**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | программирование |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Разработка тестового сценария проекта | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Масевич Екатерина Александровна |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Авдеенков Владимир Александрович |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024 г.**

**Лабораторная работа №1. «Разработка тестового сценария проекта»**

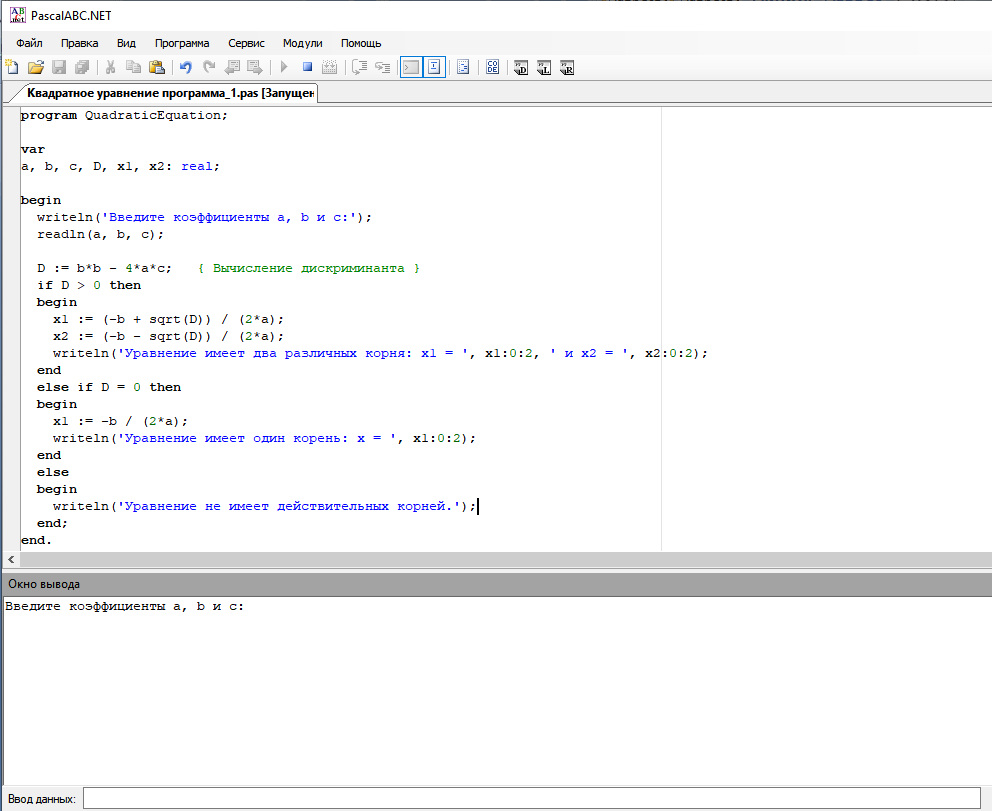
**Цель работы:**

Получить навыки разработки тестовых сценариев

Программа написана на языке программирования Паскаль

**Задания**

1. **Задание 1.** Написать (язык – любой) программу решения квадратного уравнения ах2 + bх + с = 0 (задаваемые с клавиатуры коэффициенты – a, b и c);

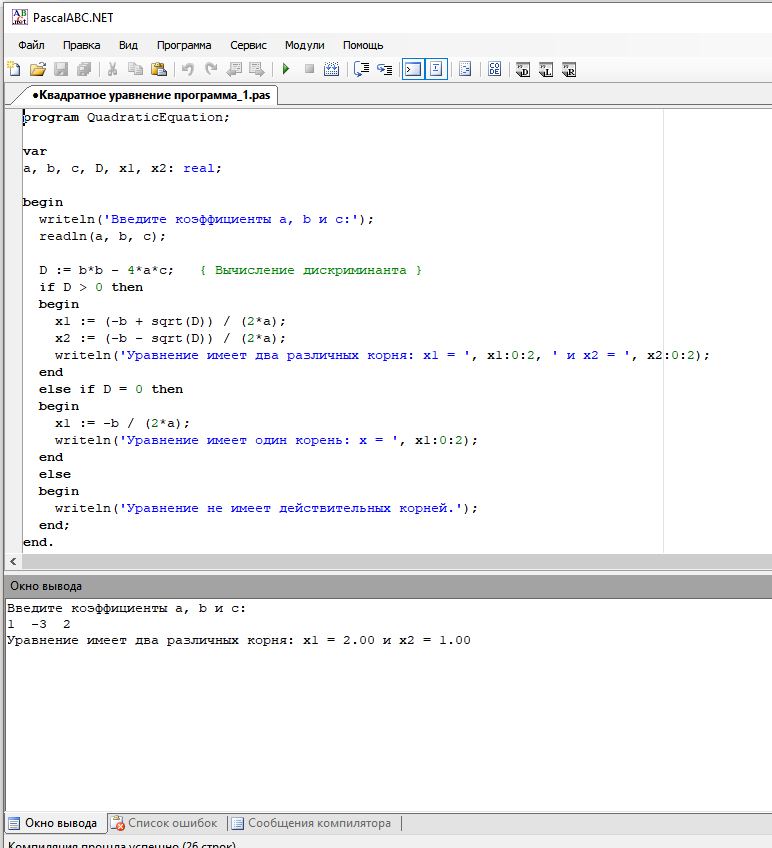


Как работает программа (скриншот)

**Задание 2.** Найти минимальный набор тестов для программы нахождения корней квадратного уравнения ах2 + bх + с = 0 (корни вещественные или комплексные, один/два/бесконечно, один/оба нулевых корня и т. д.);

**1. Тест с двумя различными корнями** входные данные a= 1, b = -3, c= 2

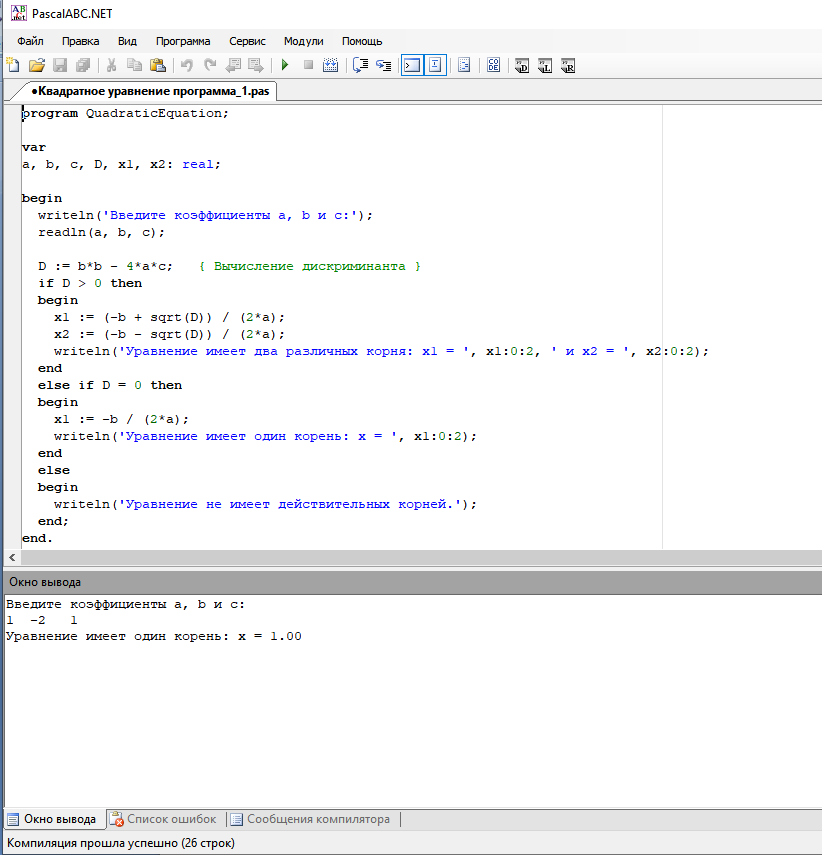
Ожидаемый результат x1=2, x2=1.



**2. Тест с одним корнем (дискриминант равен нулю):**

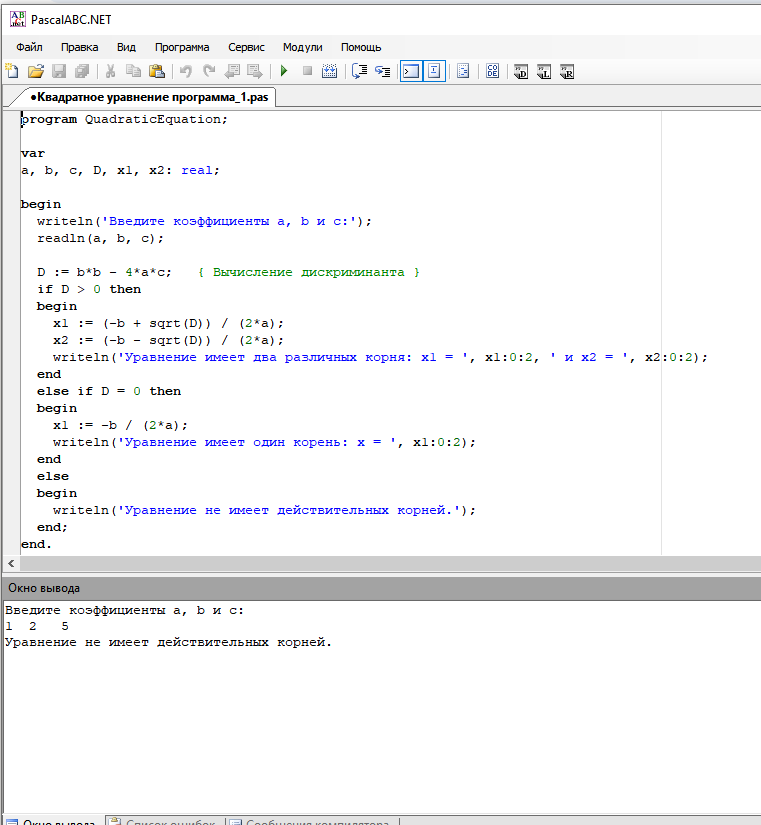
Входные данные: a=1, b = -2, c = 1.

Ожидаемый результат : x = 1.



**3. Тест без действительных корней (дискриминант меньше нуля):** Входные данные: a= 1, b =2, c= 5

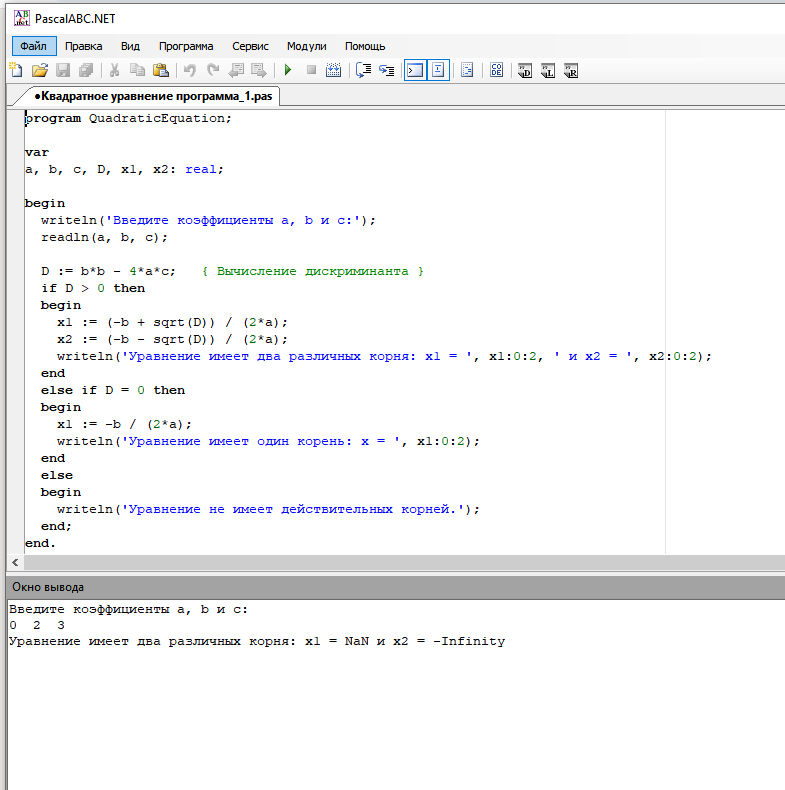
Ожидаемый результат: сообщение о том, что корней нет.



4. **Тест с нулевым коэффициентом а (линейное уравнение):**

Входные данные: a= 0, b= 2, c= 3

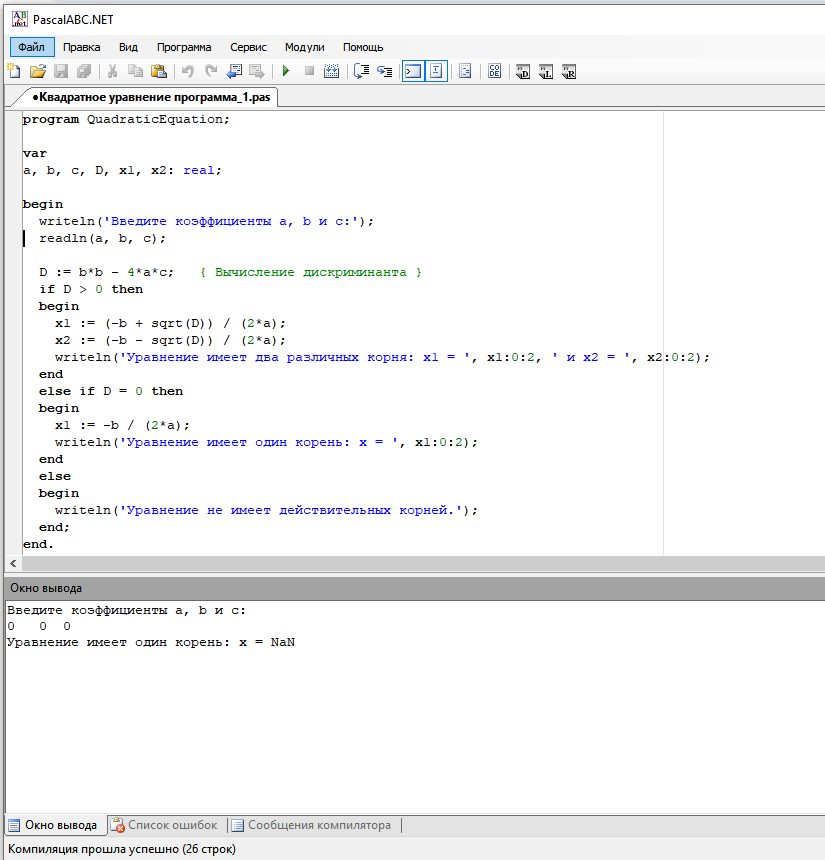
Ожидаемый результат: х= -1.5 (линейное уравнение).



5. **Тест с нулевыми коэффициентами (все коэффициенты равны нулю):**

Входные данные: a = 0, b= 0, c= 0.

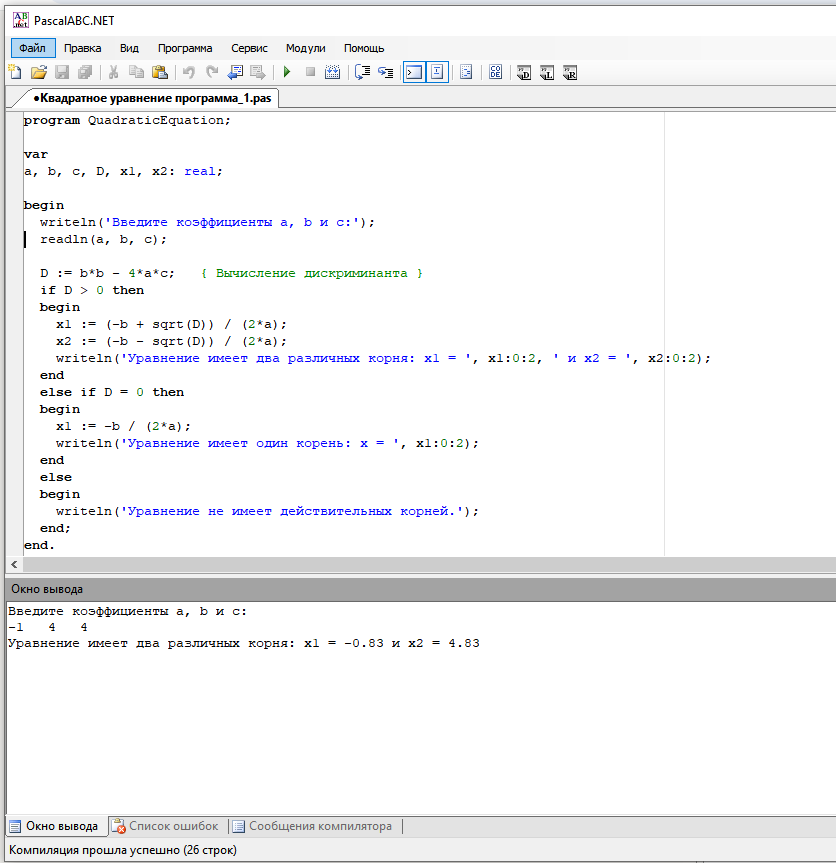
Ожидаемый результат: неопределенность или сообщение о том, что уравнение не имеет смысла.



6. **Тест с отрицательными коэффициентами:**

Входные данные: a=-1, b= 4, c=4.

Ожидаемый результат: x1= -2 (дискриминант равен нулю).



Заключение

Эти тесты помогут убедиться, что программа корректно обрабатывает различные случаи, включая наличие и отсутствие корней, а также особые ситуации с нулевыми коэффициентами.

**Теоретические вопросы**

1. **Оценка стоимости и причины ошибок в программном обеспечении.**

**Ответ- ошибки в программном обеспечении могут быть связаны с человеческим фактором или со сложностью системы.**

1. **Виды и методы тестирования.**

**Методы тестирования**  
  
1. Черный ящик:  
- Тестирование функциональности системы, не учитывая внутреннюю структуру или код.  
  
2. Белый ящик:  
- Тестирование с учетом внутренней структуры системы; включает анализ кода, а также тестирование логики, условия и ветвления.  
  
3. Серый ящик:  
- Комбинация черного и белого ящика, где тестировщик имеет частичное знание о внутренней реализации системы.  
  
4. Статическое тестирование:  
- Анализ кода и документации без выполнения программы (например, рецензии, статический анализ кода)

**Виды тестирования программного обеспечения**  
1. Функциональное тестирование:  
2. Нефункциональное тестирование:  
3. Мануальное тестирование:  
4. Автоматизированное тестирование:  
5. Регрессионное тестирование:  
  
6. Модульное тестирование:.  
  
7. Интеграционное тестирование:.  
  
8. Системное тестирование:  
  
9. Приемочное тестирование:.  
  
10. Бета-тестирование:

1. **Понятие теста.**

Тестирование информационной системы — это процесс оценки и проверки функциональности, производительности, безопасности и других характеристик информационной системы с целью обеспечения ее соответствия специфическим требованиям и ожиданиям пользователей. Этот процесс играет ключевую роль в разработке и реализации любой информационной системы, поскольку помогает выявить ошибки, недостатки и уязвимости прежде, чем система будет введена в эксплуатацию.

1. **Требования к разработке тестовых сценариев.**

1. Ясность и понятность  
- Тестовые сценарии должны быть написаны в ясном и лаконичном языке, с четким описанием. Это позволяет тестировщикам легко понимать, что именно необходимо проверить и как это сделать.  
  
2. Полнота  
- Сценарии должны охватывать все аспекты функциональности системы, включая положительные и отрицательные тесты. Это также включает проверки на граничных значениях и тесты производительности.  
  
3.Отслеживаемость  
- Каждый тестовый сценарий должен быть связан с конкретными требованиями или задачами из проектной документации. Это позволяет убедиться, что все требования были протестированы.  
4. Выполнимость  
- Все сценарии должны быть выполнимыми, и для их реализации должны быть доступны необходимые ресурсы, данные и инструменты. Сценарии должны быть адаптированы к имеющимся условиям тестирования.  
  
5. Измеримость  
- Результаты выполнения тестов должны быть легко измеримыми. Это подразумевает наличие четких критериев, по которым будет приниматься решение о том, пройден ли тест или нет.  
  
6. Редактируемость  
- Тестовые сценарии должны быть легко редактируемыми для учета изменений в требованиях, особенностях проекта и фиксирования найденных дефектов. Это позволяет поддерживать актуальность тестовой документации.  
  
7. Структурированность  
- Сценарии должны иметь четкую структуру, что облегчает их организацию и поиск. Например, использование шаблонов для тестовых сценариев может значительно ускорить процесс их написания.  
  
8. Приемлемость  
- Все сценарии должны отражать реальные условия, в которых будет работать система. Это помогает создать правдоподобные сценарии тестирования, которые обеспечивают жизнеспособность системы в реальных условиях.  
  
9. Повторяемость  
- Сценарии должны быть такими, чтобы их результаты могли быть реплицированы при повторном тестировании. Это важно для проверки исправлений и изменений в коде.  
  
 10. Согласованность  
- Тестовые сценарии должны быть согласованы между собой и следовать единому стилю написания. Это позволяет тестировщикам легче работать с ними и делать выводы.  
  
11. Документирование  
- Каждый тестовый сценарий должен содержать всю необходимую информацию, такую как описание, необходимые предварительные условия, ожидаемые результаты, кто выполняет тест и т. д.

**5. Правила разработки тестовых сценариев.**

Разработка тестового сценария происходит в пять этапов:  
  
**1. Понимание требований.** Нужно чётко и полно собрать требования к программе: изучить спецификацию, обсудить результаты работы ПО, проанализировать пользовательские сценарии.  
  
**2. Определение цели.** Нужно определить, что именно будет проверяться этим тестовым сценарием. Целью может быть как проверка функциональности, так и поиск конкретных ошибок.  
  
**3. Описание шагов.** Написание непосредственно сценария — последовательности шагов, которые нужно выполнить, чтобы протестировать конкретный аспект программного продукта. Шаги должны быть ясными, понятными и охватывать все необходимые действия для достижения цели сценария.  
  
**4. Описание ожидаемых результатов.** Для каждого шага необходимо прописать конкретные результаты, к которым он должен привести.   
  
**5. Документирование.** Формирование из последовательности действий документа формата, принятого в компании.