*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_Компьютерные Системы и сети (ИУ6)\_\_\_\_\_\_\_\_\_\_\_

**Отчет**

**по лабораторной работе № \_\_1\_\_**

**Название лабораторной работы: \_\_\_ Создание консольных приложений**

\_\_\_\_\_\_\_\_\_\_**в среде Turbo Delphi 2006\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дисциплина:** \_\_\_\_\_\_\_ Основы программирования \_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент гр. \_ИУ6Ц-32Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Горяйнов Т.В\_\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Веселовская О.А.\_\_\_\_**

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Цель работы: Создать консольное приложение для вычисления корней квадратного уравнения, Изучить диагностические сообщения Turbo Delphi, Изучить средства отладки программ в среде Delphi.

**Задание 1**

**Создать консольное приложение для вычисления корней квадратного уравнения.**

Порядок работы:

1. Для создания заготовки консольного приложения выберите пункт **File/New/ Other…** (пункт **File** подпункт **New** и подпункт **Other…**). На экране появляется окно выбора создаваемых проектов (см. рисунок 3).

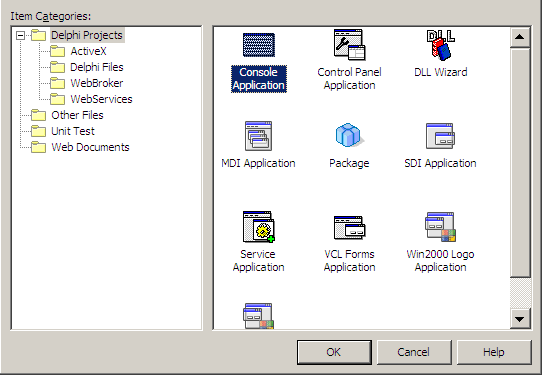
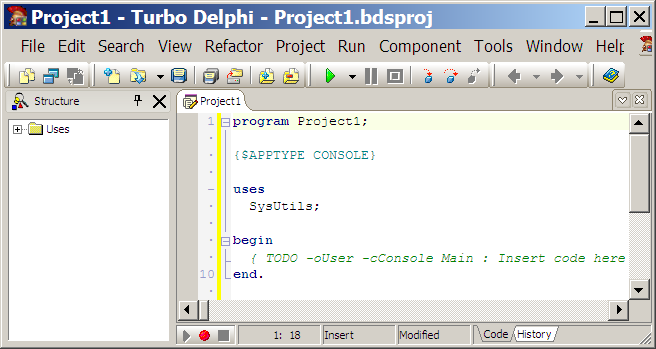


Рисунок 3 *–* Выбор Консольного приложения

Выберите иконку **Console Application**. После этого на экране появится заготовка консольного приложения со стандартным именем **Project1** (см. рисунок 4).



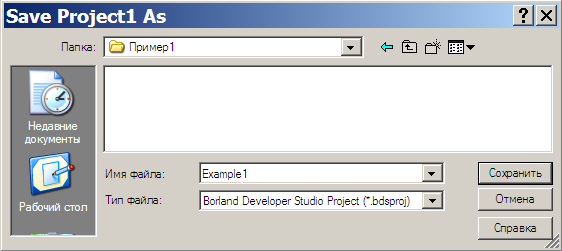
**Рисунок 4** – Вид окна среды с заготовкой консольного приложения и

без неиспользуемых окон

2. В рабочей области на экране теперь два окна: панель структуры и вкладка многооконного редактора текстов программ.

В окне структуры отображается дерево объектов программы: переменных функций, типов, библиотек и т. п. Его используют для быстрого перехода к описанию какого–либо объекта. Если щелкнуть мышью по плюсику у папки, то мы увидим ссылку только на библиотеку **SysUtils**. А двойной щелчок по этой ссылке установит курсор на объявление этой библиотеки в программе.

3. Созданный проект следует сохранить. Чтобы избежать дублирования имен проектов, каждый отдельный проект нужно сохранять ***в*** ***отдельной папке***. Поэтому перед сохранением создайте новую папку для этого проекта, например, *Пример1*. Затем используйте пункт меню **File/Save**. На экране появиться диалог **Save Project1 as**. В дереве файлов найдите созданную папку, задайте имя программы в окне *Имя файла*, например, *Example1* и нажмите на кнопку **Сохранить** (см. рисунок 5). После этого, изменится как имя файла проекта, так и имя на вкладке многооконного редактора программ.



**Рисунок *5*** *–* Внешний вид диалога сохранения проекта

4. На переименованной вкладке Example1 многооконного редактора программ вы­све­чи­ва­ет­ся заготовка консольного приложения. В этом окне набирают текст про­грам­мы. При необходимости можно в других окнах открыть файлы с другими про­граммами или данными с помощью пункта **File/Open**, но при этом проект может быть открыт только один.

Введите в открытое окно редактора программ выделенный текст:

*program Example1;*

*{$APPTYPE CONSOLE}*

*uses SysUtils;*

***Var A,B,C,D,E,X1,X2:Single;***

*Begin* ***WriteLn('Input A, B, C:');***

***ReadLn(A,B,C);***

***D:= Sqr(B)- 4\*A\*C;***

***if D>=0 then***

***begin E:=2\*A;***

***X1:= (-B+Sqrt(D))/E;X2:= (-B-Sqrt(D))/E;***

***WriteLn('X1=', X1:6:1, ' X2=', X2:6:1)***

***end***

***else WriteLn('No result');***

***ReadLn;***

*end.*

5. Для выполнения программы следует использовать кнопку быстрого вызова **Run** (зеленый треугольник на панели инструментов), пункт меню **Run/Run** иликлавишу **F9**.

Если при вводе программы вы допустили синтаксические ошибки, то компилятор выдаст на экране в окне вывода **Messages** (в нижней части окна среды) на вкладке **Build** соответствующие сообщения. При этом курсор будет установлен в тексте на место первой ошибки (см. рисунок 6).



Рисунок 6 – Вид окна при обнаружении ошибки компиляции

Для перехода на строку, содержащую следующую ошибку, необходимо дважды щелкнуть мышью по соответствующему сообщению на вкладке **Build**.

6. Если программа введена верно, то на экране появится окно Консоль, в которое будет выведен запрос на ввод чисел. Числа вводят через пробел или каждое число в своей строке, нажимая после ввода числа клавишу Enter. После ввода чисел в то же окно будет выведен результат (см. рисунок 7).

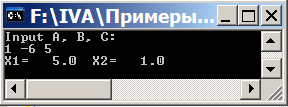
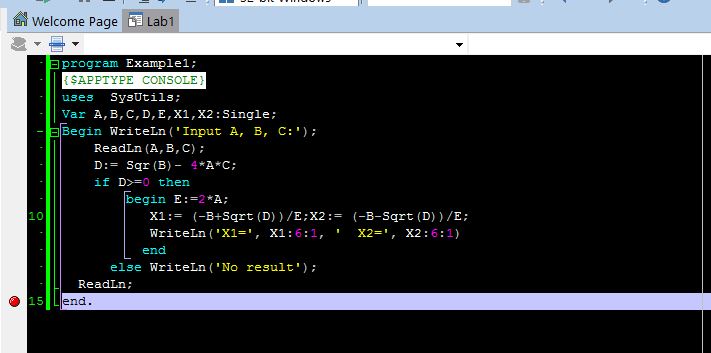


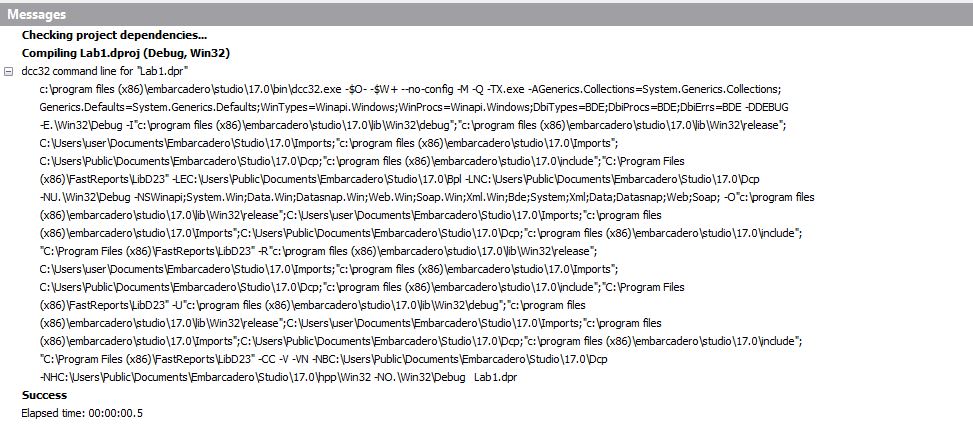
Рисунок 7 – Окно Консоли с выведенными результатами

При этом вид окна среды за окном консоли изменится, на нем появятся панели режима отладки программы: Call Stack – стек вызовов, Watch List – список отслеживаемых значений переменных, Local Variables – значения локальных переменных, Event Log – перечень событий. Особенности работы с этими окнами будут рассмотрены далее.

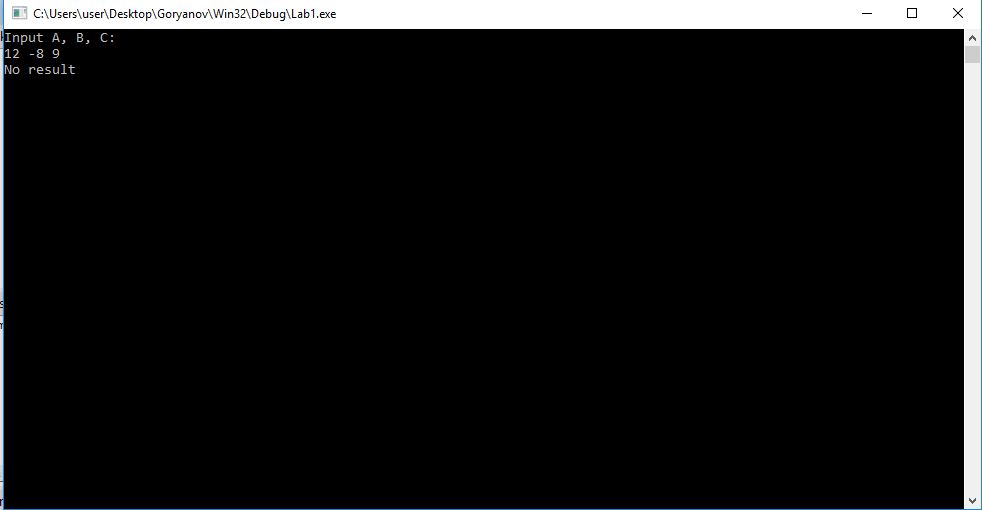
Код программы в компиляторе:



Сообщения:



Ввод данных в консоль и вывод полученных данных:



**Задание №2**

**Изучить диагностические сообщения Turbo Delphi.**

Поочередно внося ошибки в программу, фиксируйте сообщения об ошибках в специальной таблице, представленной ниже. Классифицируйте ошибку, расшифруйте сообщение системы и определите этап выполнения программы (компиляция, компоновка или выполнение), на котором была обнаружена данная ошибка.

**Таблица –** Диагностические сообщения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ошибка | Физический смысл ошибки | Проявление  ошибки | Расшифровка сообщения | Этап |
| 1. | *Var AB, C,*  *X1, X2, D, E;* | Вместо переменных A и B описана переменная AB | Получено сообщение: Undeclared identifier: 'A' | Не объявлена переменная A | Компиляция |
| 2. | Readln(A,B,C) | Синтаксическая ошибка | [dcc32 Error] Lab1.dpr(7): E2066 Missing operator or semicolon | Отсутствие «;» после команды | компиляция |
| 3. | E=2\*A; | Синтаксическая ошибка | [dcc32 Error] Lab1.dpr(9): E2029 ':=' expected but '=' found | Неправильное присвоение значения переменной | компиляция |
| 4. | *D:=spr(B)-4\*A\*C;* | Вызвана несуществующая функция | [dcc32 Error] Lab1.dpr(7): E2003 Undeclared identifier: 'Spr' | Функции SPR не существует | компиляция |
| 5. | Исходные  данные: 0 1 3 | Деление на ноль | Error: RunTime error | Деление на ноль | Выполнение |
| 6. | Исходные  данные: 1 1 3 | Ошибки нет | Результат не возможно найти | D<0 | Выполнение |
| 7. | { *E:=2\*A;* } | Пропущена строка вычислений | Некорректные вычисления | Некорректные вычисления из-за пропущенного действия | Выполнение |
| 8. | *Read(A,C);* | Не определено (не введено) значение B | Некорректные вычисления | Некорректные вычисления из-за отсутствия переменной | Выполнение |

**Задание 3**

**Изучить средства отладки программ в среде Delphi.**

Порядок работы:

1. Внесите в программу ошибку 7. Выполните пошаговую трассировку программы, отслеживая значения переменных A, B, C, D, E, x1, x2 в окне Watch. Какое значение имеет переменная E в момент вычисления корней уравнения? Почему?
2. Внесите в программу ошибку 6. Выполните пошаговую трассировку программы, отслеживая значение переменных A, B, C, D, E, x1, x2. Какое значение имеет переменная D? В какой момент обнаруживается ошибка?
3. Установите точку останова перед вычислением дискриминанта. Выполните программу до точки останова. Просмотрите значения переменных, подводя к ним курсор мыши.

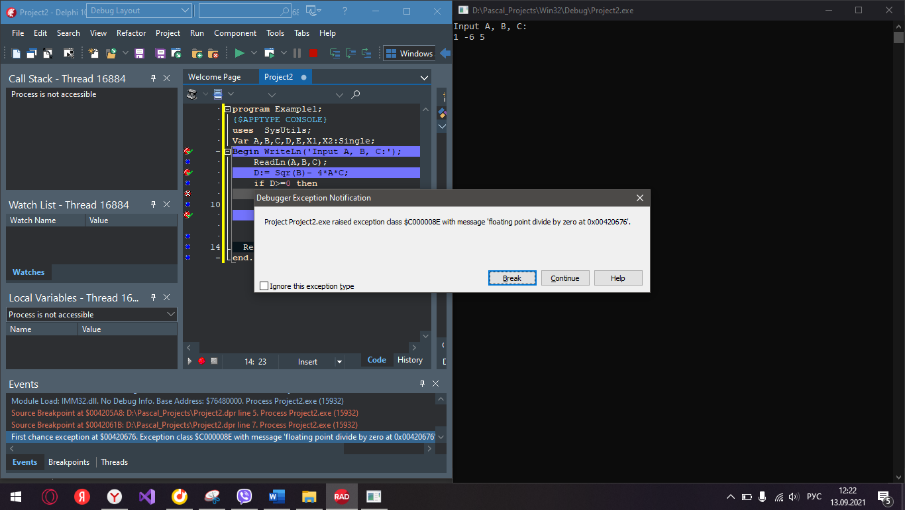
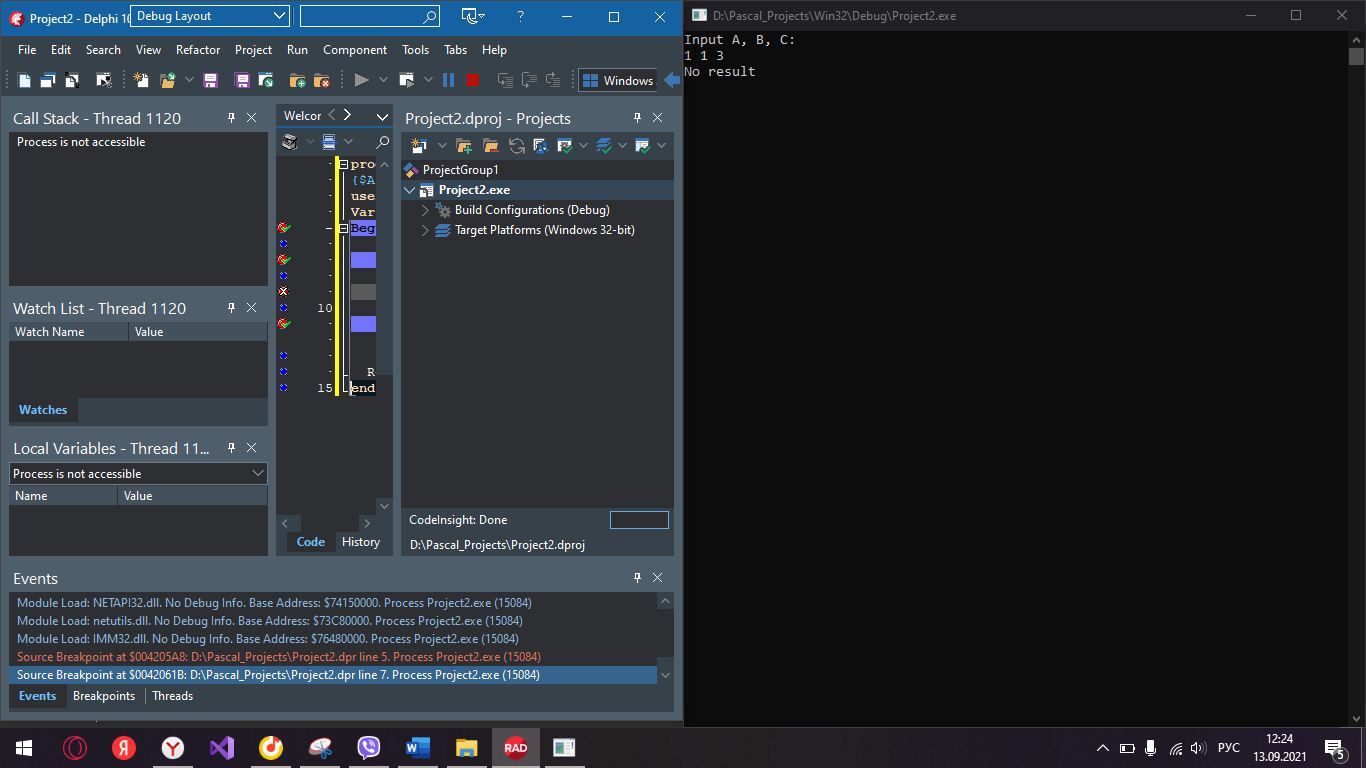
3. Ответы на вопросы и выводы занести в отчет.

1. При вводе переменных в А, В и С (1, -6, 5) переменная D после вычисления будет равна 4. Затем идет сравнение, если D больше либо равно нуля, то выполняется вычисление x1 и x2. Потом при делении (-B+Sqrt(D)) или (-B-Sqrt(D)) на E происходит ошибка, так как Е это пустая переменная, мы его только объявили и выражение { E:=2\*A; } закомментировали фигурными скобками и комментарии никакую роль при компиляции не играют.

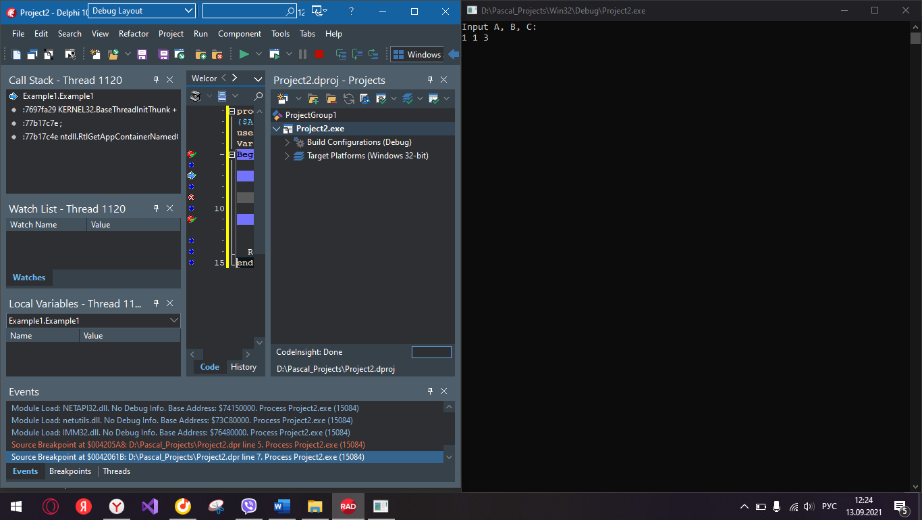
2. При вводе переменных в А, В и С (1, 1, 3) переменная D после вычисления будет равна -11. Потом идет сравнение, если D больше либо равно нуля, то выполняется блок программы нахождения корней, данном случае в переменной D отрицательное число, следовательно выполняется условие else.

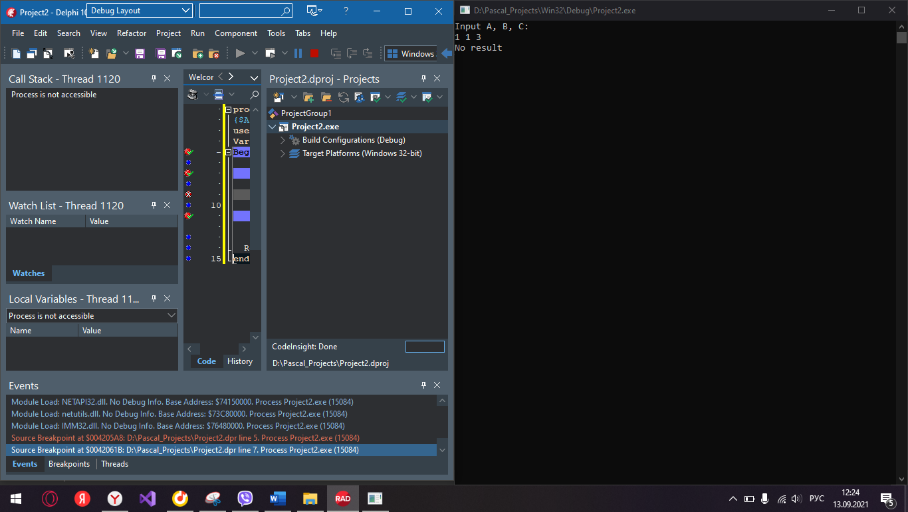
3. После ввода в Readln показывает введенные числа в переменные A, B, C.

1)



2)





**Часть 2**

Для алгоритма

Если (A<B)

A:=B;

Иначе

B:= A;

Конец

выполните следующие действия.

1. Создайте схему алгоритма в Microsoft Visio и сохраните её в формате Microsoft Visio и векторном графическом формате wmf.

2. Создайте схему алгоритма в OpenOffice Draw и сохраните её в формате OpenOffice Draw и векторном графическом формате wmf.

3. Создайте текстовый документ OpenOffice Writer и поместите в него рисунок схемы алгоритма в векторном формате wmf, полученный в Microsoft Visio.

4. Добавьте в этот же текстовый документ рисунок схемы алгоритма в векторном формате wmf, полученный в OpenOffice Draw.

5. Продемонстрируйте полученные результаты преподавателю (схему в Microsoft Visio, схему в OpenOffice Draw и текстовый документ)

