**Особенности проведения лабораторных работ**

**по программированию**

Лабораторная работа по программирования является не просто письменным проектом (отчетом о проделанной практической работе), а представляет собой учебно-практический комплекс индивидуальных заданий. Основная задача, решаемая при выполнении лабораторных работ, заключается в углублении знаний студентов в рамках выбранного профиля или конкретной дисциплины, овладение реальными навыками и грамотное оформление достижений в форме полноценного программного проекта.

Программный проект определяется как полная процедура разработки программного обеспечения от сбора требований до тестирования и обслуживания. Другими словами программный проект представляет собой последовательность действий по разработке и реализации программного продукта, характерной особенностью которого является наличие следующих характеристик (атрибутов):

*Цель* – для чего разрабатывается данный программный продукт.

*Важность* – почему достижение объявленной цели так важно и необходимо.

*Уникальность* – почему нужна новая разработка, а нельзя использовать какой-либо уже известный программный проект.

*Качество*, которым должен обладать конечный продукт.

*Время* – программный проект должен быть выполнен к указанному сроку.

*Ресурсы*  – выполняемый проект должен уложиться в заданный бюджет при использовании ограниченных средств.

При выполнении отдельной лабораторной работы все перечисленные характеристики в обязательном порядке присутствуют и обеспечивают освоение тех или иных компетенций, предусмотренных образовательным стандартом. Общий принцип выполнения лабораторных работ можно определить фразой «ВСЕ, ЧТО ДЕЛАЕТ ПРОГРАММИСТ, ДОЛЖНО БЫТЬ ПОЛЕЗНЫМ И КРАСИВЫМ». Конкретные требования по выполнению, оформлению и сдачи лабораторных работ могут незначительно отличаться у разных преподавателей. Однако во всех случаях действует одно неукоснительное правило, в соответствии с которым в сдаче зачета (или экзамена) допускается студент, выполнивший и сдавший в установленный срок все предусмотренные учебным планом работы.

**Отчет о проделанной работе и его содержание**

После проведения лабораторной работы каждый студент оформляет отчет о проделанной работе. В отчет должны быть включены следующие пункты:

- титульный лист;

- цель работы;

- персональное задание (задание, как правило, определяется вариантом, номер которого соответствует порядковому номеру студента в списке группы);

- анализ задания;

- краткие теоретические сведения (математическая постановка задачи), если это требуется при выполнении задания;

- описание математических методов, используемых студентов при реализации программного проекта;

- алгоритм/алгоритмы основной программы (с проведением детализации укрупненных алгоритмов) и всех подпрограмм, используемых при выполнении проекта;

- описание всех исходных переменны, подлежащих вводу, и переменных, которые являются результатами выполнения программы;

- для каждой подпрограммы осуществляет обоснование использования (выполненное в процессе анализа задания) и подробное описание всех входных и выходных формальных параметров (фактические параметры описываются в основной программе);

- программный код (с обязательными начальными и промежуточными комментариями) для всех частей программного проекта;

- результаты анализа правильности работы программного кода (в том числе результаты расчетов контрольных вариантов);

- результаты расчетов в соответствии с заданием и их анализ;

- выводы о проделанной работе.

**Оформление отчета**

Оформление отчета по лабораторной работе (курсовой работе, курсовому проекту, выпускной квалификационной работе и пр.) является одной из ее важных деталей. Существуют основные правила оформления отдельных элементов отчета, таких как текст отчета, формулы, рисунки, схемы (блок-схемы) алгоритмов и т.д. Работа оформляется на одной стороне листа бумаги формата А4 через 1,5 интервала с числом строк на странице не более 40. Оборотная сторона страниц оставляется для пояснений преподавателей или возможных исправлений самого студента. Текст необходимо писать, соблюдая размеры полей:

левое поле – не менее 30 мм;

правое поле – не менее 10 мм;

верхнее поле - не менее 15 мм;

нижнее поле 20 мм;

**Рекомендуемый размер шрифеа составляет 14 пикселей, шрифт – Times New Roman.** Для каждой лабораторной работы оформляется отдельный титульный лист и все остальные части отчета.

Страницы нумеруются с середины нижнего поля без черточек. Не допускается нумерация страниц с литерами (например, 15а), пропуск или повтор номеров. На титульном листе или страницах, полностью занятых рисунками, номера страниц не ставят, но учитывают.

При оформлении отчета необходимо соблюдать правила правописания Русского языка. Существует ряд требований, по которым нельзя выполнять следующие действия:

- разбивать переносом аббревиатуры;

- отрывать фамилии от инициалов и инициалы друг от друга;

- размещать в разных строках числа и их наименования (например, 32 Кбайт, 34 микросекунды, 8 Мбит/с и т.д.);

- оставлять в конце строки открывающиеся кавычки или открывающуюся скобку.

Для обозначения диапазона значений употребляют один из следующих способов: многоточие (i = 1..n), предлоги от 15 до 20. В отчете следует придерживаться принципа единообразия. При обозначении изменения параметра цикла х от начального значения x0 до конечного значения xn с постоянным шагомhx рекомендуется использовать нотацию x=x0(hx)xn , например, z=0,05(0,1)4,35 или к=0(1)m.

Стандартной формой написания дат является следующая: 13.09.23 г. Допускается использование других цифровых и словесно-цифровых форм, например, 13.09.2023 г., 13 сентября 2023 г.

В тексте отчета без пояснения используются только общепринятые сокращения, такие как: т.е. (то есть); т.п. (тому подобное); т.д. (так далее), т.к. (так как), и др. (и другие), см. (смотрите), и пр. (и прочее), ЭВМ (электронная вычислительная машина), ПК (персональный компьютер), ПО (программное обеспечение), п/п (подпрограмма).

Не допускается использование сленга при написании отчета (например, программный код весит 34 кило, рандомные числа и др.). В тексте отчета необходимо следить за правильным использованием понятий и определений. Очень часто допускается не правильно использование терминов, например, *размер масс*ива (это количество элементов в массиве) и *размерность массива* (размерность массива определяется количеством входящих в него индексов). Сокращения рис. и табл. в тексте отчета необходимо использовать совместно с указанием нужного номера. Без указания номера слова «рисунок» и «таблица» принято писать полностью. Полностью слово «Таблица 1» (или с иным номером) пишется также в заголовках таблиц (перед таблицей прижато к правому краю). Под рисунками всегда пишут Рис. 1 (или с иным номером) с выравниванием посередине. Первая ссылка на рисунок или таблицу должна присутствовать по расположения этого элемента в тексте.

При изображении схем алгоритмов (часто называют блок-схем алгоритмов) требуется неукоснительно придерживаться требований и рекомендациям ГОСТ (государственные стандарты) и ЕСПД (Единая система программной документации).

Название идентификаторов (название переменных) рекомендуется выбирать в соответствии в обозначениями, которые используются в математической постановке задачи или в используемом математическом методе. Описание математического метода предшествует разработке алгоритма/алгоритмов.

Далее приведен пример оформления отчета по проделанной лабораторной работе.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Рязанский государственный радиотехнический университет имени В. Ф. Уткина»  
Кафедра вычислительной и прикладной математики

**Отчет**

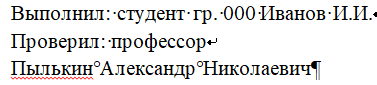
**о лабораторной работе № 5**

**по дисциплине**

«Алгоритмические языки и программирование»

**на тему**

“Процедуры и функции”

****

Рязань 2023

**Дата выполнения лабораторной работы: 13.09.2023**

**Задание (**Вариант №0)

Составьте подпрограмму вычисления двух корней квадратного уравнения (корни различны, корни равны, нет корней)

**Анализ задания**

Входные данные:

а (float) – первый коэффициент квадратного уравнения

b (float) – второй коэффициент квадратного уравнения

c (float) – третий коэффициент квадратного уравнения

Выходные данные:

answer (tuple - кортеж) – корни квадратного уравнения или сообщение «Нет корней». Проверка правильности нахождения корней осуществляется методом подстановки их значений в исходное уравнение.

Используемые математические методы:

Определение корней квадратного уравнения осуществляется с использованием формулы нахождения корней квадратного уравнения через нахождение дискриминанта. Значение дискриминанта необходимо для определения факта существования корней.

Основные этапы решения, реализуемые в программе:

1. Ввод исходных данных
2. Вызов функции решения уравнения с данными коэффициентами (используется функция solve)
3. Вызов функции нахождения дискриминанта (используется функция findDiscriminant)
4. Вызов функции нахождение корней (используется функция findRoots).
5. Вывод результатов решения задачи

**Схемы алгоритмов программы и функций**

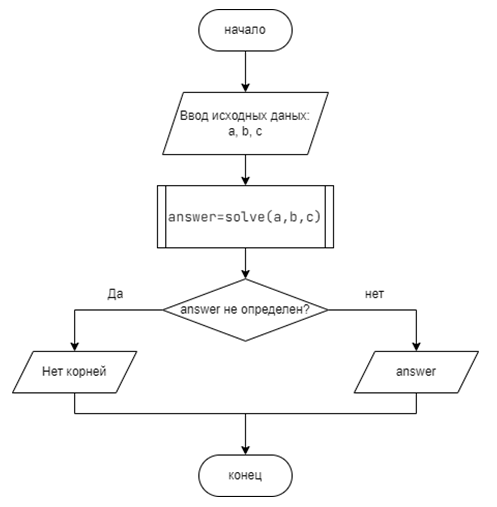


Рис. 1. Укрупненная схема алгоритма решения задачи

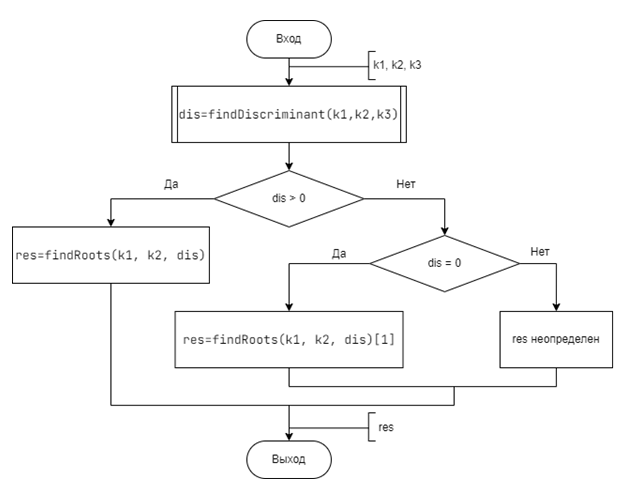


Рис. 2. Схема алгоритма функции solve нахождения корней квадратного уравнения

**Функция solve (решить):**

Формальные параметры:  
 входные – k1, k2, k3  
 выходные – res

Фактические параметры (вызываются в основной программе):  
 входные – a, b, c  
 выходные – answer

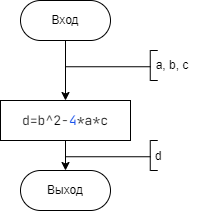


Рис. 3. Схема алгоритма функции findDiscriminant нахождения дискриминанта

**Функция findDiscriminant (найти дискриминант):**

Формальные параметры:  
 входные – a, b, c  
 выходные – d

Фактические параметры (вызываются в функции solve):  
 входные – k1, k2, k3  
 выходные – dis

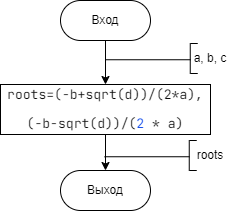


Рис. 2.Схема алгоритма функции findRoots нахождения корней квадратного уравнения

**Функция findRoots (найти корни):**

Формальные параметры:  
 входные – a, b, c  
 выходные – roots

Фактические параметры (вызываются в функции solve):  
 входные – k1, k2, k3  
 выходные – res

**Листинг программного кода:**

from math import sqrt  
  
  
*# Функция поиска дискриминанта  
# Формальные параметры:  
# а - первый коэффициент квадратного уравнения  
# b - второй коэффициент квадратного уравнения  
# c - третий коэффициент квадратного уравнения  
# Возвращаемое значение:  
# d - дискриминант*def findDiscriminant(a, b, c):  
 d = b \*\* 2 - 4 \* a \* c  
 return d  
  
  
*# Функция расчета корней квадратного уравнения  
# Формальные параметры:  
# а - первый коэффициент квадратного уравнения  
# b - второй коэффициент квадратного уравнения  
# d - дискрименант  
# Возвращаемое значение:  
# roots - кортеж из двух корней квадратного уравнения*def findRoots(a, b, d):  
 roots = round((-b + sqrt(d)) / (2 \* a),2), round((-b - sqrt(d)) / (2 \* a),2)  
 return roots  
  
  
*# Функция решения квадратного уравнения  
# Формальные параметры:  
# k1 - первый коэффициент квадратного уравнения  
# k2 - второй коэффициент квадратного уравнения  
# k3 - третий коэффициент квадратного уравнения  
# Возвращаемое значение:  
# res - корни квадратного уравнения*def solve(k1, k2, k3):  
 dis = findDiscriminant(k1, k2, k3)  
 if dis > 0:  
 res = findRoots(k1, k2, dis)  
 elif dis == 0:  
 res = findRoots(k1, k2, dis)[1]  
 else:  
 res = None  
 return res  
  
*# Цель: найти корни квадратного уравнения  
# Дата написания: 01.01.1001  
# Программист: Петров И.И.*print("Введите коэффициенты для уравнения")  
print("ax^2 + bx + c = 0:")  
a = float(input("a = "))  
b = float(input("b = "))  
c = float(input("c = "))  
  
print(f"Уравнение: {a} \*x^2 + {b} \*x + {c} = 0")  
  
answer = solve(a, b, c)  
if answer is None:  
 print("Нет корней")  
else:  
 print("Решение: ", answer)

**Результаты работы программы**

Уравнение 1:

Уравнение 2:

Уравнение 3:

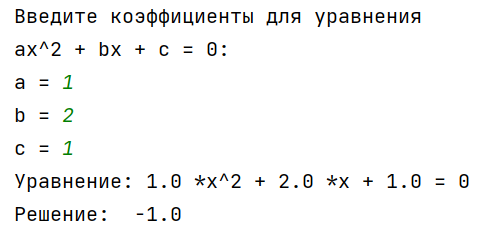


Рис. 3. Результат работы программы для уравнения 1

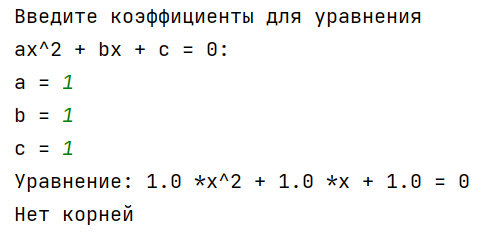


Рис. 4. Результат работы программы для уравнения 2

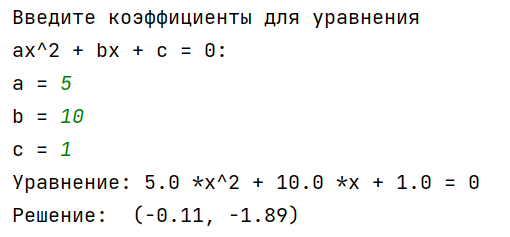


Рис. 5. Результат работы программы для уравнения 3

**Проверка полученных значений:**



Рис. 6. Решение уравнения 1 в программе Photomath



Рис. 7. Решение уравнения 2 в программе Photomath

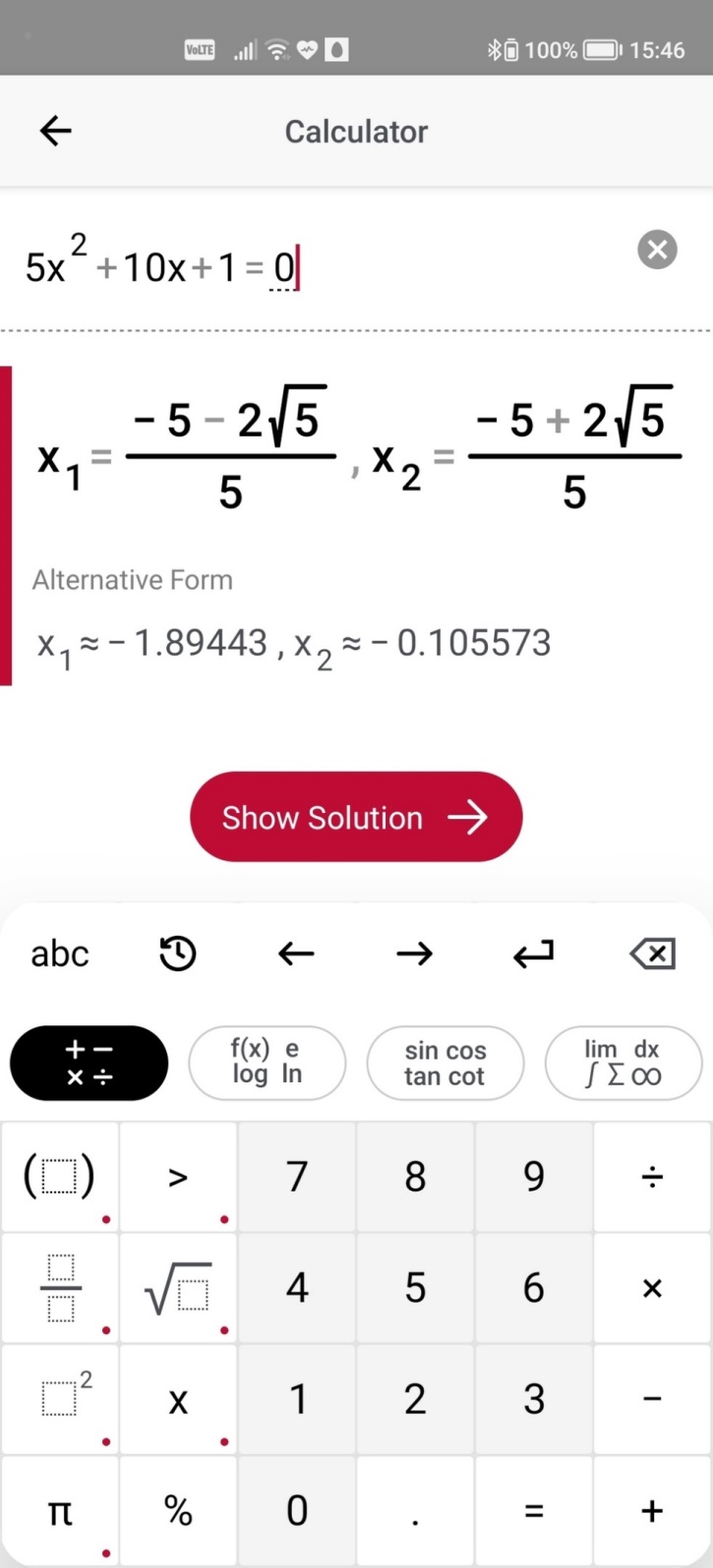


Рис. 8. Решение уравнения 3 в программе Photomath