» и «Structure». Первая показывает структуру проекта, вторая – структуру выделенного в первой вкладке элемента проекта.

Структура проекта

Структура проекта отображается на панели «Project» в виде дерева. В корне дерева 2 папки:

- «External Libraries» - содержит те библиотеки, которые были подключены и используются в Вашем проекте (по умолчанию, это библиотеки из комплекта JDK, указанного на этапе создания проекта).
- «MyJavaLabs» - содержит файлы и папки приложения:
 - папка модуля, управляющего вашим проектом (в нашем случае в папке «.idea»),
 - папка «src» - файлы с исходным кодом

Изначально папка **src** пустая. Создадим в ней корневой пакет. Для этого щелкните правой кнопкой по папке **src** и в открывшемся контекстном меню выберите *new / Package*. Откроется окно, в котором потребуется ввести имя создаваемого пакета. **Введите ваше имя латинскими буквами (целиком строчными буквами!!).**

Аналогично создайте внутри этого пакета пакет с использованием уже вашей фамилии, внутри которого создайте еще пакет с именем lab_0.

В примерах ниже предполагается фамилия и имя студента Иванов Иван, а при выполнении работы соответственно следует указывать свои имя и фамилию!

Создайте внутри пакета ivan.ivanov.lab_0 класс с именем «Task1». Аналогично щелкните правой кнопкой по пакету и в контекстном меню выберите New / Class.

Откроется окно, в котором нужно указать имя и тип создаваемого объекта (оставляем Class). Затем жмем «Ok».

После подтверждения откроется окно с исходным кодом созданного вами класса. Его содержимое будет следующим:

```
package ivan.ivanov.lab_0;
public class Task1 {
}
```

Обратите внимание на первую строчку — она определяет принадлежность класса к пакету.

Модифицируйте исходный код следующим образом:

```
package ivan.ivanov.lab_0;
public class Task1 {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Способы запуска программы

- Команда меню Run / Run
- Alt+Shift+F10 для запуска метода *main* главного класса приложения
- Shift+F10 для запуска метода *main* текущего класса

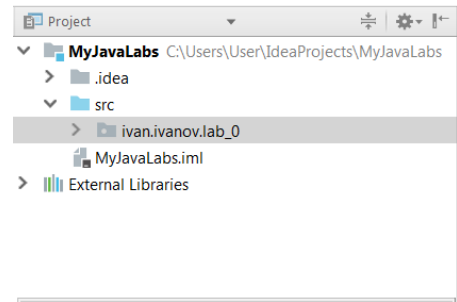


Рис.3 Структура проекта MyJavaLabs

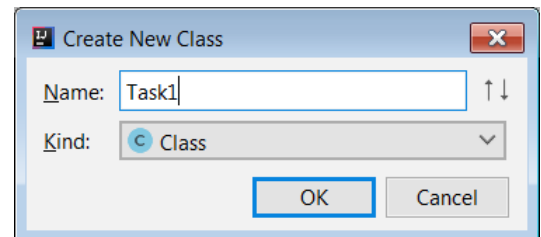


Рис.4. Создание нового класса

При первом запуске программы появится контекстное меню с выбором главного класса приложения (класса, у которого есть метод main). В появившемся окне следует либо выбрать нужный класс из списка, или выбрать пункт Edit Configuration для более детальной настройки (рис. 5).

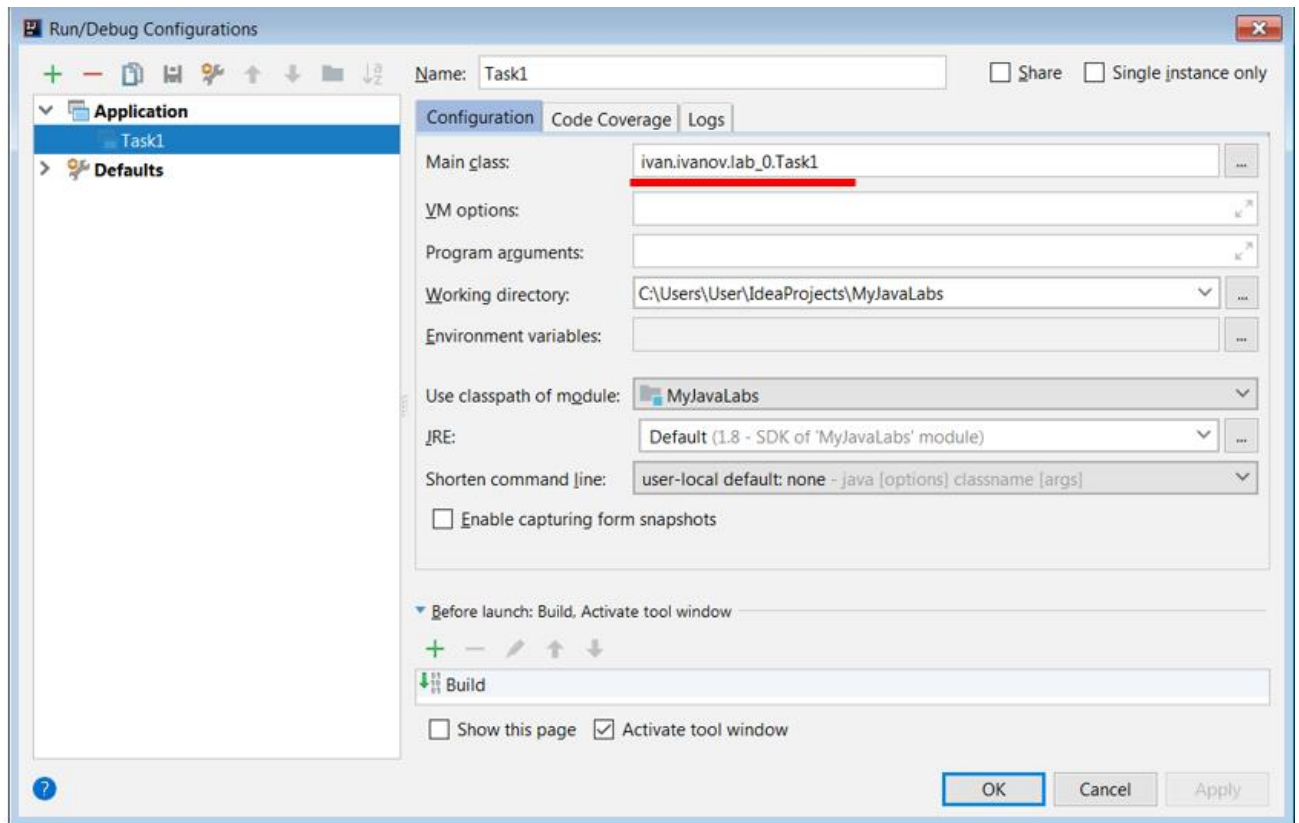
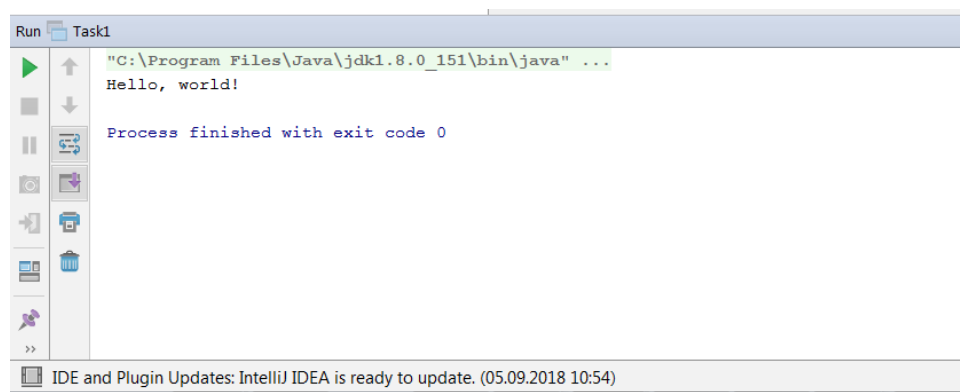


Рис. 5. Окно детальной настройки конфигурации приложения

Для задания главного класса (рис. 5) потребуется указать его полный путь (состоит из имен всех пакетов через точку).

После компилирования файлов с исходным кодом, среда отобразит результат выполнения программы в появившейся внизу вкладке «Run». Последняя строка содержит информацию о коде завершения приложения. 0 – означает, что приложение было выполнено без ошибок.



Для **вывода** информации на экран воспользуемся методами **print()/println()...** стандартного потока вывода **System.out**. Следует отметить, что данные методы позволяют выводить данные только одного типа (другими словами, при выводе выражение вывода приводится к одному типу). Для вывода разнородных данных используется метод **printf()**;

Пример. Требуется написать программу, которая вводит с клавиатуры значения двух целочисленных переменных и выводит на экран их среднее арифметическое.

В листинге программы допущена одна логическая ошибка, которая в следующем разделе будет исправлена!

```
package ivan.ivanov.lab_0;
import java.util.Scanner;
public class Task0 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int x = scanner.nextInt();
        int y = scanner.nextInt();
        System.out.println("Avg(x,y) = "+(x+y/2));
    }
}
```

4. Тестирование и отладка программы

Для выявления и устранения ошибок среды программирования предоставляют мощные инструменты, такие как диагностические сообщения компилятора и отладчик. Программа может содержать ошибки трех видов:

1. *Синтаксические ошибки*, связанные с неправильным написанием языковых конструкций, например, если вместо **System** написать **system** и т.д. Синтаксические ошибки всегда отлавливаются компилятором и об их наличии можно судить по диагностическим сообщениям компилятора;
2. *Семантические ошибки* (или ошибки периода выполнения – **Runtime Errors**), связанные с некорректным использованием языковых конструкций; например, к таким ошибкам относится переполнение

буфера и выход за границы массива. Семантические ошибки приводят к аварийному завершению программы (говорят, что программа «падает» или «вылетает») Такие ошибки компилятором не отлавливаются.

3. *Логические ошибки*, связанные с применением некорректных алгоритмов, например, если при вычислении среднего значения двух чисел вместо оператора деления использовать оператор умножения. В результате таких ошибок программа продолжает нормально работать, однако переменные содержат неправильные значения, может быть выбрана неправильная ветвь выполнения алгоритма. Такие ошибки наиболее сложны для исправления.

Единственным способом поиска семантических и логических ошибок является **процесс тестирования**. На этапе тестирования осуществляют пробный запуск программы, передавая ей в качестве входных данных некоторый predetermined набор данных, и сравнивая результат работы программы с заранее известным (ожидаемым) результатом. Если результат работы программы и ожидаемый результат не совпадают, делается вывод о наличии ошибки и начинается процесс исправления ошибки, называемый **отладкой**.

4.1 Поиск и выделение синтаксических ошибок в программе

Синтаксические ошибки среда **IntelliJ IDEA** обнаруживает до этапа компиляции и выделяет их местоположение (рис. 7).

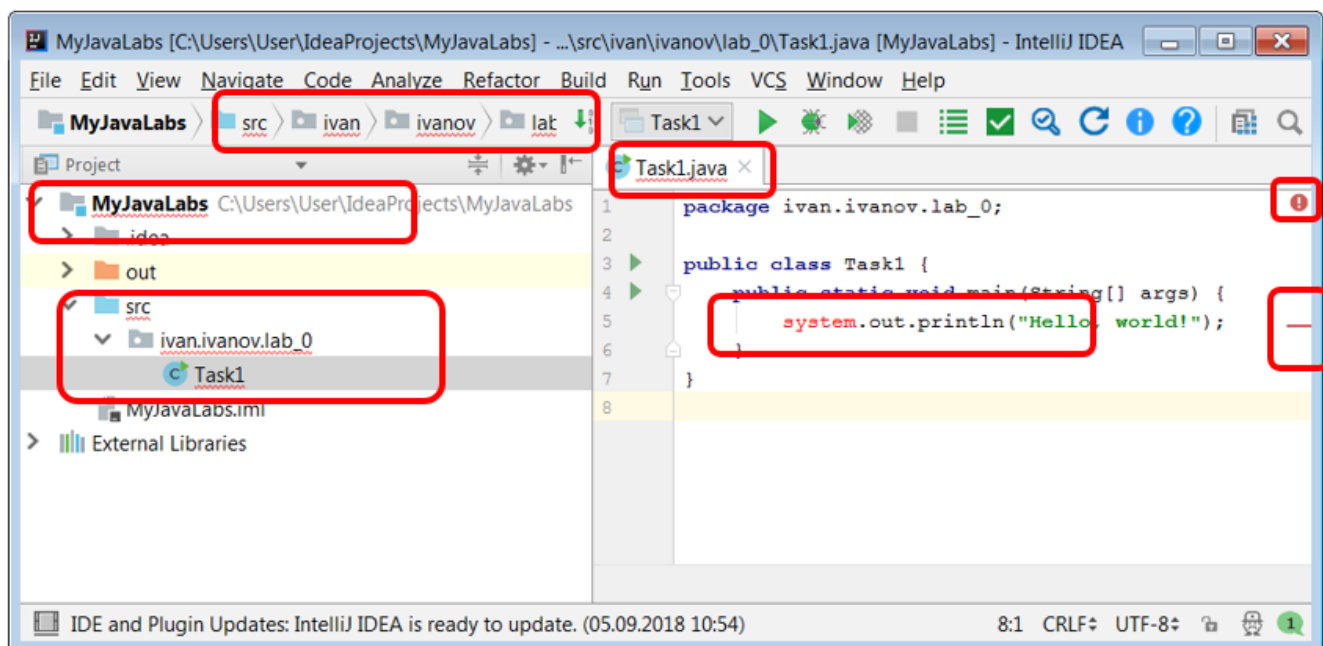


Рис. 7. Выделение синтаксических ошибок программы в среде IntelliJ IDEA

Если несмотря на имеющиеся ошибки все-таки запустить приложение, то получим сообщение компилятора об ошибке (рис. 8) с указанием ее местоположения (номер строки и номера символа в строке). Для перехода к месту ошибке достаточно дважды щелкнуть по ней.

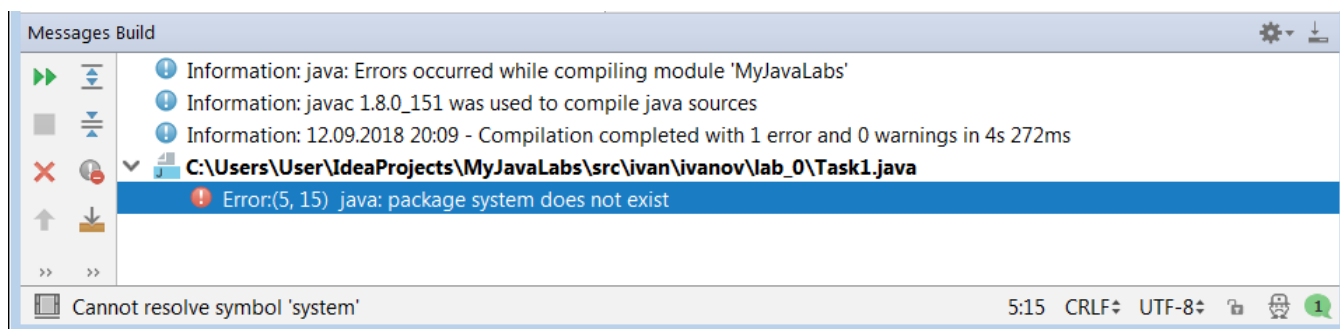


Рис. 8. Выделение синтаксических ошибок программы в среде IntelliJ IDEA

4.2 Тестирование программы

Тестирование — это проверка, которая позволяет определить: соответствует ли реальное поведение программы ожидаемому, выполнив специально подобранный набор тестов.

Тест — это выполнение определенных условий и действий, необходимых для проверки работы тестируемой функции или её части.

Набор тестов должен быть простым и полным для активизации всех ветвей алгоритма, но не избыточным, то есть таким, что удаление из этого набора любого теста лишает его полноты.

Для приведенного выше примера подготовим тестовый набор данных и запишем их в виде таблицы:

| Первое значение (x) | Второе значение (y) | Ожидаемый результат |
|---------------------|---------------------|---------------------|
| 0 | 10 | 5 |
| 10 | 0 | 5 |
| 1 | 2 | 1,5 |

Выполним программу для тестового набора данных, результаты выполнения сведем в таблицу:

| Первое значение (x) | Второе значение (y) | Ожидаемый результат | Результат программы | Тест пройден |
|---------------------|---------------------|---------------------|---------------------|--------------|
| 0 | 10 | 5 | 5 | Да |
| 10 | 0 | 5 | 10 | Нет |
| 1 | 2 | 1,5 | 2 | Нет |

Если в результате тестирования имеются не пройденные программой тесты, требуется **отладка** программы.

4.3 Отладка программы

Отладка – это локализации и устранения ошибок. Для локализации нужно выяснить по какому пути проходит программа и знать значения переменных в определённые моменты работы приложения. Нужно выставить **точки останова (breakpoints)** - в левом поле редактора напротив требуемой строчки кода ЛКМ (Левая Кнопка Мыши), можно использовать Ctrl-F8 (в IntelliJ IDEA).

Наиболее распространенные режимы отладки:

- **Пошаговое** выполнение программы: Step Over (F8) / Trace Into (F7)
- Выполнить **до курсора** (сочетание клавиш Alt+F9), затем пошаговое

- Использование **точек останова (breakpoints)** – выставляются в левом поле редактора напротив требуемой строчки кода ЛКМ (Левая Кнопка Мыши), можно использовать Ctrl-F8 (в IntelliJ IDEA)
- и др.

Комментарии, объявления полей и методов, и пустые строки не подходят для точек останова, на строках с объявлением переменной без присвоения значения (в Java) не будет срабатывать точка остановки.

Отладка программ в среде IntelliJ IDEA

Запуск программы в режиме отладки – Shift+F9 или Alt+Shift+F9

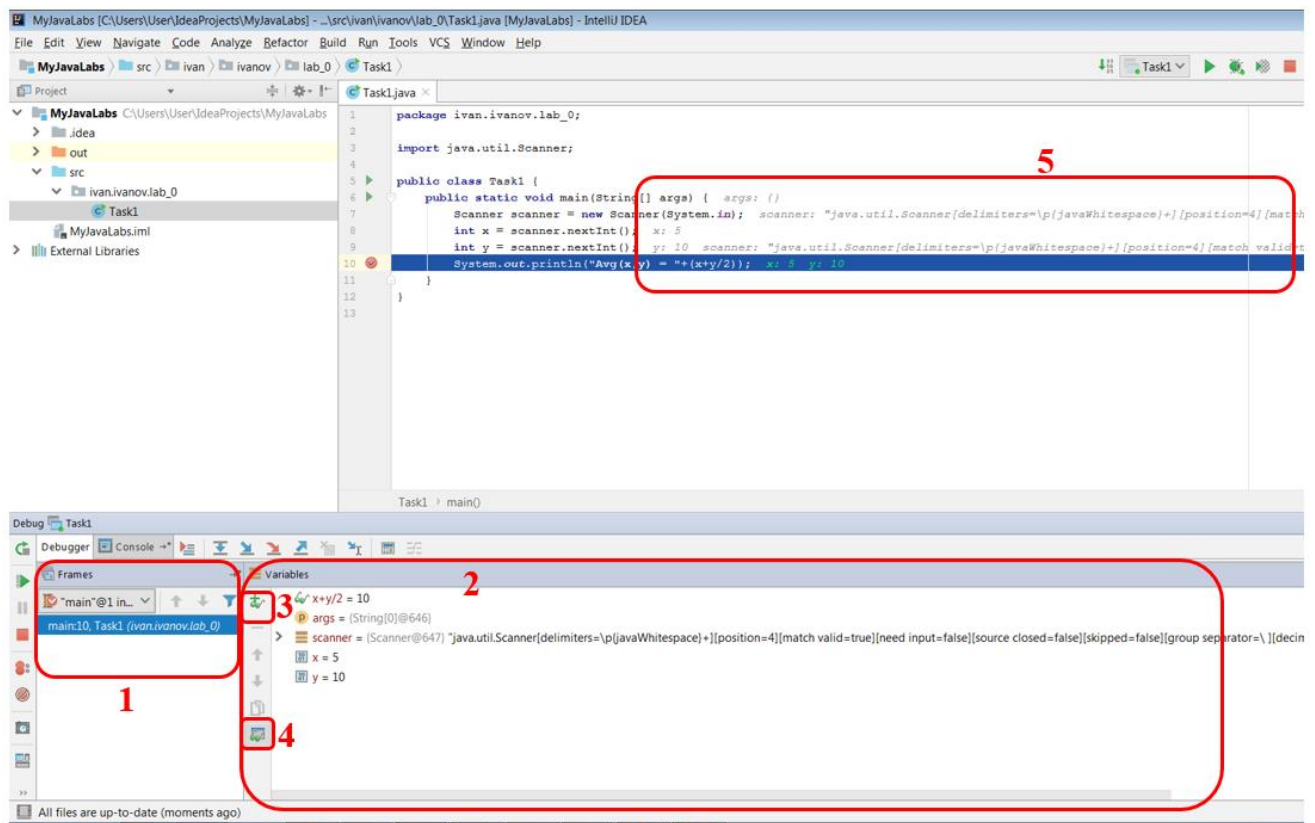



Рис. 9. Вид рабочего окна приложения в режиме отладки

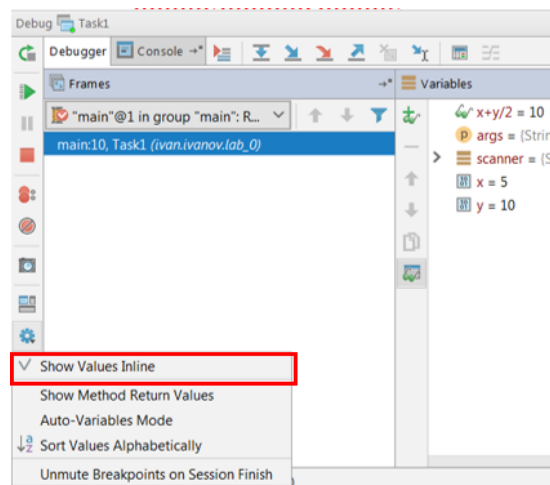
Слева внизу окно "Frames" (рис.9, метка 1) со стеком вызовов классов в виде иерархии. Можно переходить по стеку вызовов и смотреть с какими аргументами вызывались методы.

В центре внизу окно "Variables" (рис.9, метка 2) с переменными, которым присвоено значение. При изменении значения у переменной изменяется цвет (на синий) у самой переменной в окне "Variables".

Окно "Watches" может располагаться отдельно или внутри окна "Variables" при включенной опции  (Show watches in variables tab). Окно "Watches" предназначено для наблюдения за значениями определённых добавленных в это окно переменных или выражений, без необходимости перемещаться по древовидным структурам в окне "Variables". Можно добавлять имена переменных зелёным плюсом или Alt-Insert.

Можно вносить изменения в код, затем перекомпилировать изменённый класс Ctrl-Shift-F9 и продолжить отладку без её полного перезапуска.

Можно включить в настройках отладчика отображение значений переменных прямо в строках кода: **"Show values inline"** (рис.9, см. метка 5 в предыдущем рисунке).



Можно быстро отключить точку останова если щёлкнуть по ней ЛКМ с ALT – точка получит зелёную метку.

Подробнее об использовании отладчика в IntelliJ IDEA в приложении 2.

5. Обработка математических выражений

Большинство арифметических и математических стандартных методов описаны в классе Math пакета java.lang. Некоторые из них приведены в приложении 1.

Пример 1

Требуется вычислить значение выражения
$$S = \frac{3 \cos ax^2 - \ln a}{e^4 - bx} + b \sin^2 x^6$$

где $a=5$, $b=6$ – константы

```
package ivan.ivanov.lab_0;
```

```

import java.util.Scanner;
public class Task1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double x = scanner.nextDouble();
        final int A=5; //объявляем целую константу A
        final int B=6;
        double s = (3*Math.cos(A*x*x) - Math.Log(A))/
            (Math.exp(4) - B*x) +
            B*Math.pow(Math.sin(Math.pow(x,6)),2);
        System.out.println("S = "+s);
    }
}

```

Пример 2

Написать программу, в которой вводятся с клавиатуры вещественные числа и печатается значение выражения $y = e^{\sqrt{\cos(bx)+x}} \cdot \sin\left(\frac{\sqrt{ax+1}}{c}\right)$

```

package ivan.ivanov.lab_0;
import java.util.Scanner;
public class Task2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double x = scanner.nextDouble();
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();
        double y = Math.exp(Math.sqrt(Math.cos(b*x)+x))*
            Math.sin(Math.sqrt(a*x+1)/c);

        //вывод результата с точностью до 3-х знаков после запятой
        System.out.printf("y = %5.3f",y);
    }
}

```

Задание на лабораторную работу

Составить консольные приложения, реализующие приведенные задания на языке Java. В отчет включить листинги созданных программ и результаты

выполненных расчетов (скриншоты выделенного окна – сочетание клавиш Alt+PrtSc). Описание базовых математических методов класса java.lang.Math приведены в ПРИЛОЖЕНИИ 1.

Задание 1. Вычислить значения функции при указанном значении аргумента x . Значения a , b и c задать в виде констант.

В отчет поместить результаты вычислений, полученные после выполнения программы.

1) Вычислить в вещественных числах значение выражения

| Номер варианта | $y = f(x)$ | Исходные данные |
|-------------------|---|--|
| 1 | $y = \frac{\sqrt{cx + 62,7e^x}}{ax^2 + 7x + b \ln x}$ | $a = 7,2; b = 14,3$ $c = 13,4; x = 5,6$ |
| 2 | $y = \frac{ax + 3,8 \operatorname{tg} x}{\sqrt{bx^3 + c}}$ | $a = 1,23; b = 5,14$ $c = 3,97; x = 7,1$ |
| 3 | $y = \left(\frac{a}{bx^2 + 1} + cx^3 + b \sin^2 x \right)^2$ | $a = 2,27; b = 1,18$ $c = 3,92; x = 0,78$ |
| 4 | $y = \left(a\sqrt{4,19x^3 - 1} - \sqrt{b \ln x + c} \right)^{-1}$ | $a = 9,2; b = 3,5$ $c = 12,3; x = 3,2$ |
| 5 | $y = \ln \left a \sin x + b \cos(x^2) \right $ | $a = 1,2; b = 2,3$ $x = 5,6$ |
| 6 | $y = \sqrt{\frac{ax^3 + \operatorname{arctg} x}{cx + b \ln x }}$ | $a = 2,71; b = 1,63$ $c = 0,81; x = 0,51$ |
| 7 | $y = \frac{ax}{\sqrt{b^2 + 2e^x - bx}}$ | $a = 6,32; b = 3,704$ $x = 7,15$ |
| 8 | $y = \cos(ax) + b \ln(1 + bx + e^x)$ | $a = 7,1; b = 1,8$ $x = 0,9$ |
| 9 | $y = \frac{\sqrt{e^{ax} + x^2} \cdot \ln(x^2 + bx + e^{10})}{\sin(cx) + 4,2}$ | $a = 5,7; b = 6,4$ $c = 3,1; x = 2,8$ |
| 10 | $y = \frac{\sqrt{e^{2x+b}} - 1,7 \cos(cx)}{\ln(x^2 + a)} + x^3$ | $a = 2,1; b = 5,3$ $c = 1,4; x = 1,2$ |

| Номер варианта | $y = f(x)$ | Исходные данные |
|-------------------|---|---|
| 11 | $y = \frac{\ln \sqrt{x^2 + b} + cx^3}{e^x + a} - x^3$ | $a = 4,7; b = 7,21$ $c = 1,72; x = 0,91$ |
| 12 | $y = \frac{\sin \sqrt{e^x + ax^2 + b \ln x + cx^3}}{ax^2 + cx + 13,7}$ | $a = 3,7; b = 4,9$ $c = 2,5; x = 1,3$ |
| 13 | $y = \sqrt{\frac{a}{1 + bx^2}} + b \operatorname{ctg} x + \frac{e^{ax}}{3 - x}$ | $a = 4,5; b = 2,2$ $c = -1,5; x = 0,85$ |
| 14 | $y = \frac{(cx)^2 - e^{bx}}{\sqrt{x + 3a^3} + \cos(ax)}$ | $a = 4,5; b = 2,2$ $c = 1,67; x = 2,36$ |
| 15 | $y = \frac{\sin(x^2 + a^2) \cdot e^{b+x}}{\sqrt{ax^3 + x}} + \cos^2(ax)$ | $a = 4,26; b = 1,71$ $c = 3,86; x = 2,73$ |
| 16 | $y = \frac{\sqrt{\ln^2(ax + 2) + \sin(bx^2 - 1)}}{x^3} - 6,3a^2$ | $a = 4,3; b = 2,9$ $x = 1,8$ |
| 17 | $y = \sqrt{x + e^{ax}} \cdot \ln \frac{bx^2 - 1}{cx^2 + 3}$ | $a = 2,44; b = 1,39$ $c = 6,21; x = 3,1$ |
| 18 | $y = \frac{2\sqrt{\sin(ax^3 + 3) + bx^2}}{e^{-x} + 3,2} + 5,6x$ | $a = 4,17; b = 3,69$ $x = 1,2$ |
| 19 | $y = \frac{\sqrt{ax^2 + bx^3 + 9,2}}{3x} - \ln(2 + \cos^2 x)$ | $a = 6,27; b = 2,73$ $x = 2,83$ |
| 20 | $y = e^{\sqrt{ax + \operatorname{tg}(bx)}} \cdot \sin\left(\frac{\sqrt{ax + 1}}{c^2}\right)$ | $a = 2,13; b = 4,7$ $c = 2,6; x = 1,2$ |
| 21 | $y = \frac{x^{\ln x} \cdot e^{\sqrt{ax + b^2}}}{2a} - \sin\left(\frac{\sqrt{ax + 1}}{c^2}\right)$ | $a = 3,2; b = 1,67$ $x = 3,49$ |
| 22 | $y = \frac{a\sqrt{ax} + b \cos x}{2 + e^{\sqrt{ax + b^2}}} - \sin^2\left(\frac{ax + 1}{c^2}\right)$ | $a = 2,71; b = -6,23$ $c = 3,34; x = 2,43$ |
| 23 | $y = \frac{ax^2 + \sqrt{\ln x + a^2}}{b \cos x + 4,7} - \sin(a^3 x)$ | $a = -1,83; b = -2,15$ $x = 3,57$ |
| 24 | $y = \frac{\operatorname{arctg}(e^{ax}) + \frac{\ln(b + x)}{x^3}}{b^2 \cos x} + (a^3 x)^2$ | $a = 0,21; b = 2,19$ $x = 3,74$ |

| Номер варианта | $y = f(x)$ | Исходные данные |
|-------------------|--|--|
| 25 | $y = \frac{a \cos x + b e^{\sin x} + \frac{\ln(b+x)}{x^3}}{\ln x + c x^4} - \sin^2(a^3 x)^2$ | $a = 1,93; b = 3,48$ $c = 0,27; x = 1,44$ |

Задание 2. Написать программу, в которой вводятся с клавиатуры все неизвестные вещественные числа и печатается значение выражения

| Номер варианта | Выражение | Номер варианта | Выражение |
|-------------------|---|-------------------|--|
| 1 | $p = \frac{\lg(x) - e^{x+y}}{\sqrt{2+y^2} + x^3 + \ln(y) }$ | 14 | $y = \left(a \sqrt{4,19x^3 - 1} - \sqrt{b \ln x + c} \right)^{-1}$ |
| 2 | $a = \ln\left(y^{-\sqrt{x}}\right) \left(\sin(x) + e^{x+y} \right)$ | 15 | $y = \ln a \sin x + b \cos x^2 $ |
| 3 | $b = c \sqrt{y + x^2} \left(\cos(x) - c - y \right)$ | 16 | $y = \sqrt{\frac{ax^3 + \arctg(x)}{cx + b \ln x }}$ |
| 4 | $c = \arctg(x) - 3 \cdot e^{-xy} + 0,5 \frac{ x+y }{(x+y)^b}$ | 17 | $y = \frac{ax}{\sqrt{b^2 + 2e^x - bx}}$ |
| 5 | $d = \frac{e^{x-y} \lg(z)}{\arctg(y) + \sqrt{x}} + \ln x$ | 18 | $y = \cos(ax) + b \ln(1 + bx + e^x)$ |
| 6 | $e = \frac{(\cos x - \sin y)^3}{\sqrt{\lg(x)}} + \ln^2(x \cdot y \cdot z)$ | 19 | $y = \frac{\sqrt{e^x + x^2} \cdot \ln(x^2 + bx + 10)}{\sin(cx) + 4,2}$ |
| 7 | $f = x + \sqrt{ x + e^y} - \frac{z^3 \sin^2 y}{y + z^2(y-x)}$ | 20 | $y = \frac{\sqrt{e^{2x+b}} - 1,7 \cos(cx)}{\ln(x^2 + a)} + x^3$ |
| 8 | $g = \frac{1 + \cos(x+y)}{ e^x - 2y(1 + x^2 y^2) } x^3 + \arcsin y$ | 21 | $y = \frac{\ln \sqrt{x^2 + b} + cx^3}{e^x + a}$ |
| 9 | $h = 2 + \frac{x^2}{\sqrt{2}} + \frac{ y^3 }{\sqrt{3}} + \frac{z^4 (\ln x + 1) \sqrt{2}}{\sqrt{5}}$ | 22 | $y = \frac{\sin \sqrt{e^x + ax^2 + b \ln x}}{ax^2 + cx + 13,7}$ |
| 10 | $j = (1+y) \sqrt{\sin^2 z} - \frac{ y-x }{5}$ | 23 | $y = \sqrt{\frac{a}{1+bx^2}} + b \operatorname{ctg} x + e^{cx}$ |
| 11 | $y = \frac{\sqrt{cx + 62,7e^x}}{ax^2 + 7x + b \ln x}$ | 24 | $y = \frac{(cx)^2 - e^{bx}}{\sqrt{x} + \cos(ax)}$ |

| | | | |
|-----------|---|-----------|---|
| 12 | $y = \frac{(ax + 3,8tg\ x)^2 - \sin x}{\sqrt{bx^3 + c}}$ | 25 | $y = \frac{\sin(x^2 + a^2) \cdot e^{b+x}}{\sqrt{ax^3 + c}}$ |
| 13 | $y = \left(\frac{a}{bx^2 + 1} + cx^3 + b \sin^2 x \right)^2$ | | |

Контрольные вопросы

1. Состав технологии JAVA. Назначение JDK, JRE, JVM
2. Порядок выполнения программ на Java
3. Приведите варианты запуска программы на языке Java
4. Структура проекта приложения в среде IntelliJ IDEA
5. Правила отладки программ на языке Java
6. Понятие главного класса в Java и способы его задания
7. Организация консольного ввода/вывода
8. Импорт классов