

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА АЛЕКСАНДРА I»
(ФГБОУ ВО ПГУПС)

**СБОРНИК ЗАДАНИЙ ПО ИНФОРМАТИКЕ.
ТЕКСТОВЫЙ ПРОЦЕССОР И ОСНОВНЫЕ АЛГОРИТМИЧЕСКИЕ
СТРУКТУРЫ**

Учебное пособие

Для студентов заочного отделения

САНКТ-ПЕТЕРБУРГ

2025

УДК 004
ББК 32.81
К58

Рецензенты:

кандидат технических наук, главный специалист
по защите информации ПАО «Банк Санкт-Петербург»
И. А. Беляков;

кандидат технических наук, доцент кафедры «Подъемно-транспортные,
строительные, дорожные средства и оборудование» *А.М. Перепеченов*

**Дергачев А.И., Дергачев С.А., Кожевников А. И., Петрова О.Н.,
Степанская О.А., Карпунин П.С.**

К58

Сборник заданий по информатике. Текстовый процессор и основные
алгоритмические структуры: учебное пособие / Дергачев А.И., Дергачев
С.А., А.И. Кожевников, О.В. Петрова., Степанская О.А., Карпунин П.С. –
СПб.: ФГБОУ ВО ПГУПС, 2025. – 113 с.

ISBN 978-5-7641-1244-2

В первой части учебного пособия представлены теоретические основы
алгоритмизации задач и основы программирования на языках высокого уровня Visual
Basic for Applications и Python.

Во второй части учебного пособия в приложении представлены задания на
лабораторные работы по освоению текстового процессора и трех основных
алгоритмических структур: Следование, Развилка и Цикл. Указаны цели и порядок
выполнения работ.

В приложениях приведены примеры оформления отчетов.

Сборник предназначен для студентов 1 курса всех специальностей и
направлений подготовки.

УДК 004
ББК 32.81

© Дергачев А.И., Дергачев С.А., А. И. Кожевников, О. В. Петрова., Степанская О.А.,
Карпунин П.С. 2025

ISBN 978-5-7541-1243-5

© ФГБОУ ВО ПГУПС, 2025

Введение

Целью изучения дисциплины «Информатика» является ознакомление обучающихся с основами современных информационных технологий, тенденциями их развития и принципами построения информационных моделей.

Для достижения поставленной цели решаются следующие задачи:

- формирование у студентов представления об информатике как фундаментальной научной дисциплине, изучающей вопросы проектирования и защиты современных автоматизированных информационных систем;
- получение студентами знаний о значении информации в развитии современного информационного общества, об основных положениях теории информации и характеристиках информационных процессов;
- обучение студентов основам алгоритмизации и программирования как фундаментальной теоретической базе, используемой при разработке информационных технологий;
- формирование у студентов представления о современных информационных технологиях, автоматизированных информационных системах и сетях передачи данных, средствах, методах и механизмах их защиты.

В учебном пособии представлено задания на лабораторные работы: текстовый процессор и основные алгоритмические структуры – Следование, Развилка, Цикл.

Для каждого задания даны пояснения и рекомендации для их выполнения. В приложениях приведены примеры выполнения и оформления работ.

Оглавление

Введение	3
1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АЛГОРИТМИЗАЦИИ ЗАДАЧ.....	6
1.1 Основная терминология	6
2 Основы алгоритмизации задач.....	7
2.1 Основные структуры, используемые при конструировании схемы алгоритма.....	7
2.1.1 Основные требования Госстандарта к оформлению схемы	7
2.2 Конструирование основных структур	8
2.3 Структурный подход к программированию.....	10
2.4 Метод нисходящего проектирования.....	12
2.5 Структурное программирование	12
2.6 Метод тестирования алгоритма	13
3 ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ Visual Basic for Applications	14
3.1 Основы языка программирования Visual Basic for Applications	15
3.2 Основные понятия	15
3.3 Алфавит языка	15
3.4 Типы данных	17
3.5 Запись формульных выражений.....	18
3.6 Операции отношения.....	19
Логические операции	19
3.7 Оператор присваивания.....	20
3.8 Встроенные функции VBA	20
3.8.1 Встроенные функции.....	21
3.9 Организация ввода данных	22
3.9.1 Функция InputBox()	22
3.9.2 Ввод данных с помощью текстового поля	23
3.9.3 Организация вывода данных	23
3.9.4 Оператор MsgBox	24
3.9.5 Format(параметр, "прототип")	24
3.9.6 Метод DEBUG.PRINT	24
3.9.7 Вывод данных с помощью текстового поля	25
3.9.8 Операторы End Sub и Rem	25
3.10 Оператор условного перехода IF	25
3.11 Операторы цикла.....	28
3.11.1 Общая характеристика операторов цикла	28
3.11.2 Операторы ЦИКЛА С ПРЕДУСЛОВИЕМ FOR.. ..NEXT и DO WHILE.. ..LOOP	29
3.11.3 Операторы ЦИКЛА С ПОСТУСЛОВИЕМ DO.. ..LOOP WHILE	31
4 ПОДГОТОВКА ЗАДАЧ К РЕШЕНИЮ НА КОМПЬЮТЕРЕ	31
4.1 Основные требования к оформлению отчётного документа при подготовке задач к решению на компьютере	31
5 Примеры реализации теоретических положений при подготовке задач к решению на компьютере.....	33
5.1 Пример 1. Подготовка к решению задачи со структурой СЛЕДОВАНИЕ	33
5.2 Пример 2. Подготовка к решению задачи со структурой РАЗВЛКА	36
5.3 Пример 3. Подготовка к решению задачи со структурой ЦИКЛ.....	43
6 ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ PYTHON	49
7 Основные понятия.....	49
7.1 Ключевые слова	49
7.2 Литералы.....	49

7.3	Переменные	49
7.4	Алфавит языка Python	50
7.5	Синтаксис и семантика.....	50
8	Типы данных в Python.....	50
9	Объявление данных.....	51
9.1	Объявление переменных.....	51
9.2	Изменения типа данных.....	51
10	Операции.....	51
10.1	Арифметические операции	51
10.2	Операции сравнения.....	52
10.3	Логические операции.....	52
11	Ввод и вывод данных.....	54
11.1	Ввод данных	54
11.2	Вывод данных.....	54
12	Условные операторы.....	55
12.1	if-elif-else	55
13	Циклы	56
13.1	Цикл For	56
13.2	Цикл While	56
14	Рекурсия	57
15	Работа со строками и срез.....	57
16	Массив	58
17	Множество set()	58
18	Библиотека math (sin, cos...).....	59
19	Чтение текстового и табличного файла	59
19.1	Чтение файла open('')	59
19.2	Разбиение на строки readline().....	59
19.3	Разбиение строки на массив Split()	60
20	Список используемой литературы.....	61
	<i>ПРИЛОЖЕНИЕ 1</i>	62
	<i>Лабораторная работа № 1</i>	62
	Текстовый процессор	62
	<i>ПРИЛОЖЕНИЕ 2</i>	65
	<i>Лабораторная работа № 2</i>	65
	Структура Следование	65
	<i>ПРИЛОЖЕНИЕ 3</i>	73
	<i>Лабораторная работа № 3</i>	73
	Структура Развилка	73
	<i>ПРИЛОЖЕНИЕ 4</i>	79
	<i>Лабораторная работа № 4</i>	79
	Структура Цикл.....	79
	<i>Контрольные вопросы к лабораторной работе № 2</i>	89
	<i>Контрольные вопросы к лабораторной работе № 3</i>	90
	<i>Контрольные вопросы к лабораторной работе № 4</i>	91
	Приложение А.....	92
	Приложение Б	99
	Приложение В	103
	Приложение Г	108

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АЛГОРИТМИЗАЦИИ ЗАДАЧ

1.1 Основная терминология

При подготовке задачи к решению на ПК является программа для компьютера, которой ГОСТ 19781–90 даёт следующее определение:

***программа** – данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определённого алгоритма* [1, с. 160].

Для чёткого уяснения этого определения, вполне очевидно, необходимо знать и понимать понятия и определения, имеющихся в нём терминов: **данные, информация и алгоритм**, а также ряд других терминов, связанных с ними непосредственно.

Рассмотрим обоснование и определение наиболее важных терминов.

***Данные** – зарегистрированные сигналы материального мира, которые используются или могут быть использованы человеком непосредственно или через промежуточные звенья.*

***Информация** – это продукт взаимодействия данных и адекватных им методов* [2].

Алгоритм — это понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение цели.

Основные свойства алгоритма: **массовость, определённость, дискретность и результативность.**

Программирование – научная и практическая деятельность по созданию программ [1].

Непосредственно к программированию относятся следующие термины и их определения.

Алгоритмический язык – искусственный язык, предназначенный для выражения алгоритмов [1].

Языки программирования – это тщательно и изобретательно составленные последовательности слов, букв, чисел и мнемонических сокращений, которыми люди пользуются для общения с компьютерами.

Система программирования – система, образуемая языком программирования, компиляторами или интерпретаторами программ, соответствующей документацией, а также вспомогательными средствами для подготовки программ к форме, пригодной для выполнения [1].

Программное обеспечение – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ [1].

2 Основы алгоритмизации задач

Алгоритмизация функциональных и вычислительных задач заключается в составлении алгоритма их решения. Наиболее удобными для человека формами представления алгоритма решения задачи являются: во-первых, схема алгоритма, изображаемая в виде графических символов и линий логических связей (математическое обеспечение) и, во-вторых, текст программы, записанный на языке программирования высокого уровня с соблюдением обязательного условия: быть удобным для чтения человеком (программное обеспечение). При структурном подходе к программированию обе формы изображения алгоритма конструируются из трёх основных структур.

2.1 Основные структуры, используемые при конструировании схемы алгоритма

2.1.1 ОСНОВНЫЕ ТРЕБОВАНИЯ ГОССТАНДАРТА К ОФОРМЛЕНИЮ СХЕМЫ

В соответствии с ГОСТ 19.701–90 схемы алгоритмов состоят из:

а) имеющих заданное значение условных графических обозначений (символов); б) краткого пояснительного текста; 3) соединяющих линий.

Символ предназначен для графической идентификации функции, которую он отображает, независимо от текста внутри этого символа.

Госстандарт устанавливает следующие основные правила [5]:

1) при применении символов:

- символы в схеме должны быть расположены равномерно;
- не должны изменяться углы и другие параметры, влияющие на соответствующую форму символов. Символы должны быть, по возможности, одного размера;
- если объём текста, помещаемого внутри символа, превышает его размеры, следует использовать символ комментария;

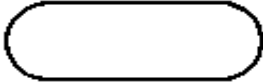
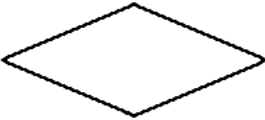
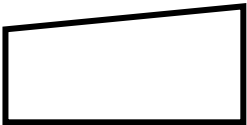
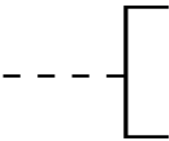
2) при выполнении соединений [5]:

- направление линии потока данных слева направо и сверху вниз считается стандартным (не требуется указывать стрелку);
- в случаях нестандартного направления линии потока данных или при необходимо внести большую ясность в схему на линиях используются стрелки;
- в схемах следует избегать пересечения линий. Изменения направления в точках пересечения не допускаются;
- если две или более линии объединяются в одну линию, место объединения должно быть смещено;

- линии в схемах должны подходить к символу либо слева, либо сверху, а исходить либо справа, либо снизу. Соединяющие линии должны быть направлены к центру символа.

Описание наиболее часто употребляемых символов, предусмотренных ГОСТ 19.701–90, приведены в табл. 1 [5] .

Таблица 1- Графические символы

Наименование	Обозначение	Наименование	Обозначение
Терминатор		Решение	
Ручной ввод		Документ	
Данные		Соединитель	
Процесс		Комментарий	

2.2 Конструирование основных структур

Для разработки схемы алгоритма и текста программы используются только три основные структуры СЛЕДОВАНИЕ и ЦИКЛ.

Структура СЛЕДОВАНИЕ характеризуется последовательным расположением всех входящих в неё фрагментов (следуют один за другим), что обеспечивает поочерёдное выполнение операций.

Структура РАЗВИЛКА включает символ «Решение» (№ 10) и два символа «Процесс» (№ 11 и 12) (рис. 1а).

В структуре РАЗВИЛКА могут отсутствовать символы на одном из направлений. Если на рис. 1а отсутствует символ 11 или 12, то такую сокращённую форму структуры РАЗВИЛКА называют (исходя из выполняемой функции) «обходом».

На рис. 1б приведён вариант с вложенными РАЗВИЛКАМИ. Изображены три структуры РАЗВИЛКА, причём каждая последующая РАЗВИЛКА вложена в предыдущую.

Структура ЦИКЛ имеет несколько разновидностей. ЦИКЛ-ПОКА (ЦИКЛ С ПАРАМЕТРОМ И ПРЕДУСЛОВИЕМ). Эта структура изображена

в виде фрагмента схемы алгоритма на рис. 2 и включает один графический символ «Развилка» (№ 11), три графических символа «Процесс» (№ 10, 12 и 13) и символ «Граница цикла» (не пронумерован), который состоит из двух частей и позволяет наглядно видеть начало и конец цикла (удобно для чтения человеком).

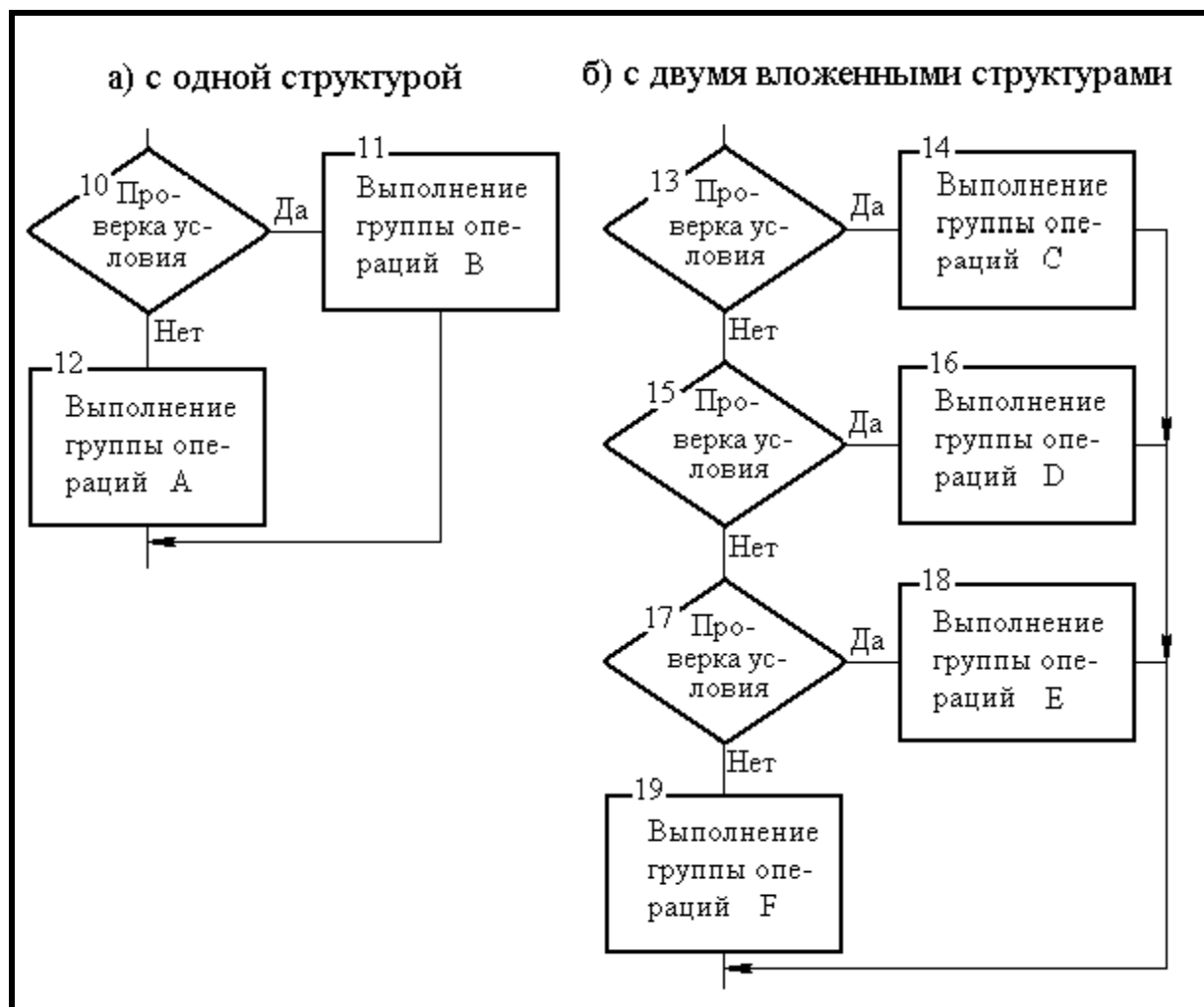


Рисунок 1 – Фрагмент схемы алгоритма со структурой РАЗВИЛКА

ЦИКЛ С ПРЕДУСЛОВИЕМ (ЦИКЛ-ПОКА) состоит из трёх заголовочных символов и тела цикла. Во всех трёх заголовочных графических символах в качестве первого математического символа указывается имя простой переменной (например, x), называемой параметром цикла. В 1-м заголовочном графическом символе (№ 10) записывается начальное значение параметра цикла (например, a), во 2-м (№ 11) – конечное значение параметра цикла (например, c), в 3-м (№ 13) – шаг параметра цикла (например, Δx).

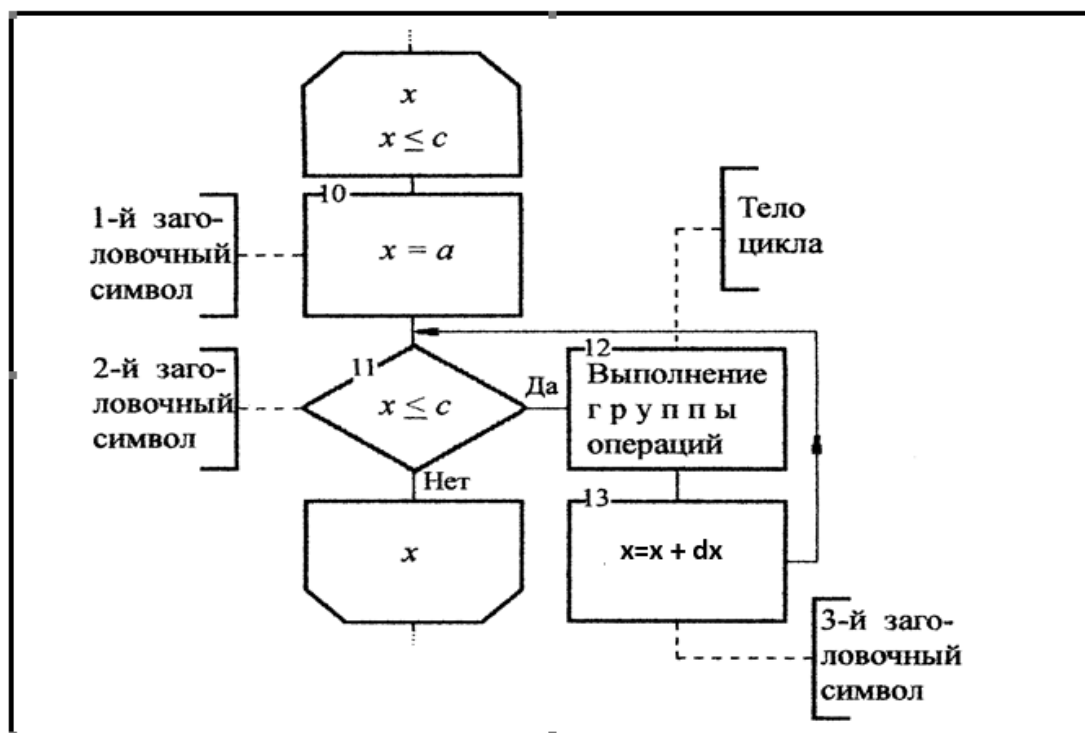


Рисунок 2 - Фрагмент схемы алгоритма со структурой ЦИКЛ С ПРЕДУСЛОВИЕМ

ЦИКЛ С ПРЕДУСЛОВИЕМ обеспечивает выполнение операций до тех пор, пока соблюдается логическое условие, указанное в символе 11. Проверка условия производится до выполнения операций тела цикла (символ 12). Если при первой проверке условие не будет выполнено, то последует выход из цикла по линии потока "Нет", и операции тела цикла не выполняются ни разу.

Выполнение ЦИКЛ С ПРЕДУСЛОВИЕМ (см. рис. 2), характеризуемого зависимостями $a < c$ и $\Delta x > 0$, производится следующим образом. В символе 10 параметру цикла x присваивается начальное значение a . В символе 11 проверяется логическое выражение $x \leq c$. Если оно истинно ("Да"), то выполняется группа операций символа 12, затем в символе 13 изменяется значение параметра цикла на величину шага Δx и в символе 11 снова проверяется выражение $x \leq c$.

2.3 Структурный подход к программированию

Структурный подход к программированию – это подготовка задач к решению на ЭВМ с использованием комплекса методов, позволяющих по сравнению с другими методами разрабатывать в более короткие сроки большие, сложные программы, работающие практически без ошибок (после их отладки). В результате существенно снижаются затраты как на создание, так и на дальнейшую эксплуатацию программ.

Достигается это главным образом благодаря тому, что используются три научно-обоснованных метода (части):

- 1) метод нисходящего проектирования (другие названия: метод проектирования «сверху вниз»; метод пошаговой детализации);
- 2) метод структурного программирования (кодирования);
- 3) метод тестирования программы (сквозного структурного контроля).

Целью структурного подхода к программированию является обеспечение условий для создания надёжной, ясной и несложной для понимания программы, которую можно использовать без участия авторов. В создании таких условий заинтересованы как разработчики программы (сокращается время на выявление и устранение ошибок), так и заказчик, принимающий программу для дальнейшей её эксплуатации (несложно разобраться в тексте программы).

Разработчики алгоритмов при их анализе и конструировании пользуются термином *простой алгоритм*, который характеризуется следующими свойствами:

- определёнными функциональными предназначениями;
- наличием единственного входа;
- наличием единственного выхода.

Умение в алгоритме задачи выделять простые алгоритмы (каждый из них может быть определён как модуль) позволяет составить алгоритм задачи из модулей. В этом случае разрабатывать алгоритм большой задачи может уже не один исполнитель, а группа разработчиков, выполняющих работу параллельно, что и позволяет завершать работу в более короткие сроки.

Модуль – это часть хорошо структурированной системы (например, схемы алгоритма, схемы программы, текста программы), способная автономно выполнять чётко определённые функции.

Запись алгоритма, предназначенного для чтения человеком, может быть произведена различными способами, из которых наибольшее распространение получили:

- словесный способ (технический текст с математическими зависимостями);
- табличный способ (таблица с расчётными формулами);
- графический способ (например, схема функций задачи, схема алгоритма, схема программы и т. п.);
- способ записи текста на алгоритмическом языке (для последующего кодирования, то есть перевода на язык компьютера).

Модульное программирование – это метод, позволяющий создавать программы для компьютера из программных модулей с определением их логических связей.

Преимущества модульного программирования заключаются в следующем:

- сокращаются сроки разработки программы (за счёт отладки программы по частям (модулям) и привлечения для отладки модулей нескольких программистов);

- упрощается отладка всей программы, составленной из ранее отлаженных модулей;
- облегчается сопровождение (обслуживание в процессе эксплуатации) программы, состоящей из отдельных модулей.

Примечание. *Оптимальный размер программного модуля от 20 до 60 строк текста программы, состоящей из операторов (60 строк - это примерно одна страница формата А4 по ГОСТ 9327–60). Сформировать такой модуль следует так чтобы за 10 мин. знакомства с ним можно было понять не менее 90% его содержания.*

Объединение отлаженных программных модулей в единую программу осуществляется в ЭВМ либо при совместной трансляции исходных модулей, либо при редактировании объектных (на машинном языке) модулей и последующим присоединении загрузочных (готовых к использованию) модулей.

2.4 Метод нисходящего проектирования

Метод нисходящего проектирования является универсальным и применяется во всех случаях разумного планирования любой человеческой деятельности. Этот метод в процессе подготовки задачи к решению на компьютере используется четырежды: 1) при разработке иерархической схемы функций задачи; 2) при разработке схемы алгоритма; 3) при составлении схемы программы; 4) при составлении текста программы для компьютера [12].

Три первые названные схемы оформляются графическим способом, но имеют принципиальное различие, заключающееся в следующем: в схеме функций задачи и в схеме программы внимание, концентрируется на определении того, что нужно сделать в разрабатываемой задаче, а в схеме алгоритма [2].

Реализация метода нисходящего проектирования при модульном программировании осуществляется благодаря тому, что составление и отладка программных модулей выполняется в последовательности от верхнего к нижнему уровню. Если отлаживается программный модуль более высокого уровня, то модули нижних уровней могут рассматриваться как «черный ящик» [2].

Программный модуль, представленный как «черный ящик», обычно состоит из операторов, сообщающих о своём выполнении. Для составления и отладки каждого программного модуля должны быть определены области допустимых и возможных значений выходных и входных данных.

2.5 Структурное программирование

Структурное программирование (кодирование) это метод построения программ использующий только иерархически вложенные конструкции, каждая из которых имеет единственную точку входа и

единственную точку выхода [5]. В основе структурного программирования лежит модульное программирование.

Метод структурного программирования используется при построении схем алгоритмов и текстов программ. Метод структурного программирования определяет формирование их из трёх видов типовых основных структур (основных структур), связанных с передачей управления: *последовательной, условного перехода и циклической* [1]. Структурное программирование предполагает, что схема алгоритма и текст программы являются структурированными. Схема алгоритма является структурированной в том случае, если, применив восходящее проектирование («снизу-вверх»), схема будет сведена к одному исходному символу (символу с названием задачи). Структурированная программа имеет наглядную чёткую структуру, позволяющую читать программу «слева направо» и «сверху вниз». Критерием выбора конкретного алгоритмического языка является его способность наилучшим образом реализовать разработанную схему алгоритма.

Метод структурного программирования позволяет создать надёжную программу любой сложности по правилам, облегчающим восприятие человеком текста созданной программы и дальнейшее её сопровождение.

Построение программы на основе методов нисходящего проектирования и структурного программирования, а также стандартизация стиля программирования предоставляет возможность получить программу высокой технологичности, что позволяет создать высококачественный программный продукт.

Стандартизация стиля структурного программирования при составлении текста программы заключается в записи на стандартном бланке со сдвигом на определённое количество позиций операторов и конструкций языка. Конкретные правила сдвига устанавливаются при изучении языка программирования.

2.6 Метод тестирования алгоритма

Метод тестирования алгоритма (сквозного структурного контроля) представляет собой процесс поиска ошибок в схеме алгоритма и программе задачи, подготовленной к решению на компьютере, по заранее составленному плану после отладки программных модулей. Задачей тестирования программы является обнаружение ошибок, а не демонстрация правильности работы программы. Тестирование по существу является опытной эксплуатацией разработанной схемы алгоритма и программы задачи.

До проведения тестирования алгоритма определяют цели тестирования. Ими могут быть проверки: работы программы в диапазоне заданных входных значений; работы программы при минимальных и максимальных значениях входных данных; реакции программы на

недопустимые исходные данные; взаимодействия между программными модулями и др.

Начинают тестирование с проверки схемы алгоритма и устранения арифметических и логических ошибок. Такие ошибки возникают в тех случаях, когда схема алгоритма либо не соответствует математической модели задачи в полном объёме, либо не позволяет получить результат с необходимой точностью. Затем осуществляется проверка работы каждого программного модуля. Проверка схемы алгоритма и текста программы проводится по принципу «сверху вниз»: от модуля верхнего уровня к модулю нижнего уровня.

Завершается тестирование системными испытаниями программы как единого целого и общим контролем взаимодействия между вызывающими и вызываемыми программными модулями. Такое взаимодействие проверяют путём передачи входных и выходных данных от вызывающего к вызываемому модулю и обратно. Если программа для вызываемого модуля не разработана, то такой модуль имитируют путём выдачи выходных данных или операторами, сообщающими о своём выполнении.

Тестирование алгоритма выполняют как на компьютере, так и с использованием других средств с решением минимального количества контрольных примеров.

П р и м е ч а н и е . Наиболее распространёнными ошибками в программах являются следующие:

- неправильное задание начальных значений переменных;
- искажение значений переменных;
- ошибка во внутренних данных программы, в типах данных;
- неверное вычисление значений;
- неверное логическое условие в разветвляющемся вычислительном процессе;
- пропуск одного или нескольких условий при организации разветвляющегося вычислительного процесса;
- неправильная организация циклического вычислительного процесса, в том числе несвоевременный выход из него.

Тестирование алгоритма считается законченным, если осуществлены все требуемые проверки на соответствие схемы алгоритма и программы целям разработки решаемой задачи, а все ошибки устранены. При устранении ошибок следует широко использовать специальные прикладные программы отладчики.

3 ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ Visual Basic for Applications

Грандиозные успехи электроники ведут к удешевлению компьютеров, что способствует всё более широкому использованию их во всех сферах человеческой деятельности. В этих условиях требуется выполнение всё большего объёма работ по программированию. В результате расходы на

программирование в настоящее время превысили общую стоимость компьютера (в начале 90-х годов расходы достигли 90%) и продолжают расти. Пользователь может сокращать эти расходы, если будет сам составлять оригинальные программы для конкретных задач. В этом случае наиболее эффективным средством является использование языков программирования высокого уровня. Появление языков высокого уровня позволило программистам больше времени уделять решению конкретных проблем.

3.1 Основы языка программирования *Visual Basic for Applications*

3.2 Основные понятия

Ключевое слово – это слово или символ, распознаваемый как элемент системы программирования *Visual Basic for Applications (VBA)*, например [Б,К]:

if, then, else, for, next, dim т.д.

Литерал – это простое выражение, определяющее данные некоторого типа, например:

190 – число

"kkkkk" – строковая константа

#27/12/2025# – дата

Константа – это именованная область памяти, сохраняющая постоянное значение в процессе выполнения программы [Б,К].

Переменная – это именованная область памяти, отведенная для временного хранения данных, которые могут изменяться во время выполнения программы [Б,К].

3.3 Алфавит языка

Алфавит языка программирования – это литерный набор, заданный описанием конкретного языка программирования для представления текстов программ и данных ввода-вывода [Б,К]. Литерный набор представляет собой полную и согласованную совокупность знаков (символов), которыми могут быть: буквы, цифры и специальные знаки (запятая, точка, плюс, минус и т. п.) и пробел.

Для языка VBA набор включает 153 символа, распределяемых в зависимости от их предназначения по следующим шести группам:

1) 52 прописные и строчные буквы латинского алфавита: **A-Z**, **a-z**;

2) 66 прописных и строчных букв русского алфавита, используемых при записи комментариев и символьных констант: **A-Я**, **a-я**;

3) 10 арабских цифр: **0-9**;

4) 9 знаков арифметических операций и операций отношения:

+ плюс; **/** деление (слэш); **=** равно;

- минус; **** целочисленное деление (бэкслэш); **<** меньше;

***** умножение; **^** возведение в степень; **>** больше;

5) 10 символов разделителей:

, запятая; **:** двоеточие; **(** открывающая круглая скобка;

. точка; **;** точка с запятой; **)** закрывающая круглая скобка;

' апостроф; **_** подчёркивание; **<пустая позиция>** пробел;

" кавычки;

6) 5 символов, используемых в качестве суффиксов типов данных:

% процент (целые); **&** амперсанд (длинные целые);

! восклицательный знак (вещественные);

номер (вещественные двойной точности).

Кроме операций отношений, указанных в четвертой группе, ещё три операции отношения обозначаются парными символами: **<>** не равно;

<= меньше или равно (не больше); **>=** больше или равно (не меньше).

Конструкции языка строятся по правилам, которые задает синтаксис. Если при построении конструкций нарушаются синтаксические правила, компьютер фиксирует ошибку и, как правило, распознаёт её. Синтаксические ошибки устраняются сравнительно просто, так как компьютер оказывает помощь в их выявлении.

Более сложным является устранение семантических ошибок. Семантика определяет смысловое значение записанного текста, затрагивая тем самым характер и последовательность действий, выполняемых компьютером. Сложность поиска семантических ошибок заключается в том, что такие ошибки могут быть не только в тексте программы, но и в схеме алгоритма, а

то и в математической модели. Никакой помощи в поиске подобных ошибок компьютер оказать не может

3.4 Типы данных

В VBA различают следующие типы данных.

Тип данных	Описание и диапазон значения
Byte	Хранение положительных чисел от 0 до 255
Boolean	Хранение логических значений: True, False
Currency	Хранение чисел для точных вычислений в диапазоне от -922337203685477,5808 до 922337203685477,5807
Date	Хранение даты и времени. Даты от 1.01.100 до 31.12.9999 Время от 00:00:00 до 23:59:59
Double	Хранение чисел двойной точности от $-1,79769313486232 \cdot 10^{308}$ до $-4,94065645841247 \cdot 10^{-324}$ от $4,94065645841247 \cdot 10^{-324}$ до $1,79769313486232 \cdot 10^{308}$
Integer	Хранение целых чисел от -32768 до 32767
Long	Хранение целых чисел от -2147483648 до 2147483647
Single	Хранение чисел одинарной точности от $-3,402823 \cdot 10^{38}$ до $-1,401298 \cdot 10^{-45}$ от $1,401298 \cdot 10^{-45}$ до $3,402823 \cdot 10^{38}$
String	Хранение текста длиной до 2 млрд. символов
Variant	Хранение любого типа данных

Объявление данных

Все данные, используемые в программе, рекомендуется объявить.

Объявление констант

В VBA константы объявляются следующим образом:

Const имя константы [As тип]=значение

Объявленная константа может быть использована в любом выражении, где допускается значение того же типа. Попытка изменения значения константы вызывает ошибку. Пример объявления константы:

Const Pi As Double = 3.14159

Объявление переменных

Переменные объявляются следующим образом:

Dim имя переменной As <тип>

Имя каждой переменной записывается в одной строке. Однако допускается перечислять имена переменных через запятую. Пример объявления переменных:

Dim D As Byte (переменная *D* типа Byte)

Dim K, Z As Single (переменная *K* типа Variant, а *Z* – Single)

Dim D As Integer, C As Double (переменная *D* типа Integer, а *C* – Double)

Dim Znak As String (переменная *Znak* типа String)

Примечания.

1. При присваивании переменным имени следует придерживаться следующих правил:

- использовать только буквы латинского алфавита, регистр букв не имеет значения;
- длина имени не должна превышать 256 символов;
- можно использовать любую комбинацию букв, цифр, символов, кроме точек, пробелов и символов %, &, !, #, @;
- имена должны быть уникальны в той области, в которой они определены.

3.5 Запись формульных выражений

Выражение – это последовательность операций, которые необходимо произвести над данными, чтобы получить требуемое значение.

Результатом выполнения операций выражения может быть числовая или символьная константа, переменная или иное значение.

В VBA существует пять категорий операций: 1) арифметические операции; 2) операции отношения; 3) логические операции; 4) функциональные операции; 5) строковые операции.

3.6 Операции отношения

Операции отношения и примеры записи выражений с их использованием приведены в табл. 2.

Таблица 2. - Операции отношения и их использование

Операции	Название операций	Выражение в VBASIC
=	Равенство	$x = y$
< >	Неравенство	$x < > y$
<	Меньше	$x < y$
>	Больше	$x > y$
<=	Меньше или равно	$x <= y$
>=	Больше или равно	$x >= y$

Логические операции

В VBA имеется шесть логических операций, название и толкование которых приведены в табл. 3.

Логические операции осуществляют манипуляции над битами («Булевы операции»). Они возвращают значение **TRUE** (ИСТИНА – не нулевое значение) и **FALSE** (ЛОЖЬ – нулевое значение).

Таблица 3 - Название логических операций

Операция	Название	Пояснение
NOT	Отрицание	NOT A истинно тогда и только тогда, когда A ложно
AND	Логическое умножение	A AND B истинно тогда и только тогда, когда истинно A и истинно B
OR	Логическое сложение	A OR B истинно тогда и только тогда, когда хотя бы одно из A и B истинно
XOR	Исключающее ИЛИ	A XOR B истинно тогда и только тогда, когда значения A и B не совпадают
EQV	Эквивалентность	A EQV B истинно тогда и только тогда, когда A и B одновременно истинны или одновременно ложны
IMP	Импликация	A IMP B принимает значение ЛОЖЬ, если A истинно, а B ложно, и значение ИСТИНА в других случаях

3.7 Оператор присваивания

Оператор вычисляет значение выражения, стоящего справа от знака равенства, и присваивает его значение переменной, расположенной слева от знака равенства, и имеет вид:

имя переменной=выражение

Выражение – комбинация ключевых слов, функций, переменных и констант, разделенных знаками операций и круглыми скобками. Операции выполняются слева направо, порядок выполнения определяется приоритетом операций и скобками.

При записи арифметических выражений используются следующие символы арифметических операций (табл. 4).

Таблица 4 - Символы арифметических операций

Операции	Знак	Использование	Приоритет
Возведение в степень	$\wedge$	$a \wedge b$	14
Отрицание	$-$	$-b$	13
Умножение	$*$	$a * b$	12
Деление	$/$	a / b	12
Целочисленное деление	$\backslash$	$a \backslash b$	11
Остаток от деления на целое	Mod	$a \text{ Mod } b$	10
Сложение	$+$	$a + b$	9
Вычитание	$-$	$a - b$	9

3.8 Встроенные функции VBA

В VBA имеется большой набор встроенных функций, использование которых существенно облегчает процесс программирования. Обращение к встроенной функции записывается следующим образом:

имя функции(аргумент)

Имя функции определяет имя подпрограммы, вычисляющей значение функции. После выполнения подпрограммы имени функции присваивается вычисленное значение.

3.8.1 ВСТРОЕННЫЕ ФУНКЦИИ

В VB имеется большой набор встроенных функций, использование которых существенно облегчает процесс программирования. Обращение к встроенной функции записывается следующим образом:

имя функции(аргумент)

Имя функции определяет имя подпрограммы, вычисляющей значение функции. После выполнения подпрограммы имени функции присваивается вычисленное значение. В библиотеке VB хранятся следующие встроенные функции, приведенные в табл. 5.

Тригонометрические функции работают в радианной мере. Преобразование градусной меры в радианную и обратно выполняется следующим образом:

Значение угла в радианах = Значение угла в градусах * $\pi/180$

Значение угла в градусах = Значение угла в радианах * $180/\pi$

Таблица 5. Основные стандартные математические функции и функции обработки данных

Обозначение функции		Пояснения
В математике	В VBASIC	
$\sin x$	SIN(X)	x – задаваемый угол в радианах; для преобразования градусов в радианы следует число градусов умножить на $\pi/180^\circ$
$\cos x$	COS(X)	
$\operatorname{tg} x$	TAN(X)	
$\operatorname{arctg} x$	ATN(X)	Значение угла допустимо задавать в диапазоне от $-\pi/2$ до $\pi/2$
$\ln x$	LOG(X)	$x > 0$
e^x	EXP(X)	$x < 88,02969$
$ x $	ABS(X)	Абсолютное значение x
$\sqrt{x}$	SQR(X)	$x > 0$
$[x]$	INT(X)	Функция выводит число, округленное до меньшего целого

Примечание. Логарифм по заданному основанию можно получить путем деления натурального логарифма числа на натуральный логарифм заданного основания.

$$Lg_{10} x = \operatorname{Log}(x) / \operatorname{Log}(10)$$

Все остальные функции могут быть получены из этих четырех. Полный список тригонометрических функций можно найти в справочной библиотеке Visual Basic. Тригонометрические функции работают в радианной мере. Преобразование градусной меры в радианную и обратно выполняется следующим образом:

$$\text{Значение угла в радианах} = \text{Значение угла в градусах} * \pi / 180$$

$$\text{Значение угла в градусах} = \text{Значение угла в радианах} * 180 / \pi$$

В стандартном Visual Basic для определения константы π в секции General необходимо поместить код:

$$\text{Constant Pi As Double} = 3.14159$$

Рассмотрим примеры записи математических выражений на языке VBA [б,к];

$$1. \gamma = \frac{\lg \frac{a}{b} \cdot \sqrt{\sin x + \cos x^2}}{|a^b|} \cdot 10^{-5}$$

$$\text{Gamma} = \text{Log}(a/b) / \text{Log}(10) * \text{Sqr}(\text{Sin}(x) + \text{Cos}(x^2)) / \text{Abs}(a^b) * 1E-5$$

$$2. \beta = e^{a+b} \cdot \sqrt[3]{\arctg^2 a} \cdot \frac{1 + \frac{\gamma}{2}}{3 \cdot \text{tg} \frac{l}{2}} \cdot 1,5 \cdot 10^6$$

$$B = \text{Exp}(a+b) * \text{Atn}(a)^{2^{1/3}} * (1 + \text{gam}/2) / (3 * \text{Tan}(l/2)) * 1.5E6$$

$$3. R = 0,00000015 \cdot \varphi$$

$$R = 15e-8 * fi$$

3.9 Организация ввода данных

Ввод данных можно осуществить с помощью функции InputBox() или текстового поля.

3.9.1 Функция InputBox()

С помощью этой функции ввод данных производится в стандартное диалоговое окно (рис.3).

Сокращенный синтаксис функции выглядит так:

$$a = \text{InputBox}(\text{сообщение}),$$

где, *a* – имя переменной, которой требуется присвоить введенное значение;
сообщение – это текст, отображаемый в диалоговом окне

Рассмотрим фрагмент кода ввода значения площади *S*.

```
Dim S As Double
```

```
S = InputBox("Введите значение площади")
```

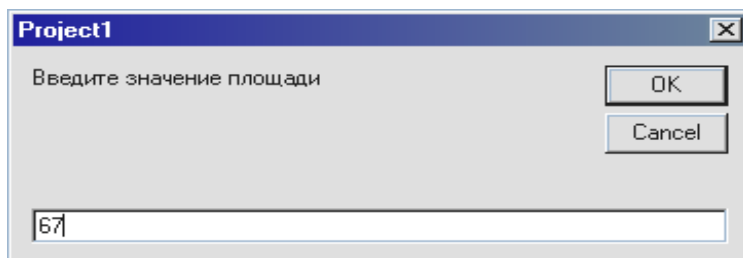


Рисунок 3 – Окно фрагмента реализации кода ввода

Вызов функции возможно организовать следующим образом:

S = InputBox("Введите значение площади", "Задание № 1") (рис.4):

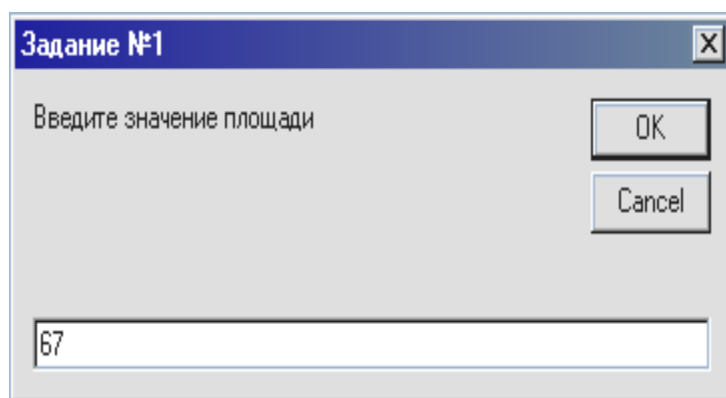


Рисунок 4 – Окно фрагмента реализации кода ввода

3.9.2 ВВОД ДАННЫХ С ПОМОЩЬЮ ТЕКСТОВОГО ПОЛЯ

Для организации ввода значения *S* с помощью текстового поля необходимо выполнить следующие действия:

- на форме разместить текстовое поле для ввода значения площади (элемент управления "Текстовое поле", по умолчанию свойство *Name* этого элемента имеет значение *Text1*);
- в программе записать оператор

```
S=Val(TextBox1.Text) или S=Text1
```

3.9.3 ОРГАНИЗАЦИЯ ВЫВОДА ДАННЫХ

Вывод данных можно выполнить различными способами, например:

- с помощью оператора `MsgBox`;
- с помощью функции `MsgBox()`;
- с помощью оператора `Debug.Print`;
- с помощью текстового поля.

3.9.4 ОПЕРАТОР `MSGBOX`

Оператор `MsgBox` выводит данные в стандартное диалоговое окно, которое может иметь текст сообщения, заголовок и атрибут.

Синтаксис оператора вывода сообщений в стандартное диалоговое окно следующий:

MsgBox сообщение,

где

сообщение – это текст, отображаемый в диалоговом окне (текст должен быть заключен в двойные кавычки и не превышать 1024 символов);

3.9.5 `FORMAT(ПАРАМЕТР,"ПРОТОТИП")`

При построении прототипа можно использовать символы:

- 0 – резервирует позицию цифрового разряда, отображает цифру, если у числа, представленного параметром, есть цифра в этой позиции;
- # – аналогичен первому параметру, но не отображаются незначащие нули;
- . (точка) – разделитель целой и дробной частей числа.

Например,

использование функции `Format(2.2^2, "##.###")` дает результат 4.84, а использование функции `Format(2.2^2, "00.000")` дает результат 04.840.

3.9.6 МЕТОД `DEBUG.PRINT`

Метод `Debug.Print` позволяет результаты вычислений выводить на форму.

Если в методе `Debug.Print` выводимые значения разделяются *точкой с запятой*, то они печатаются один за другим через пробел, а если *запятой*, то каждое новое значение печатается в начале следующей зоны печати (в Visual Basic зоны начинаются через каждые 14 символов). Изменить ширину зоны позволяет функция ***Tab*** (число).

Debug.Print Tab(8); format(k,"0.00"); Tab(15); d

'Вывод в строку с использованием значений табуляции

3.9.7 ВЫВОД ДАННЫХ С ПОМОЩЬЮ ТЕКСТОВОГО ПОЛЯ

Например, для организации вывода значения переменной **d** с помощью текстового поля необходимо выполнить следующие действия:

- на форме разместить текстовое поле для вывода в него значения **d** (элемент управления "Текстовое поле"; пусть свойство *Name* этого элемента будет *TextBox3*);

- в программе записать оператор: *TextBox3.Text=Str(d)* или *TextBox3=d*, где **Str ()** – функция, преобразующая числовое значение в строковое.

Тогда при выполнении программы в текстовое поле, имеющее имя *TextBox3*, выведется значение переменной **d**.

3.9.8 ОПЕРАТОРЫ END SUB И REM

End Sub – Оператор - указатель обозначает конца программы, процедуры или блока.

Rem – оператор, указывающий на комментарий. Является невыполняемым, используется для облегчения чтения и понимания человеком текста программы. Язык VBA позволяет записывать комментарий с использованием апострофа ['].

Синтаксис операторов комментария

REM < Текст > или '< Текст >

Примеры: 1) **REM** Программа_Лабораторная работа №2
2) 'Программа_Лабораторная работа №2

3.10 Оператор условного перехода IF

Оператор условного перехода **IF**, используя метод парных сравнений, определяет значение логического (булевского) выражения (значение условия) и в зависимости от результата производит выбор одного или группы операторов, позволяющих осуществить вычислительный процесс по одному из двух направлений. Этот оператор формирует основную структуру условного перехода (называемую для краткости РАЗВИЛКА).

Система VBASIC позволяет записывать оператор **IF** как в блочной, так и в линейной форме.

Блочная форма записи оператора IF представляет собой общий блок операторов, ключевых слов и логических выражений. Начинается общий блок с имени оператора **IF**, записываемого в первой строке, а заканчивается конструкцией **END IF**, записываемой в последней строке.

Синтаксис блочной формы записи оператора IF

```
IF_<1-е_ЛВ>_THEN
.....<операторы 1-й группы>
...[ELSEIF_<2-е_ЛВ>_THEN
.....<операторы 2-й группы>]]
...[ELSEIF ....
.....[ELSEIF_<(n-1)-е_ЛВ>_THEN
.....<операторы (n-1)-й группы>]]
.....[ELSE_<операторы n-й группы>]
END_IF
```

где

IF и **ELSEIF** – имена операторов соответственно общего блока и частных блоков;

END IF – оператор **END** с аргументом **IF** (конец оператора **IF**);

THEN и **ELSE** – ключевые слова;

ЛВ – логическое выражение;

n – количество условий в математической модели;

_ – символ, обозначающий пробел.

IF и **END IF** образуют **операторные скобки** (открывающую и закрывающую). Такие скобки, являющиеся характерными для структурного программирования, определяют границы действия оператора и записываются на определённом расстоянии от границы левого поля.

В блочную структуру **IF** можно вставлять любое конечное количество логических выражений, относящихся к **ELSEIF**. Количество логических выражений всегда будет на единицу меньше количества условий, указываемых в математической модели.

Семантика блочной формы записи оператора IF: ЕСЛИ 1-е логическое выражение (ЛВ) «истина», ТО выполняется блок **THEN** с операторами 1-й группы (прочтены 1-я и 2-я строки, указанные в синтаксисе), ИНАЧЕ выполняется первый вложенный оператор **ELSEIF**. ЕСЛИ 2-е ЛВ «истина», ТО выполняется блок **THEN** с операторами 2-й группы (прочтены третья и четвертая строки) ИНАЧЕ ... (и так продолжать до предпоследнего вложенного оператора **ELSEIF** с (n-2)-м ЛВ). Последний вложенный оператор **ELSEIF** читается так: ЕСЛИ (n-1)-е ЛВ «истина», ТО выполняется блок **THEN** с операторами (n-1)-й группы, ИНАЧЕ выполняется блок **ELSE** с операторами n-й группы.

Вполне очевидно, что в том случае, если ни одно из условий **ELSEIF** не выполнено (т. е. все ЛВ оказались «ложь»), то выполняется блок **ELSE** с операторами n-й группы. В другом случае ни одна группа операторов со 2-й до (n-1)-й, относящаяся к **ELSEIF**, не будет выполняться, если 1-е ЛВ окажется «истина». Таким образом в обоих рассмотренных случаях в вычислительном процессе участвует только одна из двух групп операторов группы, относящиеся к внешней РАЗВИЛКЕ (либо 1-я, либо n-я).

Каждый последующий оператор **ELSEIF** (реализующий очередную РАЗВИЛКУ) вложен в предыдущий (см. рис.5 и идентичные вложения структур РАЗВИЛКА на рис. 1.б). Если у рассматриваемого **ELSEIF** значение ЛВ «истина», то к вычислительному процессу подключается группа операторов, относящаяся только к этому ЛВ, и действие оператора **IF** прекращается.

Имя оператора **IF**, а также **ELSEIF**, **ELSE** и **END IF** должны быть записаны в строке обязательно первыми. Запись групп операторов, относящихся к блокам **THEN** и **ELSE**, должна производиться только в тех строках, которые указаны в синтаксисе блочной формы записи. **ELSEIF** и **ELSE** вместе с относящимися к ним группам операторов могут быть опущены. Признаком, по которому система VBA узнаёт блочную форму записи, является отсутствие в строке после **THEN** какого-либо оператора (но комментарий допускается).

Для удобства чтения человеком (требование ГОСТ 19781–90) оператора **IF** его составляющие следует располагать уступом, как это показано в синтаксисе этого оператора при записи в блочной форме.

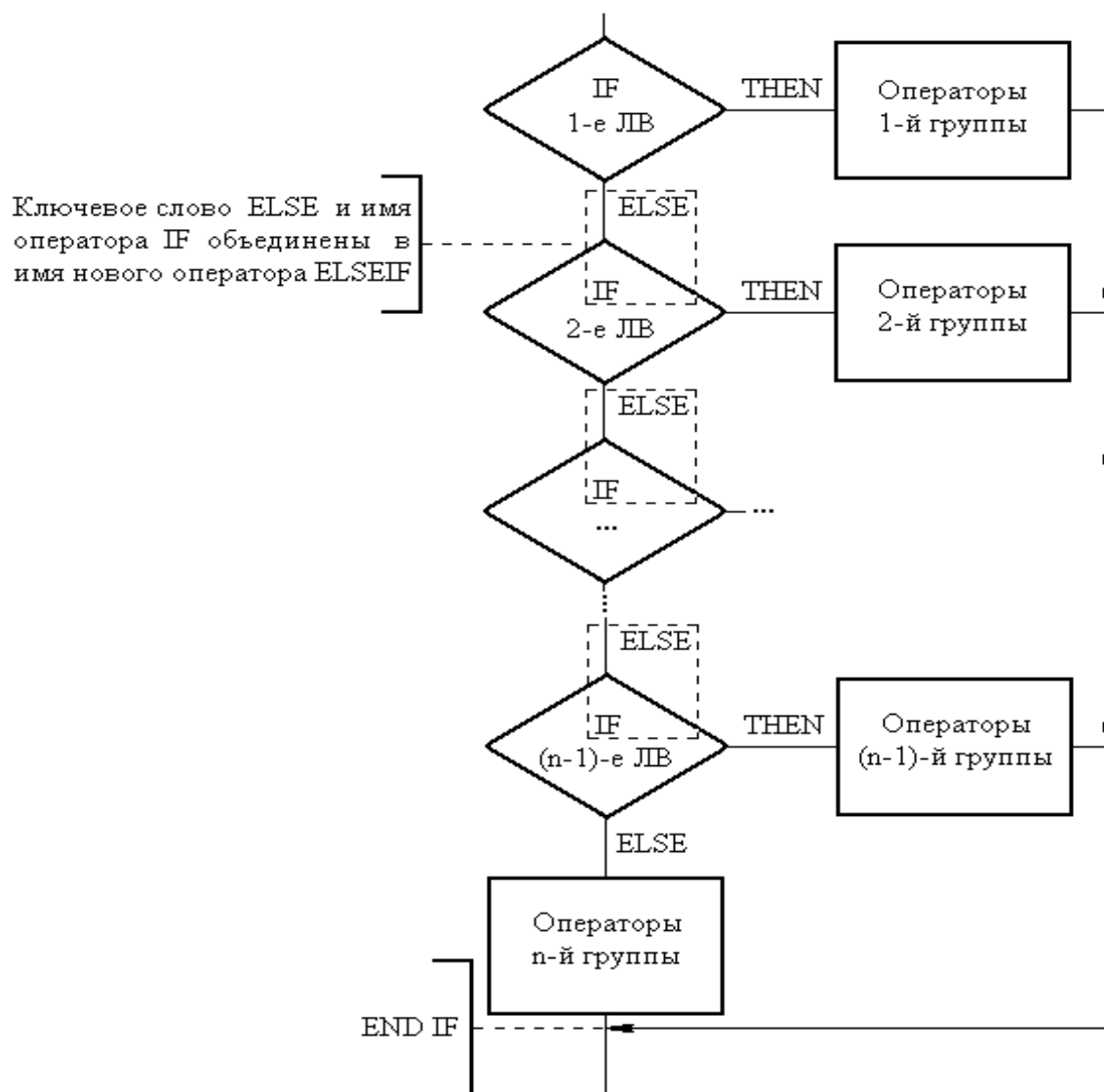


Рис. 5 Схема алгоритма записи оператора **IF** при записи в блочной форме

3.11 Операторы цикла

3.11.1 ОБЩАЯ ХАРАКТЕРИСТИКА ОПЕРАТОРОВ ЦИКЛА

В VBASIC используется группа операторов цикла трёх видов:

- 1) **FOR...NEXT**;
- 2) **DO WHILE(UNTEL).. ..LOOP**;
- 3) **DO...LOOP WHILE(UNTEL)**.

Оператор **FOR...NEXT** и **DO WHILE(UNTEL).. ..LOOP** имеет структуру **ЦИКЛ С ПРЕДУСЛОВИЕМ**. Этот оператор позволяет с помощью трёх задаваемых значений параметра цикла (начального, конечного и шага) повторять определённое количество раз действия, указываемые в блоке операторов тела цикла.

Второй и третий виды операторов **DO.. ..LOOP WHILE(UNTEL)** имеют структуру **ЦИКЛ С ПОСТУСЛОВИЕМ**, в которой блок операторов

тела цикла повторяется (неизвестное заранее количество раз) до тех пор, пока условие не станет **истинным** (Не ноль; Да) или **ложным** (Ноль; Нет).

Три вида операторов по признаку расположения условия по отношению к телу цикла характеризуются следующими структурами:

- оператор **FOR.. ..NEXT** имеет структуру **ЦИКЛ С ПРЕДУСЛОВИЕМ** (ЦИКЛ-ПОКА);

- в группе **DO WHILE(UNTEL).. ..LOOP** и

DO.. ..LOOP WHILE(UNTEL) одни операторы имеют структуру **ЦИКЛ С ПРЕДУСЛОВИЕМ** (ЦИКЛ-ПОКА); а другие **ЦИКЛ С ПОСТУСЛОВИЕМ** (ЦИКЛ-ДО).

Для операторов ЦИКЛА характерны следующие общие правила:

- а) вход в ЦИКЛ возможен только через его начало, т. е. через его заголовок;

- б) все начальные значения, которые входят в логическое выражение во всех операторах, реализующих **ЦИКЛ С ПРЕДУСЛОВИЕМ**, должны быть определены до входа в ЦИКЛ;

- в) в теле цикла оператора **FOR.. ..NEXT** не допускается переопределять значения параметра цикла.

3.11.2 ОПЕРАТОРЫ ЦИКЛА С ПРЕДУСЛОВИЕМ FOR.. ..NEXT и DO WHILE.. ..LOOP

Реализует только ЦИКЛ С ПРЕДУСЛОВИЕМ (фрагмент схемы алгоритма с подробным описанием структуры реализуется рассматриваемым оператором **FOR.. ..NEXT**, приведен на рис. 6).

Синтаксис оператора

FOR _<ПЦ> _ = _<начное_ЗПЦ> _<ТО> _<конечное_ЗПЦ> _[**STEP** _<шаг_ПЦ>]
_<блок операторов тела цикла>
NEXT _<ПЦ>

где **FOR** – имя оператора.

С ним в одной строке (заголовке цикла) записываются:

- 1) **ПЦ** (параметр цикла, т. е. имя числовой простой переменной);
- 2) начальное **ЗПЦ** (значение параметра цикла: числовое значение или выражение), конечное **ЗПЦ** и шаг **ПЦ** (значение шага);
- 3) ключевые слова: **ТО** и **STEP**;

NEXT – ключевое слово с тем же ПЦ, который указан в заголовке цикла.

В операторе **FOR** заранее задаётся количество повторений операторов тела цикла благодаря трём значениям параметра цикла, указываемых в заголовке цикла. Если **STEP** (шаг параметра цикла) не указан, то система VBASIC «по умолчанию» присваивает ему значение, равное единице.

NEXT - изменяет параметр цикла после каждого повторения цикла.
FOR и **NEXT** - выполняют роль операторных скобок, определяющих начало

и конец оператора.

Цикл будет выполняться, если сумма начального значения **ПЦ** и положительного шага **ПЦ**, будет меньше или равна (а при отрицательном шаге больше или равна) конечному значению **ПЦ**. При равенстве начального и конечного значений **ПЦ** цикл будет выполнен один раз. При шаге **ПЦ** равном нулю - произойдёт «зацикливание»).

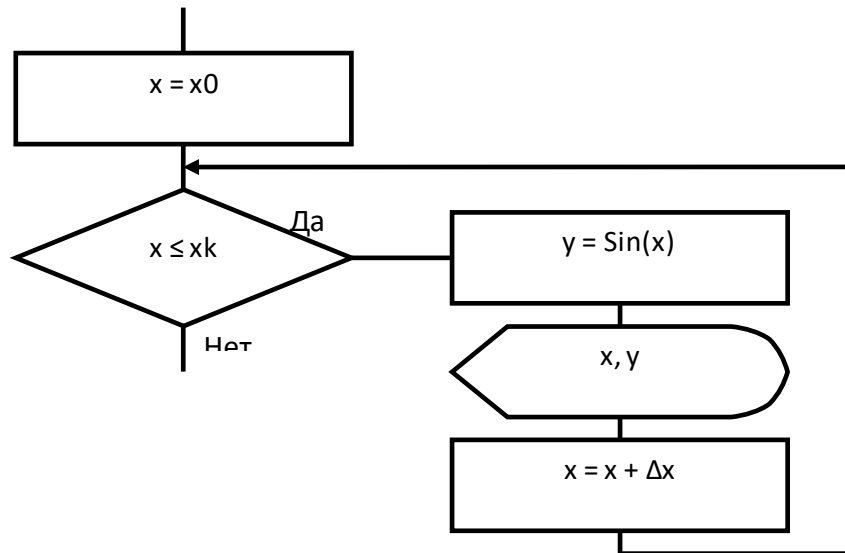


Рисунок 6 - Фрагмент схемы алгоритма Цикл с ПРЕДУСЛОВИЕМ

При разработке кода приложения, использующего основную алгоритмическую структуру Цикл с предусловием

*Использование оператора **FOR....NEXT**.*

Фрагмент кода приложения при конкретных значениях x_0 , x_k и Δx может быть записан в виде:

For $x = x_0$ **To** x_k **Step** dx

$y = \sin(x)$

Print "x = " & *format*(x , "0.00") & " y = " & *format*(y , "0.00")

Next

*Использование оператора **Do While.. ..Loop**.*

Фрагмент кода приложения может быть записан в виде:

$x = x_0$

Do While $x \leq x_k$

$y = \sin(x)$

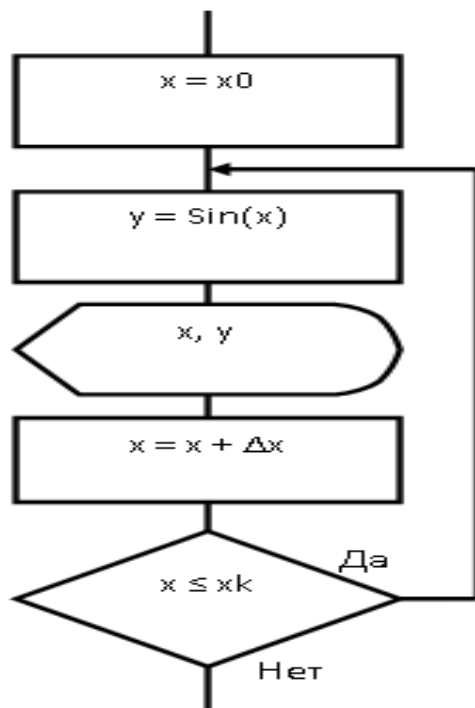
Print "x = " & *format*(x , "0.00") & " y = " & *format*(y , "0.00")

$$x = x + dx$$

Loop

3.11.3 ОПЕРАТОРЫ ЦИКЛА С ПОСТУСЛОВИЕМ DO.. ..LOOP WHILE

При вычислении значений функций y и вывода полученных значений на экран монитора в соответствии с алгоритмом, реализующим структуру *Цикл с постусловием* (рис.7), используется оператор цикла:



$x = x0$

Do

Тело цикла

$x = x + dx$

Loop While $x \leq xk$

Фрагмент кода с данными

$x = 0$

Do

$y = \text{Sin}(x)$

Debug.Print "x = " & format(x, "0.00") &

" y = " & format(y, "0.0")

$x = x + 2$

Loop while $x \leq 10$

Рисунок 7 – Фрагмент алгоритма

Цикла с ПОСТУСЛОВИЕМ

4 ПОДГОТОВКА ЗАДАЧ К РЕШЕНИЮ НА КОМПЬЮТЕРЕ

4.1 Основные требования к оформлению отчётного документа при подготовке задач к решению на компьютере

Основные требования ГОСТ 19.701–90 к оформлению схемы алгоритма приведены в п. 2.1. С учётом поставленной цели и требований Госстандартов отчётный документ, связанный с подготовкой задачи к решению на ПК, должен иметь следующую структуру [12]:

1. Описание постановки задачи (функциональная часть).
 - 1.1. Характеристика задачи.
 - 1.2. Выходные данные.
 - 1.3. Входные данные (постоянные и переменные).
2. Описание алгоритма (математическое обеспечение).
 - 2.1. Математическое описание (в том числе модель задачи).

- 2.2. Выбор метода решения.
 - 2.3. Алгоритм решения (в виде схемы алгоритма).
 - 3. Отладка программы (программное обеспечение):
 - 3.1. Исходный текст программы.
 - 3.2. Исходные данные (их значения).
 - 3.3. Выявление и исправление синтаксических и семантических ошибок.
 - 3.4. Результаты решения задачи другими средствами.
 - 3.5. Результаты решения задачи на компьютере.
 - 4. Текст программы (программное обеспечение).
 - 4.1. Описание программных элементов.
 - 4.2. Текст ...-программы (вместо многоточия указывается название языка программирования).
- Вывод о работоспособности программы.

5 Примеры реализации теоретических положений при подготовке задач к решению на компьютере

5.1 Пример 1. Подготовка к решению задачи со структурой СЛЕДОВАНИЕ

1. Постановка задачи

1.1. Характеристика задачи. Определить целым числом ускорение a ($м/с^2$) скатывающегося с горки вагона массой m ($т$) с учетом инерции его вращающихся частей по формуле

$$a = \frac{g}{1 + \frac{kn}{m}}$$

где g - постоянная ускорения свободного падения, $м/с^2$;

k - коэффициент, учитывающий инерцию вращающихся частей вагона и обеспечивающий совместимость единиц измерения;

n - количество осей вагона, ось.

При решении контрольного примера входным данным присвоить значения: а) постоянным: $g = 9,81 м/с^2$; $k = 0,42 т/ось$;

б) переменным целого типа: $n = 4$ оси; $m = 60 т$.

1.2. Выходные данные:

а) перечень выходных данных: n , m - переменные целого типа

a - переменная вещественного типа

1.3. Входные данные: а) постоянные: g , k ; б) переменные: n , m .

2. ОПИСАНИЕ АЛГОРИТМА

2.1. Математическое описание. Модель задачи:

$$g = 9,81 \quad (1)$$

$$k = 0,42 \quad (2)$$

$$a = \frac{g}{1 + \frac{kn}{m}} \quad (3)$$

2.2. Выбор метода решения. Выбор не производится, так как при решении задачи выполнение арифметических действий определяется последовательностью записи формул (1) – (3).

2.3. Алгоритм решения. Приведён на рис.8 в виде схемы алгоритма решения задачи

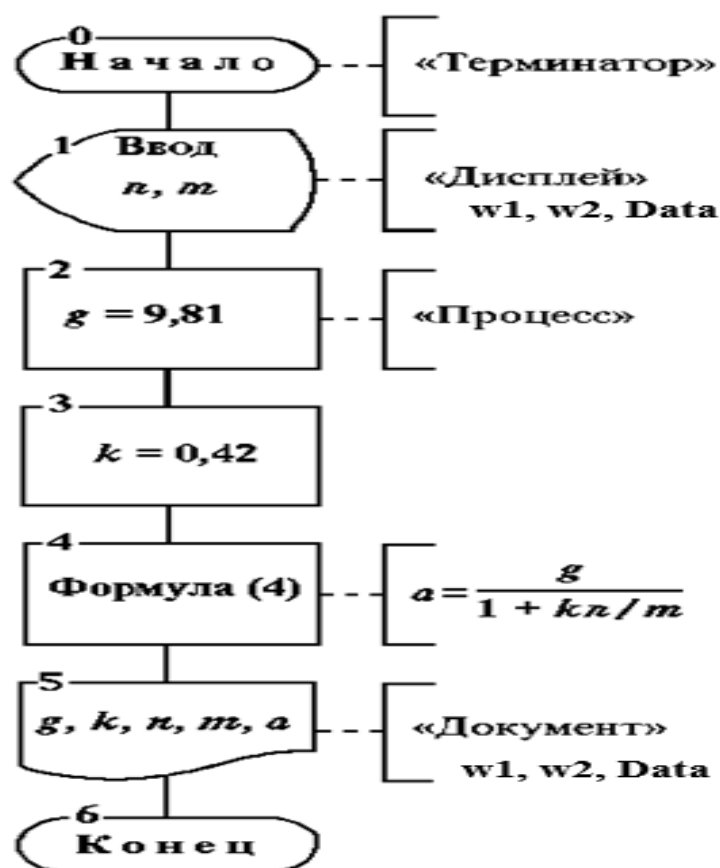


Рисунок 8 - Схема алгоритма

4. ТЕКСТ ПРОГРАММЫ

4.2. Программный код

```

Private Sub CommandButton1_Click()
'Программа А_И_Веревкин. Структура СЛЕДОВАНИЕ
'Задание N 16. Лабораторная работ N 2
Dim n As Integer, m As Integer
Dim w1 As String, w2 As String, Data As String
Const g As Single = 9.81
Const k As Single = 0.42
w1 = InputBox("Укажите учебную группу")
w2 = InputBox("Укажите инициалы имени и отчества, фамилию")
Data = InputBox("Укажите дату тремя парами арабских цифр")
n = Val(TextBox1.Text)
m = Val(TextBox2.Text)
a = g / (1 + k * n / m)
TextBox3 = Str(a)
Label4.Caption = "Исполнил студент  " & w1 + "
Label5.Caption = Data
End Sub

```

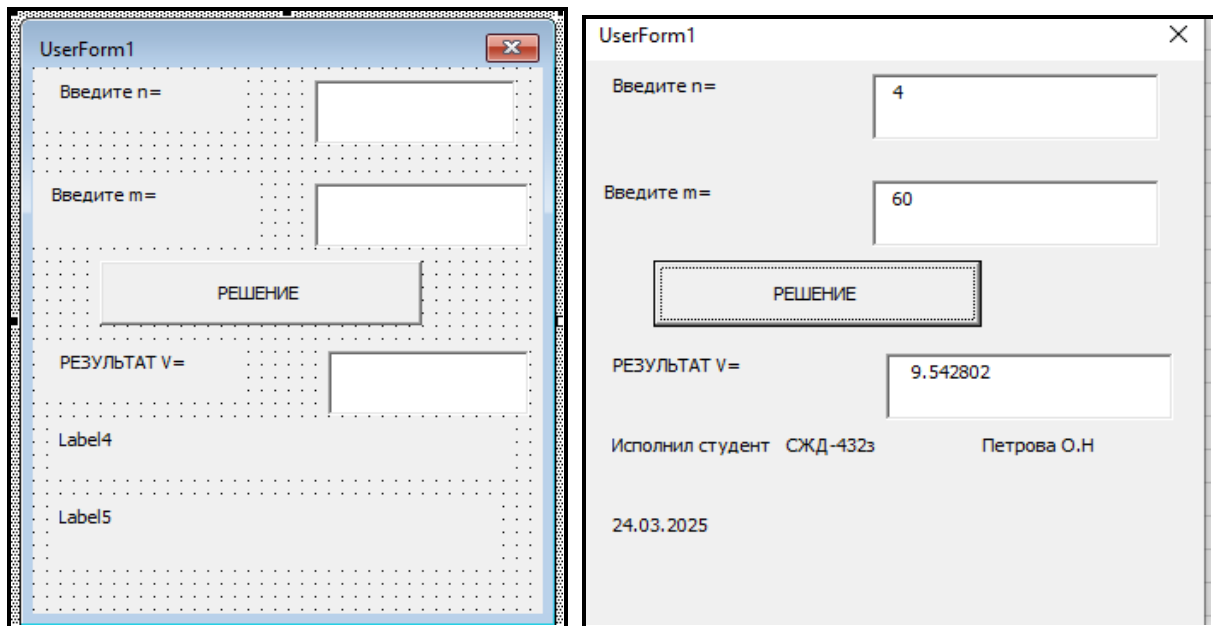


Рисунок 9 - Результат решения
Решение примера в Python

```

1  #Программа А_И_Веревкин. Структура СЛЕДОВАНИЕ
2  #Задание N 16. Лабораторная работ N 2
3  G = 9.81
4  K = 0.420
5  w1 = input('Укажите учебную группу ')
6  w2 = input('Укажите инициалы имени и отчества, фамилию ')
7  Data = input('Укажите дату тремя парами арабских цифр ')
8  n = float(input('Введите n: '))
9  m = float(input('Введите m: '))
10 a = G / (1 + K * n / m)
11 print ('Результат a =', a)
12 print ('Исполнил студент  ' , w1 , '      ' , w2)
13 print(Data)

```

Укажите учебную группу *СЖД-432з*

Укажите инициалы имени и отчества, фамилию *А.И.Веревкин*

Укажите дату тремя парами арабских цифр *24.03.2025*

Введите n: *4*

Введите m: *60*

Результат a = 9.542801556420233

Исполнил студент СЖД-432з А.И.Веревкин

24.03.2025

5.2 Пример 2. Подготовка к решению задачи со структурой РАЗВИЛКА

1. ОПИСАНИЕ ПОСТАНОВКИ ЗАДАЧИ

1.1. **Характеристика задачи.** При подготовке данной задачи к решению на ПК текстовую часть рассматривать как математическое множество (является очень эффективным приёмом при многократном использовании одних и тех же коротких текстов в различных местах документа).

1.1.1. Характеристика текстовой части задачи. Оформить результаты решения контрольных примеров с использованием необходимых заголовков и поясняющих текстов, указанных в задании. Заголовки и поясняющие тексты в качестве математических множеств следует включить не только в текст программы, но и в математическую модель, и в схему алгоритма. Результаты решения контрольных примеров завершить реквизитами “Подпись” и “Дата”.

Для обеспечения логических построений, связанных с выдачей на печать необходимых заголовков и поясняющих текстов, использовать переменные целого типа: n (количество решаемых контрольных примеров) и i (порядковый номер решаемого контрольного примера). При оформлении использовать: для реквизита “Подпись” – множество W_1 (наименование должности и воинское звание) и множество W_2 (расшифровка подписи), а для реквизита “Дата” – множество D_{ATA} .

1.1.2. Характеристика расчётной части задачи. Вычислить значения функции y с точностью до сотых по формулам

$$y = \begin{cases} a + b + c^2, & \text{если } a = -1; & (1) \\ a + \sin(c + \pi), & \text{если } a = 0; & (2) \\ \sqrt{a + bc}, & \text{если } a = 1; & (3) \\ a \ln c - \text{в остальных случаях;} & (4) \end{cases}$$

где постоянная $\pi = 3,14$; типы переменных: a – целый; b, c – вещественный.

Разработать макет выходного документа. При решении контрольных примеров переменным a, b, c присвоить значения:

- для второго – $a = -1; b = 6,52; c = 3,7;$
- для третьего – $a = 0; b = 11,03; c = 5,6;$

- для четвёртого – $a = 1$; $b = 8, 35$; $c = 4, 2$.

1.2. Выходные данные:

а) перечень выходных данных:

- скалярных величин: n, i, π, a, b, c, y ;

- множеств: $X_1 - X_6, W_1, W_2, D_{ATA}$;

б) расположение выходных данных определено в макете выходного документа, представленного на рис. 10.

1.3. Входные данные:

а) постоянные: π ;

б) переменные: $n, i, a, b, c, X_6, W_1, W_2, DATA$.

2. ОПИСАНИЕ АЛГОРИТМА

2.1. Математическое описание. Модель задачи:

$$y = \begin{cases} a + b + c^2, & \text{если } a = -1; & (1) \\ a + \sin(c + \pi), & \text{если } a = 0; & (2) \\ \sqrt{a + bc}, & \text{если } a = 1; & (3) \\ a \ln c - \text{в остальных случаях;} & (4) \end{cases}$$

2.2. **Выбор метода решения.** При решении задачи следует использовать метод парных сравнений, так как выбор вариантов решения производится путем

сравнения двух величин. Так, в рассматриваемой модели задачи:

1) в формулах (1) – (4) численное значение « a » сравнивается соответственно с « -1 », « 0 » и « 1 »;

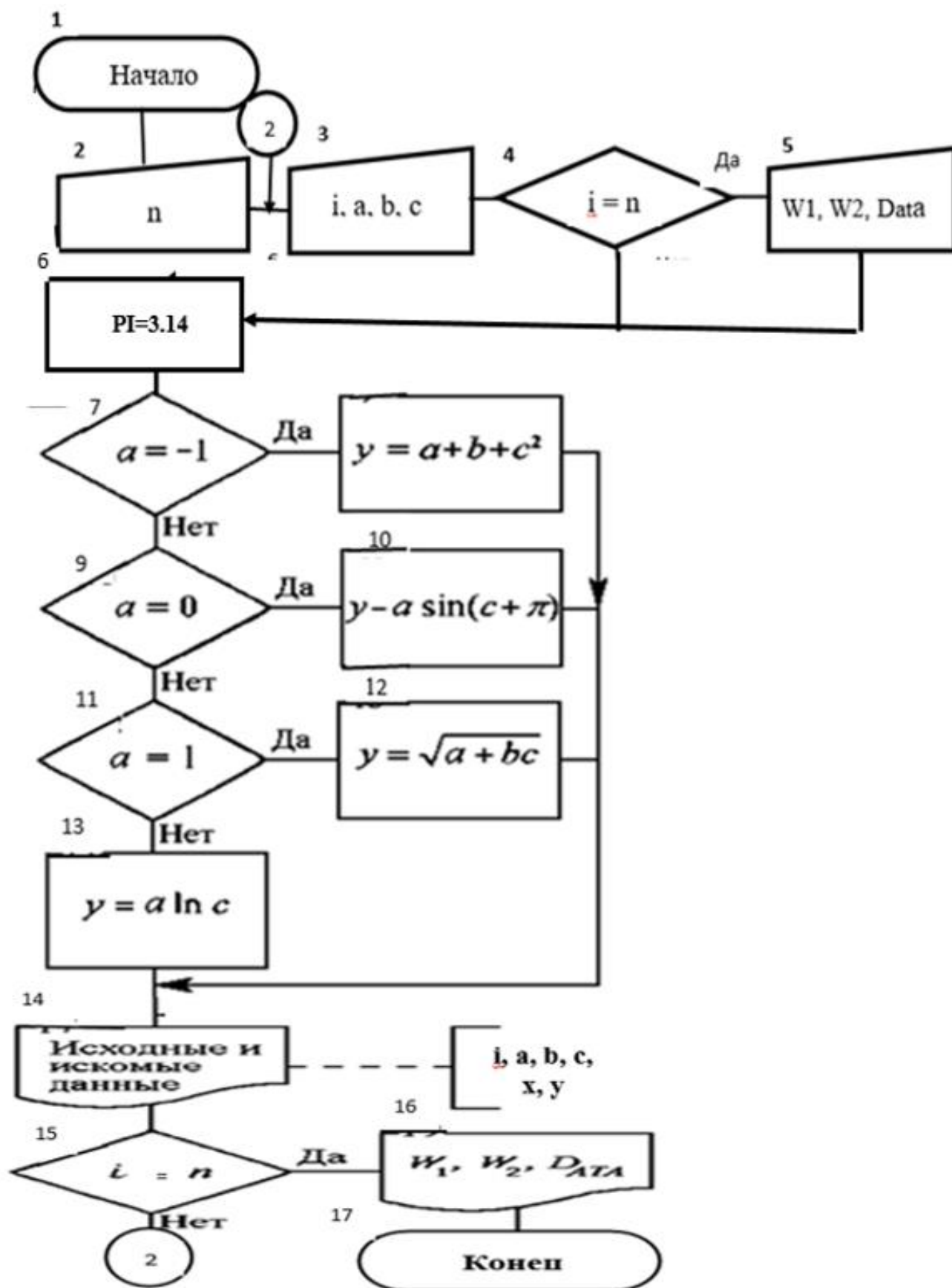


Рисунок 10 - Схема алгоритма

4.2. Программный код

CommandButton1

```
Private Sub CommandButton1_Click()  
Dim n As Integer, i As Integer, a As Integer  
Dim b As Single, c As Single, y As Single  
Const pi As Single = 3.14  
Dim W1 As String, W2 As String, Data As String  
n = InputBox("Сколько решается контрольных примеров? n =")  
i = InputBox("Укажите порядковый номер решаемого примера i =")  
a = InputBox("Введите число целого типа a =")  
b = InputBox("Введите число вещественного типа b =")  
c = InputBox("Введите число вещественного типа c =")  
If i = n Then  
    W1 = InputBox("Укажите учебную группу")  
    W2 = InputBox("Укажите инициалы имени и отчества, фамилию")  
    Data = InputBox("Укажите дату тремя парами арабских цифр")  
End If  
If a = -1 Then  
    y = a + b + c ^ 2  
ElseIf a = 0 Then  
    y = a * Sin(c + pi)  
ElseIf a = 1 Then  
    y = Sqr(a + b * c)  
Else  
    y = a * Log(c)  
End If  
If i = 1 Then  
    Debug.Print " PI = "; pi: Debug.Print "    n = "; n  
End If  
Debug.Print  
Debug.Print " i = " & Format(i, "#") & " a = " & Format(a, "##") & " b = ";  
Debug.Print Format(b, "#0.0#") & " c = " & Format(c, "0#.#0")  
Debug.Print  
Debug.Print " y = " & Format(y, "##.##")  
If i < n Then GoTo 1:  
Debug.Print: Debug.Print "Учебная группа"; W1; Tab(44); W2  
Debug.Print Data  
End Sub
```

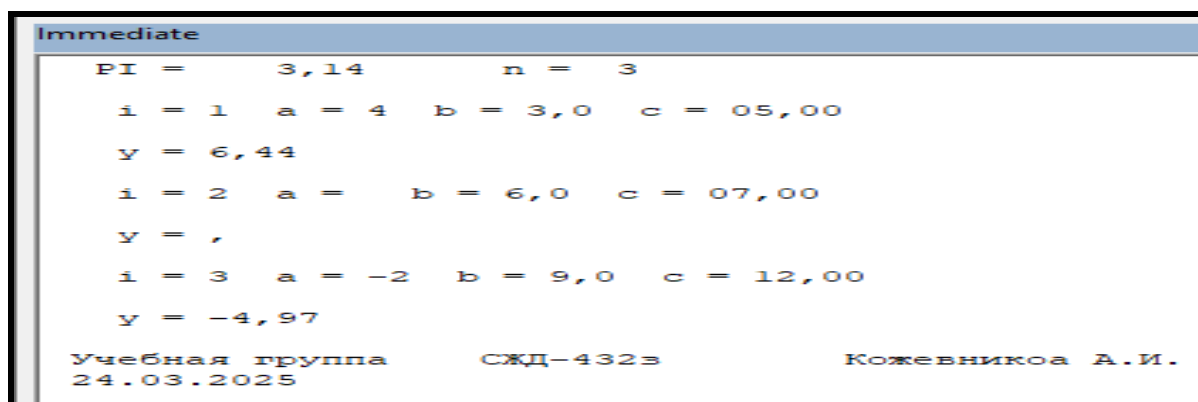
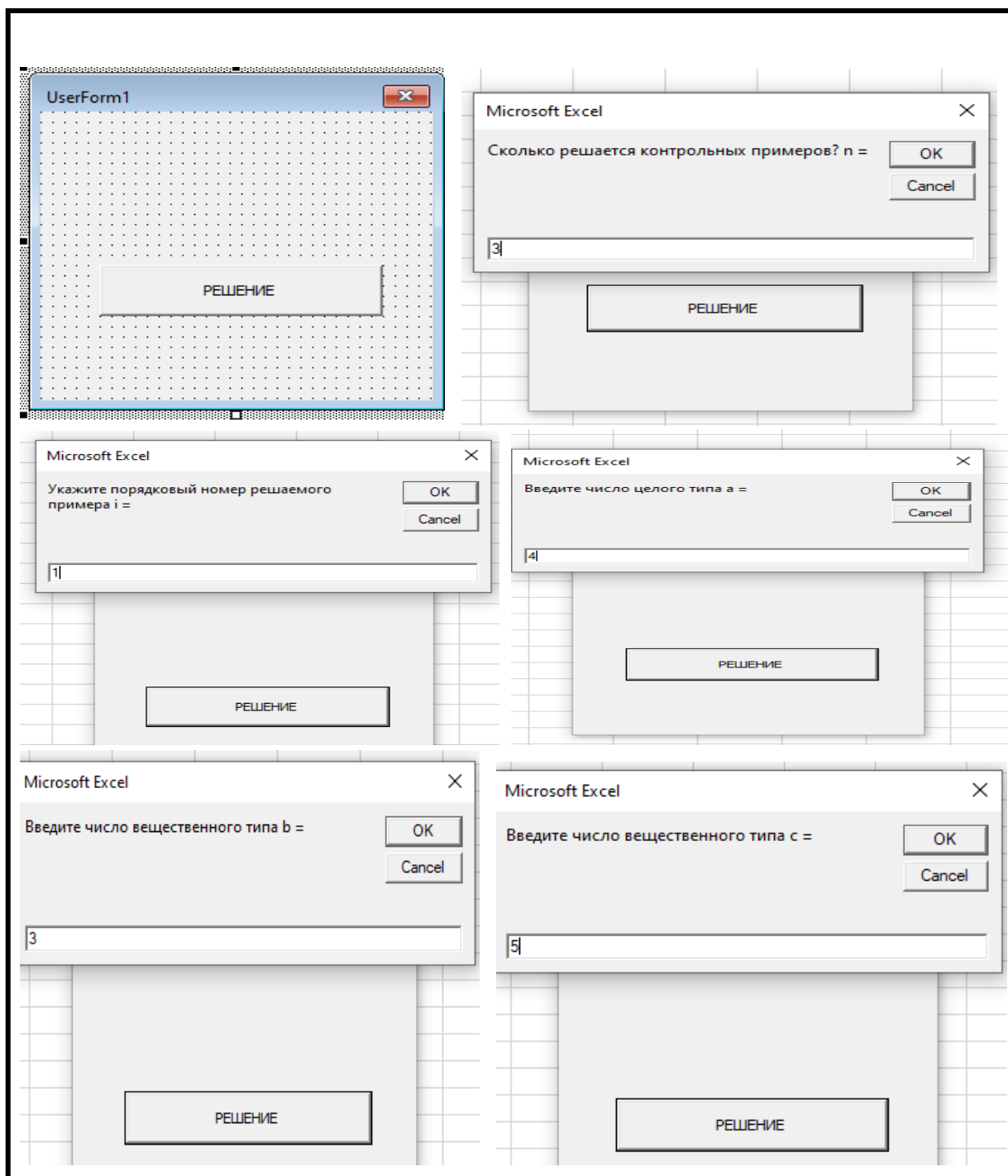


Рисунок 11 - Результаты решения на ПК

Решение примера в Python

```
1  import math
2  PI = 3.14
3  n = int(input('Сколько решается контрольных примеров? n = '))
4  for pp in range(n):
5      i = int(input('Укажите порядковый номер решаемого примера i = '))
6      a = int(input('Введите число целого типа a = '))
7      b = float(input('Введите число вещественного типа b = '))
8      c = float(input('Введите число вещественного типа c = '))
9      if i == n:
10         w1 = input('Укажите учебную группу ')
11         w2 = input('Укажите инициалы имени и отчества, фамилию ')
12         data = input('Укажите дату тремя парами арабских цифр ')
13     if a == -1:
14         y = a + b + c ** 2
15     elif a == 0:
16         y = a * math.sin(c + PI)
17     elif a == 1:
18         y = math.sqrt(a + b * c)
19     else:
20         y = a * math.log(c)
21     if i == 1:
22         print (' PI =', PI, '      n =', n)
23     print()
24     print (' i =', i, ' a =', a, ' b =', b, ' c =', c)
25     print ()
26     print (' y =',y)
27 print ('Учебная группа ', w1, '      ', w2)
28 print (data)
```

Сколько решается контрольных примеров? $n = 3$

Укажите порядковый номер решаемого примера $i = 1$

Введите число целого типа $a = 4$

Введите число вещественного типа $b = 3$

Введите число вещественного типа $c = 5$

$PI = 3.14$ $n = 3$

$i = 1$ $a = 4$ $b = 3.0$ $c = 5.0$

$y = 6.437751649736401$

Укажите порядковый номер решаемого примера $i = 2$

Введите число целого типа $a = 0$

Введите число вещественного типа $b = 6$

Введите число вещественного типа $c = 7$

$i = 2$ $a = 0$ $b = 6.0$ $c = 7.0$

$y = -0.0$

Укажите порядковый номер решаемого примера $i = 3$

Введите число целого типа $a = -2$

Введите число вещественного типа $b = 9$

Введите число вещественного типа $c = 12$

Укажите учебную группу *СЖД-432з*

Укажите инициалы имени и отчества, фамилию *Кожевников А.И.*

Укажите дату тремя парами арабских цифр *24.03.2025*

$i = 3$ $a = -2$ $b = 9.0$ $c = 12.0$

$y = -4.969813299576001$

Учебная группа *СЖД-432з*

Кожевников А.И.

24.03.2025

5.3 Пример 3. Подготовка к решению задачи со структурой ЦИКЛ

1. ОПИСАНИЕ ПОСТАНОВКИ ЗАДАЧИ

1.1. **Характеристика задачи.** Вычислить значения функции с точностью до сотых по формуле

$$V = 0,36\pi d^2 h$$

где постоянная $\pi = 3,14$; переменные: h – целого типа; d – вещественного типа.

При решении контрольного примера входным данным присвоить значения: $d_1 = 10$; $d_k = 11$; $\Delta d = 0,5$; $h_1 = 12$; $h_k = 14$; $\Delta h = 1$.

1.2. Выходные данные:

- а) перечень выходных данных: $\pi, d_1, d_k, \Delta d, h_1, h_k, \Delta h, \{d, h, V\}$.
- б) выходной документ оформить в соответствии с макетом, представленным на рис. 9.8.

1.3. Входные данные:

- а) постоянные: π ;
- б) переменные: $d_1, d_k, \Delta d, h_1, h_k, \Delta h, W_1, W_2, D_{ATA}$

1.3. Входные данные:

- а) постоянные: π ;
- б) переменные: $d_1, d_k, \Delta d, h_1, h_k, \Delta h, W_1, W_2, D_{ATA}$

2. ОПИСАНИЕ АЛГОРИТМА

2.1. Математическое описание. Модель задачи:

$$\pi = 3,14 \quad (1); \quad V = 0,36\pi d^2 \quad (2)$$

$$\mathbf{d} = \{d_1, d_1 + \Delta d, \dots, d_k\} \quad (3); \quad \mathbf{h} = \{h_1, h_1 + \Delta h, \dots, h_k\} \quad (4)$$

2.2. Выбор метода решения. Различные значения функции V определяются по одной и той же формуле, то есть вычисления выполняются путем повторения группы арифметических действий. Этим обосновывается выбор метода простой итерации.

2.3. Алгоритм решения. Приведён на рис. 12 в виде схемы.

Структура ЦИКЛ С ПРЕДУСЛОВИЕ, в которую вложена другая структура ЦИКЛ аналогичного вида (ЦИКЛ в ЦИКЛЕ). Каждый цикл имеет три заголовочных символа: во внешнем цикле ими являются символы 5, 6 и 16, а во внутреннем 10, 11 и 14. В заголовочных символах указаны заданные

величины соответственно: начальные значения параметров циклов в символах 5 и 10, конечные значения параметров циклов в символах 6 и 11, шаги параметров циклов в символах 16 и 14. Телом внешнего цикла являются символы 8, 10–14, внутреннего цикла - символы 12 и 13.

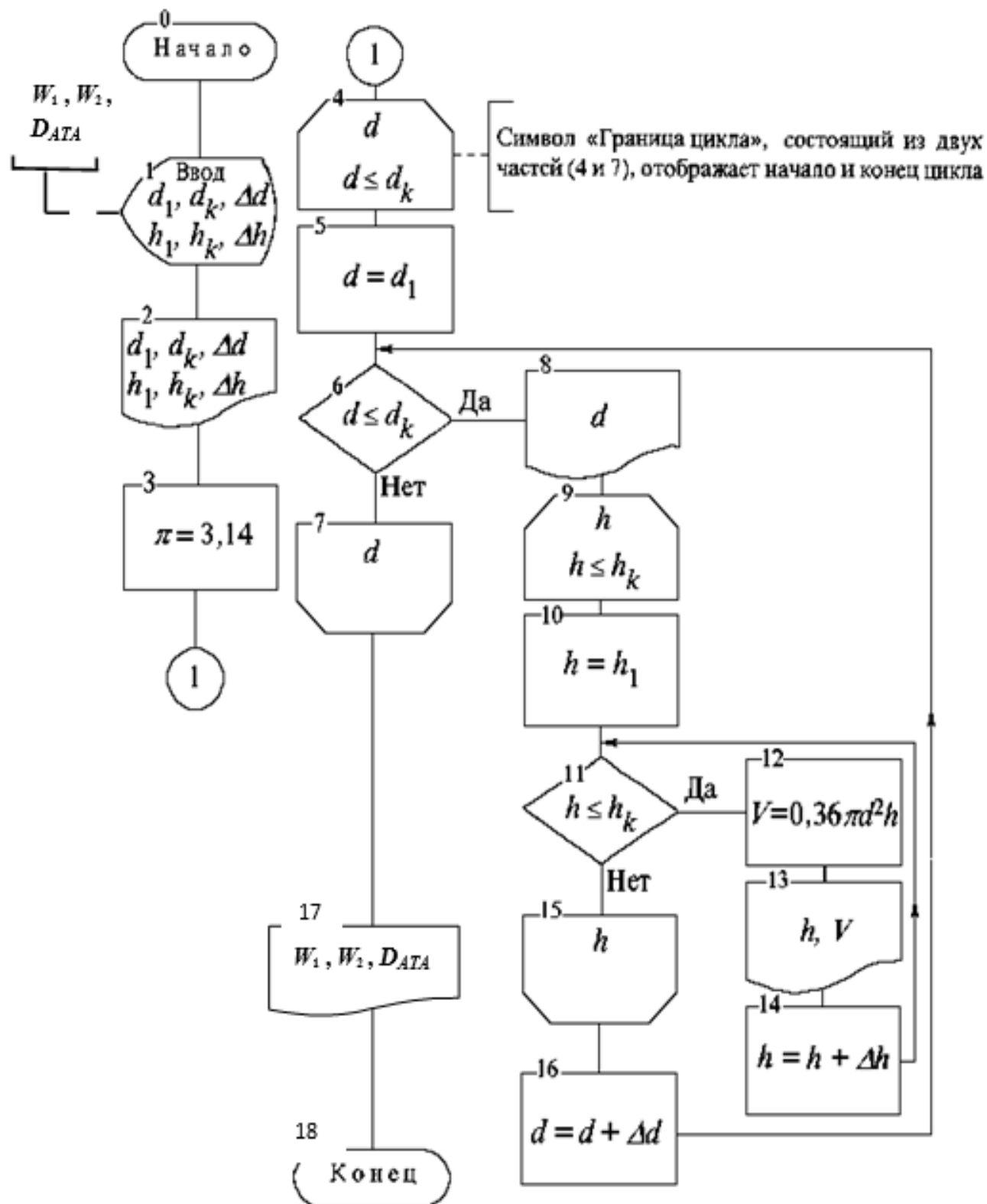


Рис. 12 - Схема алгоритма

4. ТЕКСТ ПРОГРАММЫ

Программный код

```

Private Sub CommandButton1_Click()
'Структура ЦИКЛ
'Задание N 14. Лабораторная работа N 4
Dim h1 As Integer, hk As Integer, dh As Integer
Dim d1 As Single, dk As Single, Dd As Single, V As Single, x1 As String
Dim x2 As String, x3 As String, x4 As String, x5 As String, x6 As String
Dim W1 As String, W2 As String, Data As String
d1 = InputBox("Введите вещественное значение (начальное значение) d1 = ")
dk = InputBox("Введите вещественное значение (конечное значение) dk = ")
Dd = InputBox("Введите вещественное значение (шаг) Dd = ")
h1 = InputBox("Введите целое значение (начальное значение) h1 = ")
hk = InputBox("Введите целое вещественное значение (конечное значение) hk = ")
dh = InputBox("Введите целое вещественное значение (шаг) Dh = ")
W1 = InputBox("Укажите учебную группу")
W2 = InputBox("Укажите инициалы имени и отчества, фамилию")
Data = InputBox("Укажите дату тремя парами арабских цифр")
Const PI As Single = 3.14
x1 = "3.5. Результаты решения контрольного примера на ПК"
x2 = "Значения данных:"          x3 = "-исходных постоянных:"
x4 = "-исходных переменных:"      x5 = "-искомых:"
Debug.Print Tab(6); x1:           Debug.Print
Debug.Print Tab(6); x2: Debug.Print x3; " PI = "; PI
Debug.Print x4; "    d1 = "; d1; "    dk = "; dk; "    Dd = "; Dd
Debug.Print Tab(25); "    h1 = "; h1; "    hk = "; hk; "    Dh = "; dh
Debug.Print x5;:                 Debug.Print Tab(25); "d        h        V "
    For d = d1 To dk Step Dd:      'Заголовок внешнего цикла
        Debug.Print Tab(24); Format(d, "00.0");
        For h = h1 To hk Step dh:  'Заголовок внутреннего цикла
            V = 0.36 * PI * d ^ 2 * h
            Debug.Print Tab(30); Format(h, "##");
            Debug.Print Tab(37); Format(V, "#####.##")
        Next h                    'Конец внутреннего цикла

```

Immediate

```

Значения данных:
-исходных постоянных: PI = 3,14
-исходных переменных: d1 = 10    dk = 11    Dd = 0,5
                      h1 = 12    hk = 14    Dh = 1
-искомых:             d        h        V
                      10,0    12    1356,48
                      13    1469,52
                      14    1582,56
                      10,5    12    1495,52
                      13    1620,15
                      14    1744,77
                      11,0    12    1641,34
                      13    1778,12
                      14    1914,9

```

Исполнил студент СЖД-423з
24.03.2025

Дергачев А.И.

Рисунок 12 - Результаты решения на ПК

Решение примера в Python

```
1  # Структура ЦИКЛ
2  # Задание N 14. Лабораторная работа N 4
3  PI = 3.14
4  d1 = float(input('Введите вещественное значение (начальное значение) d1 = '))
5  dk = float(input('Введите вещественное значение (конечное значение) dk = '))
6  dd = float(input('Введите вещественное значение (шаг) dd = '))
7  h1 = int(input('Введите целое значение (начальное значение) h1 = '))
8  hk = int(input('Введите целое значение (конечное значение) hk = '))
9  dh = int(input('Введите целое значение (шаг) dh = '))
10 w1 = input('Укажите учебную группу ')
11 w2 = input('Укажите инициалы имени и отчества, фамилию ')
12 data = input('Укажите дату тремя парами арабских цифр ')
13 print('3.5. Результаты решения контрольного примера на ПК')
14 print('Значения данных:')
15 print('    -исходных постоянных:')
16 print('        PI =', PI)
17 print('    -исходных переменных:')
18 print('        d1 =', d1, '    dk =', dk, '    dd =', dd)
19 print('        h1 =', h1, '    hk =', hk, '    dh =', dh)
20 print('    -искомых:')
21 print('        d            h            v')
22 d = d1
23 while d <= dk:
24     h = h1
25     while h <= hk:
26         v = 0.36 * PI * d**2 * h
27         print('        ', d, '            ', h, '            ', round(v, 2))
28         h = round(h + dh, 7)
29     d = round(d + dd, 7)
30 print('Исполнил студент ', w1, '            ', w2)
31 print(data)
```

Введите вещественное значение (начальное значение) $d1 = 10$

Введите вещественное значение (конечное значение) $dk = 11$

Введите вещественное значение (шаг) $dd = 0.5$

Введите целое значение (начальное значение) $h1 = 12$

Введите целое значение (конечное значение) $hk = 14$

Введите целое значение (шаг) $dh = 1$

Укажите учебную группу *СЖД-423з*

Укажите инициалы имени и отчества, фамилию *Дергачёв А.И.*

Укажите дату тремя парами арабских цифр *24.03.2025*

3.5. Результаты решения контрольного примера на ПК

Значения данных:

-исходных постоянных:

$$PI = 3.14$$

-исходных переменных:

$$d1 = 10.0 \quad dk = 11.0 \quad dd = 0.5$$

$$h1 = 12 \quad hk = 14 \quad dh = 1$$

-искомых:

d	h	v
10.0	12	1356.48
10.0	13	1469.52
10.0	14	1582.56
10.5	12	1495.52
10.5	13	1620.15
10.5	14	1744.77
11.0	12	1641.34
11.0	13	1778.12
11.0	14	1914.9

Исполнил студент *СЖД-423з*

Дергачёв А.И.

24.03.2025

В ы в о д . Результаты решения контрольных примеров, выполненные на калькуляторе и на ПК совпали. Следовательно, программа отлажена, схема алгоритма и текст программы составлены правильно (только после этого распечатать «Результаты решения контрольных примеров на ПК» и «4.Программный код»).

6 ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ PYTHON

7 Основные понятия

7.1 Ключевые слова

Ключевое слово – это зарезервированное слово или символ, которое интерпретатор Python распознаёт как элемент языка.

Примеры:

- Условные операторы: if, else, elif
- Циклы: for, while, break, continue
- Определение функций: def, return, lambda
- Обработка исключений: try, except, finally
- Логические значения: True, False
- Работа с классами: class, self

Полный список ключевых слов можно получить с помощью:

```
import keyword  
print(keyword.kwlist)
```

7.2 Литералы

Литерал – это константное значение, которое определяет данные определённого типа.

Примеры:

- Целое число (int): 150, -42, 0x1F (шестнадцатеричное)
- Дробное число (float): 3.14, -0.001, 2.5e3 (экспоненциальная запись)
- Строка (str): "Привет", 'Python';, ""Многострочный текст""
- Логическое значение (bool): True, False
- Список (list): [1, 2, 3], ["a", "b", "c"]
- Множество (set): {1, 2, 3}, {"a", "b"}
- Словарь (dict): {"name": "Alice", "age": 30}

7.3 Переменные

Переменная – это именованная область памяти, предназначенная для хранения данных, которые могут изменяться в процессе выполнения программы.

Правила именования переменных:

- Могут содержать буквы (латиница, кириллица), цифры и символ _.
- Не могут начинаться с цифры.
- Регистр букв имеет значение (var ≠ Var ≠ VAR).

- Нельзя использовать ключевые слова (if, for, while и т. д.).

Пример объявления переменных:

```
x = 10          # int
y = 3.14        # float
name = "Alice"  # str
is_active = True # bool
numbers = [1, 2, 3] # list
```

7.4 Алфавит языка Python

Алфавит Python включает:

1. Латинские буквы: A-Z, a-z
2. Кириллические буквы: А-Я, а-я (используются в строках и комментариях)
3. Цифры: 0-9
4. Специальные символы:
 - Арифметические операции: +, -, *, /, //, %, **
 - Операции сравнения: ==, !=, <, >, <=, >=
 - Логические операции: and, or, not
 - Разделители: ,, ., :, ;, (), [], {}, _, ';, "
 - Пробел () и табуляция (\t)

7.5 Синтаксис и семантика

- Синтаксис – правила записи конструкций языка.
- Пример синтаксической ошибки: if x = 10: (нужно ==).
- Семантика – смысл написанного кода.
- Пример семантической ошибки:

```
x = y + 1    # y не определён
```

Правильно:

```
y=3
x = y + 1
```

8 Типы данных в Python

Тип данных	Описание	Пример
int	Целое число произвольной длины	42, -100, 0xFF
float	Число с плавающей	3.14, -0.001, 2.5e3

	точкой	
str	Строка	"Hello", 'Python'
bool	Логическое значение	True, False
list	Изменяемый список	[1, 2, 3], ["a", "b"]
tuple	Неизменяемый кортеж	(1, 2, 3), ("x", "y")
set	Множество (уникальные элементы)	{1, 2, 3}
dict	Словарь (ключ: значение)	{"name": "Alice", "age": 30}
bytes	Неизменяемая последовательность байтов	b'ABC'
NoneType	Отсутствие значения	None

9 Объявление данных

9.1 Объявление переменных

```
x = 10      # int
y = 3.14    # float
name = "Alice"  # str
is_active = True  # bool
```

9.2 Изменения типа данных

```
a = int("10")    # str → int
b = float("3.14") # str → float
c = str(100)     # int → str
```

10 Операции

10.1 Арифметические операции

Операция	Пример
----------	--------

+	(сложение)	$5 + 3 \rightarrow 8$
-	(вычитание)	$10 - 4 \rightarrow 6$
*	(умножение)	$2 * 3 \rightarrow 6$
/	(деление)	$10 / 3 \rightarrow 3.333...$
//	(целочисленное деление)	$10 // 3 \rightarrow 3$
%	(остаток от деления)	$10 \% 3 \rightarrow 1$
**	(возведение в степень)	$2 ** 3 \rightarrow 8$
abs()	(модуль)	$abs(-4) \rightarrow 4$

10.2 Операции сравнения

Операция	Пример
== (равно)	$5 == 5 \rightarrow \text{True}$
!= (не равно)	$5 != 3 \rightarrow \text{True}$
< (меньше)	$3 < 5 \rightarrow \text{True}$
> (больше)	$5 > 3 \rightarrow \text{True}$
<= (меньше или равно)	$5 <= 5 \rightarrow \text{True}$
>= (больше или равно)	$5 >= 3 \rightarrow \text{True}$

10.3 Логические операции

Операция	Пример
and (И)	True and False → False
or (ИЛИ)	True or False → True
not (НЕ)	not True → False

Таблица 6 - Основные стандартные математические функции и функции обработки данных

Обозначение функции		Пояснение
в математике	в Python	
		При использовании математических функций подключается модуль <code>math</code> . Для этого в начале программы добавляется строка: <code>import math</code>
$\sin x$	<code>math.sin(x)</code>	x — задаваемый угол в радианах. - Для преобразования градусов в радианы следует число градусов умножить на выражение $\pi/180^\circ$.
$\cos x$	<code>math.cos(x)</code>	
$\tan x$	<code>math.tan(x)</code>	
$\arctg x$	<code>math.atan(x)</code>	При определении арктангенса значение угла допустимо задавать в диапазоне от $-\pi/2$ до $+\pi/2$.
$\arcsin x$	<code>math.asin(x)</code>	
$\arccos x$	<code>math.acos(x)</code>	
$\ln N$	<code>math.log(N)</code>	Натуральный логарифм N ($\log_e N$)
$\lg N$	<code>math.log10(N)</code>	Десятичный логарифм N ($\log_{10} N$)
$\log_a X$	<code>math.log(X,y)</code>	Аргумент N функции LOG должен быть $N > 0$.
e^m	<code>math.exp(m)</code>	
a^b	<code>math.pow(a,b)</code>	Также допустим вариант записи: $a**b$
$ a $	<code>math.fabs(a)</code>	Абсолютное значение величины a
$\sqrt{a}$	<code>math.sqrt(a)</code>	Аргумент a функции Sqrt должен быть $a \geq 0$.
$[x]$	<code>int(a)</code>	Выдаёт число, округлённое до меньшего целого.

Примеры записи математических выражений на языке Python

$$\gamma = \frac{\lg \frac{a}{b} \cdot \sqrt{\sin x + \cos x^2}}{|a^b|} \cdot 10^{-5}$$

```
gamma=  
math.log10(a/b)*math.sqrt(math.sin(x)+math.cos(x**2))/math.fabs(a**b)*10**-5
```

$$\beta = e^{a+b} \cdot \sqrt[3]{\arctg^2 a} \cdot \frac{1 + \frac{\gamma}{2}}{3 \cdot \tg \frac{l}{2}} \cdot 1,5 \cdot 10^6$$

```
beta=  
math.exp(a+b)*math.arctan(a)**2**(1/3)*(1+gamma/2)/(3*math.tan(l/2))*1.5*10**6
```

11 Ввод и вывод данных

11.1 Ввод данных

Ввод данных работает как присвоение переменной значения, введенного в определённом формате.

В Python код:

```
a=1  
b=2  
a=b
```

Означает, что мы присвоили переменной a значение переменной b , следовательно после выполнения кода $a=2$.

Подобным образом работает ввод значений через `input()`. Мы присваиваем переменной значение которое мы вводим на клавиатуре при запуске программы.

```
a= input()      # str  
b= int(input()) # int
```

11.2 Вывод данных

Вывод данных осуществляется через `print()`

Если написать:

```
print(a)
```

то программа выведет значение переменной a

Если же написать:

```
print('a')
```

то программа выведет a считая её строкой, а не переменной

Чтобы вывести значения через пробел в одну строчку, следует записать их через запятую, выводить можно значения разного типа

К примеру:

```
a=1
```

```
b='ab'
```

```
print(b, 'cd', a)
```

То программа выведет:

ab cd 1 #ad - это значение переменной, 'cd' - строка, которую мы ввели при выводе, 1 - значение переменной a

Чтобы вывести значения в разных строчках надо написать print несколько раз.

```
print('a')
```

```
print('b')
```

```
print('c')
```

Программа выведет:

```
a
```

```
b
```

```
c
```

#a, b, c - являются строками, введенными в print(), если не использовать кавычки то они считались бы переменными

12 Условные операторы

12.1 if-elif-else

Структура проверки:

Тип проверки(if/elif/else) условие на которое делается проверка : код который необходимо выполнить если проверка верна

if x > 0: #if это первичная проверка условия, выполняется всегда вне зависимости

```
print("Положительное")
```

elif x < 0: #elif это вторичная проверка условия, выполняется только если стоящая перед ним проверка if или elif дала отрицательный результат

```
print("Отрицательное")
```

else: #else если предыдущая проверка условия дала отрицательный результат, то воспроизводится код написанный в else

```
print("Ноль")
```

13 Циклы

Циклы сделаны для повторения кода указанного в них до того момента, пока условие продолжения цикла верно:

13.1 Цикл For

Цикл for выполняется так (названия переменных взяты из примера):

1. Сначала переменной *i* присваивается значение переменной *a*
2. Выполняется код помещенный в цикл(в коде можно использовать переменную *i*, и она будет меняться по мере продвижения цикла)
3. Переменной *i* присваивается значение ее суммы с переменной *c* (если значение переменной *i* было изменено при выполнении кода внутри цикла, эти изменения не будут учитываться)
4. Делается проверка: цикл остановится если *i* стало больше или равно переменной *b*
5. Если проверка выполняется и цикл не остановился, то все шаги за исключением первого повторяются

a=1

b=5

c=1

```
for i in range(a, b, c): #переменные a,b и c могут быть только типа int, а c
    не может быть равным 0
    print(i)
```

Программа выведет:

1 #Так как начальным числом цикла является переменная *a* = 1, то и переменная *i* =1

2 #далее *i* увеличивается на величину переменной *c* =1, то она увеличится на 1 => *i*=2

3 #по аналогии #*i*=4, если прибавить *c* = 1, то *i* = 5, а значит, что она перестанет быть меньше переменной *b*, поэтому цикл завершается

Особенности:

Если обозначить две переменные: for *i* in range(*a*, *b*) то переменные будут считаться числами начала и конца цикла , шаг будет равен 1.

Если обозначить только одну переменную for *i* in range(*a*), то программа будет считать началом цикла 0, концом переменную *a*, и цикл будет с шагом 1

13.2 Цикл While

Цикл while выполняется до тех пор, пока условие правдиво.

while условие:

код который выполняется в том случае если условие правдиво

Пример:

```
x = 0
while x < 5:
    print(x)
    x = x + 1
```

Программа выведет:

```
0 #так как print() находится перед x = x + 1, то x = 0
1 #так как при предыдущем цикле к x прибавилась 1, то теперь он равен 1
2
3
4
```

14 Рекурсия

Рекурсия это тип кода который можно вызывать при выполнении программы
def Название рекурсии(переменные над которыми будут проводиться операции внутри рекурсии):

```
    Код
    return переменная чье значение мы возвратим после
    выполнения рекурсии (return, так же завершает выполнение рекурсии)
    #Чтобы вызвать рекурсию следует написать её название и в скобках указать
    те переменные с которыми следует провести изменения
    Название рекурсии(переменные)
def F(x, y):
    b=x+y      #2 Рекурсия делает изменения с переменными
    return b   #3 в качестве результата, возвращает на место вызова
    рекурсии значение переменной b
x=1
y=2
a=F(x, y)     #1 Мы вызвали рекурсию и присвоили ее результат
переменной a
print(a)
Программа выведет:
3 # значение переменной a
```

15 Работа со строками и срез

В Python каждый символ в строке нумеруются в зависимости от его места, счёт начинается с 0.

Также помимо положительного номера, если и отрицательный, он обозначает порядковый номер с конца, последний символ в строке будет под отрицательным номером -1, предпоследний: -2 и тд

Срез:

С помощью срезов мы можем выделять их строка часть текста. для этого нужно написать название строки и в квадратных скобках указать, с какого по какой символ необходимо сделать срез.

Название строки[От кокого, до какого символа нужно сделать срез]

```
s = "Python"
print(s[0])          #программа выведет: 'P'
print(s[1:4])        #программа выведет: 'yth'
print(s[1:-1])       #программа выведет: 'ytho'
Если написать третий символ внутри скобок, то он будет шагом среза.
s = "121212"
print(s[::2])        #программа выведет: '111'
print(s[::-1])       #программа выведет: '212121'
```

С помощью len(название строки) можно узнать количество символов в строке

```
s = "Python"
print(len(s))        #программа выведет: 6
```

16 Массив

Массив позволяет хранить множество данных, нумерация данных такая же, как и у строк.

Чтобы создать массив, необходимо написать:

```
Название массива = []
```

Чтобы добавить значение в массив надо использовать команду:

```
Название массива.append(данные которые необходимо добавить в массив)
функция sort(), сортирует значения в числовом или алфавитном порядке
sort(Название массива)
```

Срезы работают так же, как и со строками

```
a=[]
print(a)             #программа ничего не выведет так как массив пустой
a.append(2)
a.append(1)
print(a)             #программа выведет: [2, 1]
a=sort(a)
print(a)             #программа выведет: [1, 2]
a.append(3)
a.append(4)
print(a[1:2])        #программа выведет: [2, 3]
```

17 Множество set()

Множества в отличие от массивов хранят только уникальные значения, в порядке числового или алфавитного порядка.

Создание множества:

```
Название множества = set()
```

```
a=set()
```

Добавление элемента в множество:

```
Название множества.add(Элемент который нужно добавить)
```

```
a=set()
a.add(2)
a.add(1)
a.add(1)
print(a)
```

Программа выведет:

{1, 2} #символ 1 хоть и был добавлен дважды, но так как это множество, а не массив, она будет записана только 1 раз

Разбиение строки на множество символов:

```
a=set('1231')
print(a)
```

Программа выведет:

{'1', '2', '3'} #в строке было 4 символа, 1 2 3 1, но 1 будет в множестве только 1 раз, не смотря на то что в строке она повторяется

18 Библиотека math (sin, cos...)

Основная библиотека для математических вычислений, в основном используется для вычисления sin, cos, tg и тд.

Для использования библиотеки её необходимо импортировать в начале кода, после импортирования библиотека будет работать на протяжении всего кода, повторный импорт не нужен

```
import math
```

Вычисление sin a:

```
import math
```

```
a=math.sin(a)
```

Вычисление cos, tg и тд. делается аналогичным способом.

19 Чтение текстового и табличного файла

Для демонстрации примеров используется файл “123.txt” с содержимым:

123 456

123456

12 34 56

19.1 Чтение файла open(‘’)

Чтобы импортировать файл в код сначала надо написать:

Название переменной которая будет отвечать за файл = open(‘название файла’)

```
a=open( “123 . txt” )
```

Если читаемый файл лежит не в одной папке с программой, то вместо названия следует написать весь путь к файлу.

19.2 Разбиение на строки readline()

Далее в большинстве задач необходимо чтение по строкам, для этого нужно использовать `readline()`. Эта функция позволяет разделять файл на строки и анализировать их последовательно, или добавлять в массив, для анализа после чтения.

Название переменной которой присваивается строка = название переменной, которая отвечает за `файл.readline()`.

```
a=open( "123 . txt")  
d=a.readline()
```

Программа выведет:

```
123 456      #первая строка
```

```
a=open( "123 . txt")
```

```
print(a)
```

```
for i in range(3):
```

```
    d=a.readline()
```

```
    print(d)
```

Программа выведет:

```
123 456      #первая строка
```

```
123456       #вторая строка
```

```
12 34 56     #третья строка
```

Таким образом, эта команда считывает первую строчку файла, чтобы считать 2, 3, 4... строчку, нужно повторить команду необходимое количество раз (чаще всего для этого удобнее использовать циклы)

19.3 Разбиение строки на массив `Split()`

Если в строке хранится несколько значений, то следует использовать `split()`, эта функция разделяет строку на массив, символ разделения вписывается внутрь скобок, если скобки пустые, разделительным знаком будет считаться пробел.

Название массива = название строки.`split(символ разделения)`

```
a=open( "123 . txt")
```

```
print(a)
```

```
for i in range(3):
```

```
    d=a.readline().split()
```

```
    print(d)
```

Программа выведет:

```
[&apos;123&apos;, &apos;456&apos;]      #первая строка файла с разделением  
через пробел
```

```
[&apos;123456&apos;] #вторая строка файла с разделением через пробел
```

```
[&apos;12&apos;, &apos;34&apos;, &apos;56&apos;] #третья строка файла с  
разделением через пробел
```

20 Список используемой литературы

1. ГОСТ 19.781–90. Обеспечение систем обработки информации программное. Термины и определения. Взамен ГОСТ 19.781–83 и ГОСТ 19.004–80; Введ. 01.01.92. – М.: Изд-во стандартов, 1990. – 14 с. ОКСТУ 4001. Группа Э00 СССР.
2. МЕТОДИКА ПОСТРОЕНИЯ СХЕМ АЛГОРИТМОВ ОСНОВНЫХ АЛГОРИТМИЧЕСКИХ СТРУКТУР *Дергачев А.И., Кожевников А.И., Петрова О.В., Тарбаева Е.А.* Нанотехнологии: наука и производство. 2024. № 4. С. 23–30.
3. Симонович и др. Информатика: Базовый курс: Учебное пособие СПб: Питер, 2021. – 638 с.
4. Заморин А.П., Марков А.С. Толковый словарь по вычислительной технике и программированию: Основные термины. / Под ред. проф. М. Р. Шура-Бура. – М.: Рус. яз., 1988. – 221 с.
5. ГОСТ 19.701–90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Взамен ГОСТ 19.002–80 и ГОСТ 19.003–80; Введ. 01.01.92. – М.: Изд-во стандартов, 1990. – 26 с. ОКСТУ 5004. Группа Т55 СССР..
6. Информатика: Учебное пособие. Для экономистов М.: ИНФРА-М 2012 - 410 с.
7. Грошев А.С. Информатика: Учебник для вузов/ А. С. Грошев – Архангельск, Арханг. гос. техн. ун-т, 2010, 484 с.
8. Булавский П. Е. Структурный подход к программированию : учеб. пособие по дисциплинам «Информатика» и «Информатика в экономике». Часть 1. Запись текстов программ на алгоритмическом языке Visual Basic / П. Е. Булавский, А. И. Дергачев, А. М. Перепеченов. – СПб. : ФГБОУ ВО ПГУПС, 2017.
9. Булавский П. Е. Сборник заданий и макеты форм отчетов по выполнению лабораторных работ : учеб.-метод. пособие по дисциплинам «Информатика» и «Информатика в экономике» для студентов заочной формы обучения / П. Е. Булавский, А. И. Дергачев, А. М. Перепеченов. – СПб. : ФГБОУ ВО ПГУПС, 2017.
10. Дергачев А. И. Сборник учебно-методических материалов и контрольных решений для проведения занятий со студентами университета всех специальностей по дисциплине «Информатика» / А. И. Дергачев, В. П. Андреев, Н. В. Байдина, Н. Ф. Костяно, А. М. Перепеченов. – URL : http://library.pgups.ru/elib/multim/2015/inform_01.zip.
11. Дергачев А. И., Дергачев С. А., Перепеченов А. М., Куранова О. Н., Петрова О. В. СТРУКТУРНЫЙ ПОДХОД К ПРОГРАММИРОВАНИЮ Часть 2 Запись текстов программ на алгоритмическом языке Visual Basic for Application Учебное пособие – СПб. : ФГБОУ ВО ПГУПС, 2020
12. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РЕШЕНИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ СИМПЛЕКС-МЕТОДА *Божко Л.М., Дергачев А.И., Дергачев С.А.* Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. 2024. № 3. С. 3–15

Лабораторная работа № 1

Текстовый процессор

1 Цель работы

Самостоятельное ознакомление с работой в текстовом процессоре MS Word.

2 Основные сведения из теории

Для выполнения лабораторной работы рекомендуется использовать Справку Word, а также любую литературу по этому текстовому редактору.

3 Порядок выполнения работы

Работа выполняется индивидуально в дисплейном классе на первом лабораторном занятии, начиная с пункта 2. Самостоятельно выполняется пункт 1 и оформляется отчет. На втором занятии предьявляется отчет в распечатанном виде и проходит защита лабораторной работы.

4 Содержание работы

Лабораторная работа состоит из четырех разделов, каждый из которых начинается с заголовка.

1. Работа с текстом

- 1.1. Подготовить реферат на одну полную страницу по теме варианта (первый абзац начать с Tab, между словами один пробел).
- 1.2. Внутри текста не должно быть заголовков и списков. В тексте должно быть не менее четырех абзацев.
- 1.3. На тексте реферата продемонстрировать приемы редактирования и форматирования фрагментов текста, выравнивание (по левому и правому краю, по центру и по ширине), смену шрифта, начертания и размера, изменение цвета текста, фона и стилей.
- 1.4. Показать отступы и интервалы перед и после абзаца, между строк.
- 1.5. В текст вставить сноску и ссылки на рисунки из раздела 4.
- 1.6. Вставку и удаление фрагментов текста показать на защите работы.

2. Работа с таблицами

- 2.1. Вставить таблицу, состоящую из 3 строк и 5 столбцов.
- 2.2. Первые две строки заполнить числами. Все числа выровнять по центру.
- 2.3. В третьей строке объединить первые две и три последние ячейки.
- 2.4. В левую ячейку ввести слово «Итого», а в правую – формулу для вычисления суммы всех чисел первой и второй строк.
- 2.5. Залить нижние ячейки разным цветом.
- 2.6. Внешние границы таблицы и двух нижних ячеек выделить жирным контуром.

3. Работа со списками.
 Все списки должны быть осмысленными, содержательными и относиться к теме реферата.
 - 3.1. Нумерованный список (1, 2, 3, 4, 5).
 - 3.2. Маркированный (*, *, *, *, *).
 - 3.3. Многоуровневый. Три уровня по 2-3 элемента списка в каждом уровне, например:
 1.
 - 1.1.
 - 1.1.1.
 - 1.1.2.
 - 1.1.3.
 - 1.2.
 - 1.2.1.
 - 1.2.2.
 2.
 - 2.1.
 - 2.1.1.
 - 2.1.2.
 - 2.2.
 - 2.2.1.
 - 2.2.2.
 - 2.2.3.
 3.
4. Работа с рисунком
 - 4.1. Вставить три одинаковых рисунка (рисунок должен быть иллюстрацией к реферату из раздела 1)
 - 4.2. У второго рисунка пропорционально изменить размеры (масштаб).
 - 4.3. Третий рисунок обрезать.
 - 4.4. Рисунки расположить один под другим.
 - 4.5. Подписать рисунки.
5. Разбить весь материал на 4 страницы по одному разделу на каждой.
6. В начало документа вставить две пустые страницы для титульного листа и оглавления.
7. Пронумеровать страницы.
8. Вставить верхний колонтитул (Фамилия И. О., группа, номер варианта, название лабораторной работы).
9. Собрать автоматически оглавление.
 - 9.1. Изменить стиль заголовков с «Обычный» на стиль «Заголовок 1».
 - 9.2. Разместить оглавление на второй странице.
10. На первой странице поместить титульный лист. Обратить внимание на то, что на титульном листе не должны отображаться верхний колонтитул и номер страницы.

5 Темы рефератов

1. История компании Microsoft.
2. История компании Apple.
3. Аппаратные средства персональных компьютеров.
4. Программные средства персональных компьютеров.
5. Операционные системы семейства Windows.
6. История развития вычислительной техники.
7. Системы счисления.
8. Социальные сети.
9. История сети Интернет.
10. Поисковые системы Интернета.
11. Электронные платежные системы.
12. Электронные деньги.
13. Смарт-карты.
14. Графические пакеты.
15. Языки программирования.
16. Беспроводные технологии.
17. Локальные сети.
18. Информатика как наука.
19. Информатизация общества
20. Телекоммуникационные средства в современном компьютерном мире.
21. Новые разработки ноутбуков.
22. Принтеры и их классификация. Новинки на рынке принтеров.
23. Флэш-память.
24. Программные средства ПЭВМ.
25. Особенности операционной системы Windows Mobile.
26. INTERNET – глобальная телекоммуникационная сеть.
27. Облачные хранилища.
28. Программа Skype.
29. Информационная безопасность в сетях ЭВМ.
30. Компьютерные вирусы.
31. Компьютерные преступления.
32. Криптография.
33. Защита баз данных.
34. Проблема искусственного интеллекта.
35. Новые информационные технологии на транспорте.

Допускается по согласованию с преподавателем выбрать тему самостоятельно.

ОФОРМЛЕНИЕ ОТЧЕТА

Отчет о лабораторной работе № 1 состоит из 6 страниц, полностью выполняется в текстовом процессоре (шрифт – Times New Roman, размер 12 пт), распечатывается и сшивается.

Лабораторная работа № 2**Структура Следование****1 Цель выполнения работы**

Изучить этапы подготовки задач к их решению на ЭВМ. Научиться применять стандартные блоки для отображения алгоритмов. Познакомиться с основными правилами разработки программ и их отладки.

2 Основные сведения из теории

Для выполнения лабораторной работы и ответов на контрольные вопросы рекомендуется использовать конспект лекций, методические материалы, представленные в ЭИОС, примеры и литературу, приведенные в настоящем учебном пособии.

3 Порядок выполнения работы

Подготовить задачу к решению на ЭВМ, выполнить постановку задачи, дать математическое описание, разработку алгоритма и программы. Вручную рассчитать контрольный пример по предложенным численным значениям входных данных и отладить программу.

4 Задания**Вариант 1**

Объем правильной четырехугольной пирамиды равен V . Боковые грани пирамиды наклонены к плоскости основания под углом φ . Определить полную поверхность пирамиды по формуле

$$S = 2\sqrt[3]{36V^2} \cdot \operatorname{ctg}^2 \varphi \cdot \frac{\cos^2 \varphi}{\cos^2 \varphi} \cdot \frac{2}{2}, \text{ если } V = 920 \text{ см}^3; \varphi = 0,76 \text{ рад.}$$

Вариант 2

Вычислить объем правильной усеченной пирамиды, если заданы ее высота h и площадь оснований S_1 и S_2 .

$$V = \frac{1}{3} h (S_1 + S_2 + \sqrt{S_1 \cdot S_2}), \text{ где } h = 15 \text{ см}, S_1 = 20 \text{ см}^2, S_2 = 10 \text{ см}^2.$$

Вариант 3

На высоте конуса, как на диаметре, описан шар. Найти объем части шара, заключенный внутри конуса, если высота конуса H , а угол при вершине его осевого сечения равен $2L$.

$$V = \frac{1}{6} \pi H^3 \sin^2 L (1 + \cos^2 L), \text{ если } H = 10 \text{ см}; L = 0,35 \text{ рад.}$$

Вариант 4

Плоскость, проходящая через центр нижнего основания цилиндра под углом α к основанию, пересекает верхнее основание по хорде, равной b и стягивающей дугу β . Вычислить объем цилиндра по формуле

$$V = \frac{\pi b^3 \operatorname{ctg} \frac{\beta}{2} \cdot \operatorname{tg} \alpha}{8 \sin^2 \frac{\beta^2}{2}}, \text{ если } b = 10 \text{ см; } \alpha = 15^\circ; \beta = 60^\circ.$$

Вариант 5

Объем правильной треугольной пирамиды равен V , угол наклона боковой грани к основанию пирамиды равен φ . Найти полную поверхность пирамиды по формуле

$$S = \sqrt[3]{36V^2 \cdot \operatorname{tg} \varphi \cdot \operatorname{ctg} \frac{\varphi}{3}}, \text{ если } V = 600 \text{ см}^3; \varphi = 0,75 \text{ рад.}$$

Вариант 6

Основание прямого параллелепипеда – ромб с острым углом φ и меньшей диагональю d . Найти объем параллелепипеда, если большая диагональ его составляет с плоскостью боковой грани угол β :

$$V = \frac{d^3 \operatorname{ctg}^3 \frac{\varphi}{2}}{2 \sin \beta} \sqrt{\sin \left(\frac{\varphi}{2} + \beta \right) \cdot \sin \left(\frac{\varphi}{2} - \beta \right)}, \text{ если } d = 10 \text{ см; } \varphi = 1,2 \text{ рад; } \beta = 0,5 \text{ рад.}$$

Вариант 7

Найти полную поверхность правильной треугольной пирамиды по данному объему V и углу α между боковой гранью и плоскостью основания:

$$S = \frac{2\sqrt{3} \cdot \cos^2 \alpha / 2}{\cos \alpha} \cdot \sqrt{9V^2 \operatorname{ctg}^2 \alpha}, \text{ при } V = 950 \text{ см}^3; \alpha = 0,7 \text{ рад.}$$

Вариант 8

Полная поверхность правильной четырехугольной пирамиды равна S , угол наклона боковой грани к плоскости основания равен ψ . Определить объем пирамиды по формуле

$$V = \frac{S \times \sin \psi}{24 \cos^3 \left(\frac{\psi}{2} \right)} \times \sqrt{2S \times \cos \psi}, \text{ если } S = 0,5 \text{ м}^3; \psi = 0,2 \text{ рад.}$$

Вариант 9

Две боковые грани треугольной пирамиды – прямоугольные равнобедренные треугольники, гипотенузы которых равны C и образуют между собой угол ω . Найти объем пирамиды по формуле

$$V = \frac{1}{6} C^3 \cdot \sin \frac{\omega}{2} \sqrt{\cos \omega}, \text{ если } C = 14 \text{ см; } \omega = 0,65 \text{ рад.}$$

Вариант 10

Вычислить поверхность и объем шарового сегмента, если заданы высота шарового сегмента h и радиус шара R по формулам:

$$S = 2\pi Rh, V = \pi h^2 \left(R - \frac{1}{3} h \right),$$

если $h = 18,5$ см; $R = 25$ см.

Вариант 11

Вычислить поверхность и объем шарового пояса, если заданы радиус шара R , высота шарового пояса h и радиусы основания шарового пояса r_1 и r_2 по формулам:

$$S = 2\pi Rh$$

$$V = \frac{1}{6} \pi h^3 + \frac{1}{2} \pi (r_1^2 + r_2^2) h,$$

где $h = 23,5$ см; $R = 25,5$ см; $r_1 = 4,5$ см; $r_2 = 6,8$ см.

Вариант 12

Найти полную поверхность правильной треугольной пирамиды, если известны ее объем V и угол α между боковой гранью и плоскостью основания:

$$S = \frac{2\sqrt{3} \cdot \cos^2 \frac{\alpha}{2}}{\cos \alpha} \cdot \sqrt[3]{9V^2 \operatorname{ctg}^2 \alpha}, \text{ если } V = 950 \text{ см}^3; \alpha = 0,7 \text{ рад.}$$

Вариант 13

Вычислить объем правильной треугольной пирамиды, если известен двугранный угол при боковом ребре φ и радиус R круга, описанного около одной из боковых граней:

$$V = \frac{R^2}{12} \times \frac{\cos \frac{\varphi}{2}}{\sin^6 \left(\frac{\varphi}{2} \right)} \times \left(3 \sin^2 \frac{\varphi}{2} - \cos^2 \frac{\varphi}{2} \right), \text{ если } R = 10 \text{ см; } \varphi = 65^\circ.$$

Вариант 14

Через две образующие конуса, составляющие угол α , проведена плоскость, образующая с плоскостью основания конуса угол β . Плоскость сечения равна p . Вычислить высоту конуса по формуле:

$$H = \sqrt{p \times \operatorname{ctg} \frac{\alpha}{2} \times \sin \beta}, \text{ если } p = 7,5 \text{ см}^2; \alpha = 30^\circ; \beta = 60^\circ.$$

Вариант 15

Около конуса описана пирамида. Основанием пирамиды служит прямоугольный треугольник, один из острых углов которого равен ψ . Определить объем пирамиды, если известно, что радиус основания конуса равен r и образующая наклонена к плоскости основания под углом β :

$$V = \frac{1}{3} r^3 \operatorname{tg} \beta \cdot \operatorname{ctg} \frac{\psi}{2} \cdot \operatorname{tg} \left(\frac{\pi}{4} + \frac{\psi}{2} \right), \text{ если } r = 5; \psi = 0,2 \text{ рад}; \beta = 0,8 \text{ рад}.$$

Вариант 16

По неподвижной наклонной плоскости, образующей угол α с горизонтом, начинает соскальзывать без трения тело массой m_1 . На расстоянии L от начала движения в него попадает тело массой m_2 , летящее горизонтально. При этом тела останавливаются. Определить скорость второго тела до удара по формуле

$$V = \frac{m_1 \sqrt{2gL \sin \alpha}}{m_2 \cos \alpha}, \text{ если } m_1 = 0,25 \text{ кг}; L = 1,2 \text{ м}; m_2 = 0,3 \text{ кг};$$

$$\alpha = \pi/6; g = 9,81 \text{ м/с}^2.$$

Вариант 17

Основание прямой призмы – ромб. Одна из диагоналей призмы равна L и составляет с плоскостью основания угол, равный α , а с одной из боковых граней угол, равный β . Найти объем призмы по формуле

$$V = \frac{L^3 \sin 2\alpha \cdot \sin \beta \cdot \cos \alpha}{4 \sqrt{\cos(\alpha + \beta) \cdot \cos(\alpha - \beta)}}, \text{ если } L = 14 \text{ см}; \alpha = 40^\circ; \beta = 30^\circ.$$

Вариант 18

В шар радиуса R вписан усеченный конус. Основания усеченного конуса отсекают от шара два сегмента с дугами в осевом сечении соответственно равными α и β . Найти боковую поверхность отсеченного конуса

$$S = 2\pi R^2 \sin \frac{\alpha + \beta}{2} \cos \frac{\alpha - \beta}{4}, \text{ если } \beta = 2,15 \text{ рад}; \alpha = 0,75 \text{ рад}; R = 15 \text{ см}.$$

Вариант 19

Грани параллелепипеда – ромбы, которые равны между собой и расположены так, что встречаются в одной из вершин три острых угла. Найти объем параллелепипеда по формуле

$$V = 2a^3 \sin \frac{\omega}{2} \sqrt{\sin \frac{3}{2}\omega \times \sin \frac{\omega}{2}}; \text{ если } a = 34,7 \text{ см; } \omega = 20^\circ.$$

Вариант 20

Шар радиусом r вписан в пирамиду, в основании которой лежит ромб с острым углом α . Боковые грани пирамиды наклонены к плоскости основания под углом φ . Найти объем пирамиды по формуле

$$V = \frac{4}{3} r^3 \frac{\operatorname{ctg}^3 \frac{\varphi}{2} \cdot \operatorname{tg} \varphi}{\sin \alpha}, \text{ если } r = 5 \text{ см; } \alpha = 0,6 \text{ рад; } \varphi = 1,4 \text{ рад.}$$

Вариант 21

Вычислить площадь правильного многоугольника, если известны число сторон многоугольника n и радиус описанного круга R по формуле:

$$S = \frac{1}{2} n R^2 \sin \frac{360^\circ}{n}, \text{ если } n = 8; R = 25,5 \text{ см.}$$

Вариант 22

Вычислить объем конуса, зная радиус r шара, вписанного в конус, и угол ω , под которым из центра видна образующая конуса:

$$V = \frac{1}{3} \pi r^3 \operatorname{tg}^3 \omega \operatorname{tg} 2\omega, \text{ если } r = 5 \text{ см; } \omega = 18^\circ.$$

Вариант 23

Вычислить объем правильной треугольной пирамиды, зная, что плоский угол при вершине равен α , а радиус окружности, описанной около боковой грани, равен R .

$$V = \frac{4}{3} R^3 \sin^2 \alpha \cdot \cos \frac{\alpha}{2} \sqrt{\sin \left(\frac{\pi}{3} + \frac{\alpha}{3} \right) \cdot \sin \left(\frac{\pi}{3} - \frac{\alpha}{3} \right)},$$

если $R = 17 \text{ см; } \alpha = 0,32 \text{ рад.}$

Вариант 24

Основание прямой призмы – прямоугольный треугольник с площадью S и острым углом φ . Площадь большей грани равна Q . Найти объем призмы по формуле

$$V = \frac{Q}{2} \sqrt{S \times \sin 2\varphi}, \text{ если } S = 35 \text{ см}^2; \varphi = 0,45 \text{ рад}, Q = 100 \text{ см}^2.$$

Вариант 25

Объем правильной треугольной призмы равен V , угол между диагоналями двух граней, проведенными из одной и той же вершины, равен φ . Найти длину стороны основания призмы по формуле

$$a = \sqrt{\frac{16 \sin^2 \frac{\varphi}{2} \cdot V^2}{3 \sin\left(\frac{\pi}{6} + \frac{\varphi}{2}\right) \cdot \sin\left(\frac{\pi}{6} - \frac{\varphi}{2}\right)}}, \text{ если } V = 150 \text{ см}^3; \varphi = 0,1 \text{ рад}.$$

Вариант 26

В конус с углом при вершине осевого сечения 2α вписан шар. Площадь большого круга шара равна S . Определить объем конуса по формуле:

$$V = \frac{1}{3} S \sqrt{\frac{S}{\pi}} \operatorname{ctg}^3 \left(\frac{\pi}{4} - \frac{\alpha}{2} \right) \operatorname{ctg} \alpha, \text{ если } S = 185 \text{ см}^2; \alpha = 12^\circ.$$

Вариант 27

В правильной четырехугольной пирамиде расстояние от центра основания до боковой грани равно d , угол между высотой пирамиды и боковой гранью равен ψ . Определить полную поверхность пирамиды по формуле

$$S = 2\pi d^2 \cdot \frac{\cos^2 \left(\frac{\pi}{4} + \frac{\psi}{2} \right)}{\sin \psi \cdot \cos^2 \psi}, \text{ если } d = 8 \text{ см}; \psi = 0,6 \text{ рад}.$$

Вариант 28

Шар радиусом r вписан в пирамиду, в основании которой лежит ромб с острым углом β . Боковые грани пирамиды наклонены к плоскости основания под углом φ . Найти объем пирамиды по формуле

$$V = \frac{4}{3} r^3 \cdot \frac{\operatorname{ctg}^3 \frac{\beta}{2} \cdot \operatorname{tg} \varphi}{\sin \beta}, \text{ если } r = 5 \text{ см}; \beta = 1,27 \text{ рад}; \varphi = 0,53 \text{ рад}.$$

Вариант 29

Полная поверхность конуса равна S . Образующая его наклонена к плоскости основания под углом λ . Вычислить объем конуса по формуле

$$V = \frac{1}{3} S \frac{\operatorname{tg} \frac{\lambda}{2} \sqrt{S \frac{\cos \lambda}{2\pi}}}{\cos \frac{\lambda}{2}}, \text{ если } S = 150 \text{ см}^2; \lambda = 0,55 \text{ рад.}$$

Вариант 30

В правильной пирамиде двугранный угол при основании равен γ . Боковая поверхность равна S . Найти расстояние от центра основания до боковой грани по формуле

$$r = \frac{\sin \gamma}{3} \sqrt{S \sqrt{3} \cos \gamma}, \text{ если } S = 100 \text{ см}^2; \gamma = 0,85 \text{ рад.}$$

Вариант 31

Вычислить площадь трапеции, если известны диагонали d_1 , d_2 и тупой угол между ними α по формуле

$$S = \frac{1}{2} d_1 d_2 \sin \alpha, \text{ если } d_1 = 45 \text{ см; } d_2 = 68 \text{ см; } \alpha = 125^\circ.$$

Вариант 32

Вычислить площадь правильного многоугольника, если известны число сторон многоугольника n и радиус вписанного круга r (апофема) по формуле

$$S = nr^2 \operatorname{tg} \frac{\pi}{n}, \text{ если } n = 25; r = 44,5 \text{ см.}$$

Вариант 33

Вычислить площадь треугольника, если известны длины его сторон a , b и c по формуле Герона

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где $p = \frac{a+b+c}{2}$, если $a = 12,5 \text{ см; } b = 15,5 \text{ см; } c = 22 \text{ см.}$

Вариант 34

Вычислить максимальную высоту подъема тела, брошенного под углом к горизонту, если известны угол броска α и начальная скорость V_0 по формуле

$$h = \frac{V_0^2 \sin \alpha}{2g}, \text{ если } V_0 = 15 \text{ м/с; } \alpha = 55^\circ.$$

Вариант 35

Вычислить дальность полета тела, брошенного под углом к горизонту, если известны угол броска α и начальная скорость V_0 по формуле

$$L = \frac{2V_0^2 \cos \alpha \sin \alpha}{g}, \text{ если } V_0 = 15 \text{ м/с}; \alpha = 45^\circ.$$

5 Содержание отчета

Отчет о лабораторной работе № 2 полностью оформляется в текстовом процессоре Word (шрифт – Times New Roman, размер 12 пт), распечатывается и сшивается степлером в левом верхнем углу.

В отчете нужно представить все основные этапы подготовки и решения задачи. Текст программы копируется в отчет после ее отладки. Результаты решения представляются в виде скриншотов и подтверждаются ручным расчетом контрольных примеров.

Лабораторная работа № 3

Структура Развилка

1 Цель работы

Изучить и сравнить особенности построения алгоритмов и разработки программ алгоритмических структур «Классическая развилка», «Модифицированная развилка», «Вложенная развилка» и «Развилка с одной ветвью».

2 Основные сведения из теории

Для выполнения лабораторной работы и ответа на контрольные вопросы рекомендуется использовать конспект лекций, методические материалы, представленные в ЭИОС, примеры и литературу, приведенные в настоящем учебном пособии.

3 Порядок выполнения работы

В каждом варианте задания необходимо:

- выполнить постановку задачи, определить требуемые входные и выходные данные для решения задач;
- разработать математические модели, схемы алгоритмов и программы;
- предусмотреть печать входных и выходных данных. Значения входных данных выбрать самостоятельно для каждой ветви задания;
- вручную просчитать контрольные примеры для каждой ветви алгоритма;
- отладить программы и оформить отчет.

4 Задания

Вариант 1

$$F = \begin{cases} \sin x, & \text{если } x > \pi. \\ e^x \sin x, & \text{если } x \leq \pi. \end{cases}$$

Вариант 2

Дано действительное число. Возвести его в квадрат, если оно положительно, и в четвертую степень, если отрицательное.

Вариант 3

$$Y = \begin{cases} \ln(1+x^2), & \text{если } x > 1. \\ \frac{1}{|x|}; & \text{если } x \leq 1. \end{cases}$$

Вариант 4

Даны две точки $A(x_1, y_1)$ и $B(x_2, y_2)$. Определить, которая из точек находится ближе к началу координат. При равных расстояниях ближней считать точку A .

Вариант 5

$$Z_1 = \begin{cases} 1 - e^{-m}, & \text{если } m \leq 3. \\ \sqrt{8} \ln|m|, & \text{если } m > 3, \end{cases} \quad \text{где } m = x^3.$$

Вариант 6

Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник.

Вариант 7

$$q = \begin{cases} k^{0,5}, & \text{если } k > 2,7. \\ 11,5\sqrt[3]{k^2}, & \text{если } k \leq 2,7, \end{cases} \quad \text{где } k = \sqrt{t}.$$

Вариант 8

Даны действительные числа x и y , не равные друг другу. Меньшее из этих двух чисел заменить половиной их суммы, а большее – их удвоенным произведением.

Вариант 9

$$C = \begin{cases} \sqrt{b^2 + 1}, & \text{если } b \leq 10 \\ \ln 10 \cdot \lg b, & \text{если } b > 10, \end{cases} \quad \text{где } b = qa.$$

Вариант 10

На плоскости XOY над осью X задана своими координатами точка A . Определить, в каком квадранте она расположена.

Вариант 11

$$Y = \begin{cases} c^{0,6} \cdot |d|, & \text{если } c > 5. \\ 12 \cdot \operatorname{tg} c, & \text{если } c \leq 5, \end{cases} \quad \text{где } c = q + 1.$$

Вариант 12

Даны неравные целые числа d и f . Заменить меньшее из них большим, а большее нулем.

Вариант 13

$$y = \begin{cases} (1-x)(1+x), & \text{если } x < 0. \\ \sqrt{3x+1} & \text{в остальных случаях.} \end{cases}$$

Вариант 14

Услуги телефонной сети оплачиваются по следующему правилу: за разговоры до T минут в месяц – A руб., а разговоры сверх установленной нормы оплачиваются из расчета B руб. за минуту. Определить плату за пользование телефоном для введенного времени X разговоров за месяц.

Вариант 15

$$F = \begin{cases} x \ln x, & \text{если } x > 5,2. \\ x^3, & \text{если } x \leq 5,2. \end{cases}$$

Вариант 16

Грузовой автомобиль выехал из одного города в другой со скоростью V_1 . Через время t в этом же направлении выехал легковой автомобиль со скоростью V_2 . Определить, догонит ли легковой автомобиль грузовой через время t_1 после своего выезда.

Вариант 17

$$Y = \begin{cases} \sin x, & \text{если } x < 0. \\ \ln(1+x^2), & \text{если } x \geq 0. \end{cases}$$

Вариант 18

Перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y – меньшее.

Вариант 19

$$R = \begin{cases} \lg a, & \text{если } a > 1,3. \\ 2,5a - 11 & \text{если } a \leq 1,3, \end{cases} \text{ где } a = d + 3,5c.$$

Вариант 20

Разработать модель пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60°C .

Вариант 21

$$z = \begin{cases} 2^c - 1, & \text{если } c > 4,5. \\ 1 + c^3, & \text{если } c \leq 4,5, \end{cases} \text{ где } c = 1,5\omega.$$

Вариант 22

Продукты расфасованы в два пакета. Масса первого – m_1 кг, второго – m_2 кг. Определить, какой пакет тяжелее – первый или второй и массу более тяжелого пакета.

Вариант 23

$$Y = \begin{cases} \frac{1}{x-1}, & \text{если } x > 1. \\ \sin x, & \text{если } x \leq 1. \end{cases}$$

Вариант 24

Определить, пройдет ли график функции $y = 2a \cdot x + bx + c$ через заданную точку с координатами (g, h) .

Вариант 25

$$F = \begin{cases} \sin x, & \text{если } x > 0. \\ \operatorname{tg}^{2/3} x, & \text{если } x \leq 0. \end{cases}$$

Вариант 26

Даны действительные числа a, b, c . Удвоить эти числа, если $a \geq b \geq c$, и заменить их абсолютными значениями, если это не так.

Вариант 27

$$Y = \begin{cases} x \sin x, & \text{если } x > \pi \text{ или } x < -2\pi. \\ \sin x, & \text{если } -2\pi \leq x \leq \pi. \end{cases}$$

Вариант 28

На оси OX расположены три точки a , b , c . Определить, какая из указанных точек расположена ближе к точке a .

Вариант 29

$$M = L + d,$$
$$\text{где } L = \begin{cases} \lg a, & \text{если } a > 4. \\ \sqrt[3]{a}, & \text{если } a \leq 4. \end{cases}$$

Вариант 30

Определить, не приводит ли суммирование двух целых чисел A и B к переполнению (т. е. к результату, большему чем заданная величина C). Если будет переполнение, то сообщить об этом; иначе вывести сумму этих чисел.

Вариант 31

$$Z = \begin{cases} \frac{1}{100 \operatorname{tg} x} + 4, & \text{если } x \leq 0 \\ \sqrt{2x} \sqrt[3]{3x}, & \text{если } x > 0 \end{cases}$$

Вариант 32

Определить, находится ли точка с координатами (x_0, y_0) внутри окружности радиусом R с центром в точке (x_1, y_1) .

Вариант 33

$$F = \begin{cases} \lg x, & \text{если } x > 1 \\ |x^2 - 1|, & \text{если } x \leq 1 \end{cases}$$

Вариант 34

Два отрезка AB и CD расположены на оси X и Y соответственно и заданы своими координатами. Определить, какой отрезок длиннее. Координаты задать самостоятельно.

Вариант 35

Определить, в каком квадранте находится значение функции $\sin X$, если $X > 0$.

5 Содержание отчета

Отчет по лабораторной работе № 3 полностью оформляется в текстовом процессоре Word (шрифт – Times New Roman, размер 12 пт), распечатывается и сшивается степлером в левом верхнем углу.

Отчет должен содержать все основные этапы подготовки и решения задач. Тексты программ копируются в отчет после их отладки. Результаты решения представляются в виде скриншотов и подтверждаются ручным расчетом контрольных примеров.

Лабораторная работа № 4**Структура Цикл****1 Цель работы**

Изучить и сравнить особенности построения алгоритмов и разработки программ алгоритмических структур «Цикл» и «Цикл в цикле»: «Итерационные циклы», «Циклы с параметром», «Цикл с предусловием», «Цикл с постусловием».

2 Основные сведения из теории

Для выполнения лабораторной работы и ответа на контрольные вопросы рекомендуется использовать конспект лекций, методические материалы, представленные в ЭИОС, примеры и литературу, приведенные в настоящем учебном пособии.

3 Порядок выполнения работы

В каждом варианте задания необходимо:

- выполнить постановку задачи, определить требуемые входные и выходные данные для решения предложенных задач;
- разработать математические модели, схемы алгоритмов и программы;
- предусмотреть печать входных и выходных данных в виде таблицы с шапкой;
- начальные и конечные значения параметров циклов и величины шага их изменения задать в качестве входных данных.
- вручную просчитать контрольные примеры для выбранных самостоятельно значений параметров цикла;
- отладить программы и оформить отчет.

4 Варианты задания**Вариант 1**

Железнодорожный состав проходит первую треть пути со скоростью V_1 , а оставшуюся часть пути – со скоростью V_2 . Определить скорость на первом участке пути по формуле

$$V_1 = \frac{V_{\text{ср}} - V_2}{3V_2 - 2V_{\text{ср}}},$$

если $V_2 = 50$ км/ч, а средняя скорость поезда на всем пути $V_{\text{ср}} = 37,5; 40; 45; 62,5; 74$ км/ч.

Вариант 2

Двигаясь с ускорением a , поезд достигает скорости $V_t = 60$ км/ч. За какое время эта скорость достигнута и какой путь будет пройден за это время?

Искомые величины получить для всех a , принимающих значения от 0,4 до 1,0 м/с² с шагом 0,1 м/с².

$$t = \frac{V_t}{a}; S = \frac{at^2}{2};$$

Вариант 3

За i -ю секунду от начала движения поезд прошел L метров. Какой путь пройдет поезд за первые t секунд и какой скорости, м/с², он достигнет по истечении этого времени?

$$S_t = at^2; V_t = at; a = \frac{L}{i},$$

Отладку программы произвести для значений $i = 4$, $t = 10$, $3 \leq L \leq 9$ с шагом 0,5 м.

Вариант 4

Найти скорость поезда, при которой маятник длиной r , подвешенный в вагоне, раскачивается особенно сильно, если длина рельсов $L = 12,5$ м; $g = 9,81$ м/с², по формуле

$$V = \frac{L}{2\pi} \sqrt{\frac{g}{r}},$$

где $40 \leq r \leq 80$ см с шагом 4 см.

Вариант 5

Участок пути длиной $S = 1,0$ км локомотив проходит с постоянным ускорением a . За какое время этот путь пройден и какова скорость в конце данного участка пути, если $0,2 \leq a \leq 1,2$ м/с² с шагом 0,2 м/с²?

$$t = \sqrt{\frac{2S}{a}}; V_t = at.$$

Вариант 6

Поезд массой m трогается с места и движется по горизонтальному пути под действием постоянной силы тяги локомотива F . Коэффициент сопротивления движению k . Определить ускорение поезда и скорость, достигнутую им через t секунд после начала движения, если

$$a = \frac{F - kmg}{m}; \quad V = at,$$

где $F = 4000 \text{ Н}; k = 0,005; t = 5 \text{ с}; g = 9,81 \text{ м/с}^2;$
 $2000 \leq m \leq 4000 \text{ т}$ с шагом 250 т.

Вариант 7

Поезд массой m , движущийся со скоростью V , остановился, пройдя после торможения путь S . Определить, как изменяется величина тормозной силы и время торможения в зависимости от скорости по формулам:

$$F = \frac{V^2 m}{2S}; \quad t = \frac{2S}{V},$$

где $m = 2000 \text{ т}; S = 550 \text{ м}; 30 \leq V \leq 60 \text{ с}$ шагом 5 км/ч.

Вариант 8

Как изменяется центростремительное ускорение поезда, движущегося по закруглению дороги со скоростью V в зависимости от радиуса r ?

$$a = \frac{V^2}{r},$$

где $V = 60 \text{ км/ч}; 200 \leq r \leq 1000 \text{ м}$ с шагом 100 м.

Вариант 9

Поезд, двигаясь под уклон, прошел за t секунд путь S и развил скорость V . Как изменяется ускорение поезда и какова была его скорость в начале уклона в зависимости от времени t ?

$$V_0 = \frac{2S}{t} - V; \quad a = \frac{V - V_0}{t},$$

где $S = 340 \text{ м}; V = 19 \text{ м/с}; 15 \leq t \leq 25 \text{ с}$ шагом 1 с.

Вариант 10

Расстояние между двумя станциями поезд прошел со средней скоростью $V_{\text{ср}}$ за t минут. Разгон и торможение вместе длились t_1 минут, а остальное время поезд двигался равномерно. Определить скорость V равномерного движения при заданных значениях времени t_1 .

$$V = \frac{2V_{\text{ср}} t}{2t - t_1},$$

где $V_{\text{ср}} = 72 \text{ км/ч}; t = 20 \text{ мин}; 2,5 \leq t_1 \leq 6,5 \text{ мин}$ с шагом 30 с.

Вариант 11

Электровоз трогает с места состав массой m . С каким ускорением движется поезд в зависимости от массы, если коэффициент сопротивления $\mu = 0,005$, а сила тяги $F_T = 400$ кН, $g = 9,8$ м/с²?

$$a = \frac{F_T - \mu mg}{m},$$

где $1500 \leq m \leq 2000$ т с шагом 50 т.

Вариант 12

Электропоезд в момент выключения тока имел скорость V . Какое время и расстояние пройдет он до полной остановки по горизонтальному пути при разных значениях скорости? Коэффициент сопротивления движению μ .

$$t = \frac{V}{\mu g}; l = \frac{V^2}{2\mu g},$$

где $\mu = 0,006$; $g = 9,81$ м/с²; $5 \leq V \leq 10$ м/с с шагом 0,5 м/с.

Вариант 13

Вагон массой m подходит к неподвижной платформе со скоростью V_1 и ударяет ее, после чего платформа получает скорость V . Скорость вагона после удара уменьшилась до V_2 . Вычислить значение массы платформы для ряда значений V : $0,1 \leq V \leq 1,5$ м/с с шагом 0,25 м/с

$$m_{пл} = \frac{V_1 - V_2}{V} m,$$

где $m = 60$ т; $V_1 = 0,2$ м/с; $V_2 = 0,1$ м/с.

Вариант 14

Определить массу состава, который может везти тепловоз с ускорением a при различных коэффициентах сопротивления μ , если он развивает максимальное тяговое усилие F_T ?

$$m = \frac{F_t}{a + \mu g},$$

где $a = 0,1$ м/с²; $F_t = 300$ кН; $g = 9,8$ м/с²;
 $0,001 \leq \mu \leq 0,01$ с шагом 0,001.

Вариант 15

Сколько вагонов может везти электровоз в гору с уклоном L , если коэффициент максимального трения покоя равен k_2 , а коэффициент трения качения k_1 . Вес электровоза в 4 раза больше вагона.

$$N = \frac{4[(k_2 - k_1) \cos L - \sin L]}{\sin L + k_1 \cos L}.$$

Проанализировать изменение функции для значений

$$0^\circ \leq L \leq 6^\circ \text{ с шагом } 0,5^\circ,$$

если $k_1 = 0,001$; $k_2 = 0,1$.

Вариант 16

Скорость истечения груза из горизонтального отверстия бункера равна:

$$V = 5,9\lambda \sqrt{\frac{F}{P} \sin L},$$

где λ – коэффициент истечения;

F – площадь поперечного сечения потока;

P – периметр сечения;

L – угол наклона желоба, отклоняющего поток и создающего подпор.

Отладить программу для значений: $\lambda = 0,6$; $F = 0,36 \text{ м}^2$; $P = 2,4 \text{ м}^2$; $30^\circ \leq L \leq 90^\circ$ с шагом 10° . Результаты напечатать в таблице.

Вариант 17

К пружине подвешен груз массой m . Пружина под влиянием силы F растягивается на величину x . Определить период вертикальных колебаний груза для разных F по формуле:

$$T = 2\pi \sqrt{\frac{mx}{F}}.$$

Отладить программу для следующих значений переменных: $m = 10 \text{ кг}$; $x = 0,15 \text{ м}$; $1,85 \leq F \leq 3,2 \text{ Н}$ с шагом $0,15 \text{ Н}$.

Вариант 18

Определить смещение точки, совершающей гармоническое колебание:

$$x = 5 \sin(7,8t + 1,25),$$

где $0 \leq t \leq 8$ с шагом $0,5 \text{ с}$.

Вариант 19

Координаты точки при переходе от общих осей координат к другим, наклоненным к первым под углом L , определяются по формулам:

$$x_1 = x \cos L + y \sin L; y_1 = -x \sin L + y \cos L.$$

Как будут меняться координаты x_1 и y_1 для точки $x = 2,7$; $y = 3,4$, если $0 \leq L \leq \frac{\pi}{2}$ с шагом $\frac{\pi}{18}$.

Вариант 20

Определить число зон пригородного пассажиропотока при составлении расписаний движения поездов по формуле:

$$Z = \Pi \frac{A \cdot \tau}{M},$$

где Π – общее число остановочных пунктов на участке;
 A – среднечасовой пассажиропоток на остановочном пункте;
 τ – время на разгон, замедление и стоянку поезда;
 M – расчетная населенность поезда.

Для отладки принять: $\Pi = 12$; $\tau = 0,5$ ч; $A = 3,0$ тыс. чел.

$$1000 \leq M \leq 2000 \text{ чел; } \Delta M = 100 \text{ чел.}$$

Вариант 21

Определить диаметр d и длину l цилиндрической стальной цапфы вала, рассматривая цапфу как балку, заделанную концом. Нагрузка P на квадратную единицу диаметрального сечения цапфы не должна превышать 30 кг/см^2 ; допускаемое напряжение $R = 800 \text{ кг/см}$; полная величина давления на цапфу: $20 \leq Q \leq 27 \text{ т}$ с шагом $0,5 \text{ т}$.

$$d = \sqrt[4]{\frac{32Q^2}{\pi R P}}; l = \frac{Q}{dP}.$$

Вариант 22

Найти расстояние между точками, совершающими гармонические колебания:

$$x_1 = 0,1 \cdot \sin 2t; x_2 = 1,7 \cdot \sin(0,8t - 0,42)$$

в момент времени $0,6 \leq t \leq 1,8$ с шагом $0,2$.

Вариант 23

Какова в зависимости от дальности поездки оптимальная для пассажиров длина перегона на пригородных участках движения поездов?

$$l_{\text{опт}} = \sqrt{\frac{L_{\text{ср}} \cdot V_{\text{пеш}}}{15 \cdot \beta}} \cdot t_{\text{ст}},$$

где $L_{\text{ср}}$ – средняя дальность поездки пассажира в пригородном сообщении;
 $V_{\text{пеш}}$ – средняя скорость передвижения пешеходов;
 $t_{\text{ст}}$ – стоянка поезда с учетом затрат времени на разгон и торможение.
 Отладку программы произвести для значений:
 $\beta = 1,5$; $V_{\text{пеш}} = 5$ км/ч; $t_{\text{ст}} = 1$ мин; $20 \leq L_{\text{ср}} \leq 40$ км с шагом 2,5 км.

Вариант 24

С расстояния d фотографируют поезд, движущийся со скоростью V .
 Определить для разных объективов время t экспозиции, за которое изображение сместилось бы не более чем на $S = 0,01$ мм. Фокусное расстояние объектива – F .

$$t = \frac{S(d - F)}{FV}.$$

Отладку программы выполнить для контрольного примера:
 $V = 72$ км/ч; $d = 100$ м; $F = 22$ мм, 37 мм, 50 мм, 80 мм, 140 мм.

Вариант 25

Отклонения при свободных затухающих колебаниях описываются формулой:

$$y = 3,7 \cdot e^{-1,2t} \sin(0,8t + 0,26).$$

Найти расстояние от начала координат до точек на этой кривой в момент времени $t = 0, 2, 4, 6, \dots, 24$ по формуле

$$Z = \sqrt{t^2 + y^2}.$$

Результаты решения представить в таблице.

Вариант 26

Какое количество условного топлива израсходуют двигатели тепловоза на расстоянии l при изменении скорости V , если средняя мощность его двигателя $P = 2000$ кВт, а КПД $\eta = 25$ %. Теплота сгорания условного топлива $g = 2,8 \cdot 10^7$ Дж/кг.

$$m = \frac{P \cdot l}{V \cdot g \cdot \eta}.$$

Отладить программу для значений:

$l = 100$ км; $50 \leq V \leq 120$ км/ч с шагом 10 км/ч.

Вариант 27

Маховик, вращаясь с постоянной угловой скоростью ω_0 был отключен от двигателей и, сделав m оборотов, остановился. Найти угловое ускорение маховика.

$$\varepsilon = \frac{\omega_0^2}{4\pi m}.$$

Отладить программу для значений:

$$\omega_0 = 650 \text{ рад/с}; 25 \leq m \leq 100 \text{ об. с шагом } 5 \text{ об.}$$

Вариант 28

Паровой молот массой m_1 падает с высоты h на стальную болванку массой m_2 . Сколько раз он должен упасть, чтобы температура болванки поднялась на Δt °С? На нагрев болванки идет 50 % теплоты, полученной при ударах. Удельная теплоемкость стали $C = 460$ Дж/кгН.

$$n = \frac{2C \cdot m_2 \cdot \Delta t}{m_1 \cdot g \cdot h},$$

где $g = 9,81$ м/с²; $h = 2,5$ м, $\Delta t = 40$ °С, $m_2 = 220$ кг.

$6 \leq m_1 \leq 10$ т с шагом 0,5 т.

Вариант 29

По прямому участку пути двигаются три вагона массой m_1 , m_2 , m_3 . Какое максимальное число столкновений между ними может произойти?

$$N_{\max} = - \left\lfloor \frac{\pi}{\arccos \sqrt{\frac{m_1 \cdot m_2}{(m_1 + m_2)(m_2 + m_3)}}} \right\rfloor ;$$

где $\arccos \alpha = \frac{\pi}{2} - \arctg \frac{\alpha}{\sqrt{1 - \alpha^2}} ;$

$\lfloor \rfloor$ [– целая часть числа;

$m_1 = 100$ т; $m_3 = 100$ т; $10 \leq m_2 \leq 120$ т с шагом 10 т.

Вариант 30

Груз массой m поднимается лебедкой с ускорением a . Найти работу, произведенную за первые t секунд от начала подъема:

$$A = m(g + a) \frac{at^2}{2}.$$

Для отладки программы принять: $m = 3$ т, $a = 2$ м/с², $1 \leq t \leq 2$ с шагом 0,1 с.

Вариант 31

При быстром торможении трамвай, имевший скорость V , начал двигаться «юзом». Определить расстояние, которое пройдет трамвай с момента торможения до полной остановки. Коэффициент трения между колесами и рельсами – k .

$$S = \frac{V^2}{2} kg,$$

Если $10 \leq V \leq 50$ км/ч с шагом 5 км/ч; $k = 0,2$; $g = 9,80665$ м/сек².

Вариант 32

В прямоугольной пирамиде двугранный угол при основании равен α . Определить наклон бокового ребра к плоскости основания пирамиды по формуле

$$\beta = \arctg\left(\frac{1}{2} \operatorname{tg} \alpha\right),$$

если $30^\circ \leq \alpha \leq 60^\circ$ с шагом 3° .

Результат напечатать в градусной мере.

Вариант 33

Найти радиус основания цилиндра, имеющего при данном объеме наименьшую поверхность:

$$R = \sqrt[3]{\frac{V}{2\pi}},$$

если $150 \leq V \leq 750$ см³ с шагом 50 см³.

Вариант 34

Исследовать поведение функции $y = 5x^2 - 3x - 10$ в диапазоне от -4 до 4 с шагом изменения аргумента в 0,5.

Вариант 35

Напечатать таблицу перевода мер длины из метров в сажени, футы и аршины от 1 до 10, если 1 сажень равна 2,1366 м, 1 фут равен 0,3048 м и 1 аршин равен 0,7112 м.

5 Содержание отчета

Отчет о лабораторной работе № 4 полностью оформляется в текстовом процессоре Word (шрифт – Times New Roman, размер 12 пт), распечатывается и сшивается степлером в левом верхнем углу.

Отчет должен содержать все основные этапы подготовки и решения задач. Тексты программ копируются в отчет после их отладки. Результаты решения представляются в виде скриншотов и подтверждаются ручным расчетом контрольных примеров для трех вариантов значений аргументов.

Контрольные вопросы к лабораторной работе № 2

Структура Следование

1. Этапы разработки информационных технологий.
2. Что включает постановка задачи?
3. Что такое алгоритм, схема алгоритма?
4. Стандартные блоки и линии связи для изображения алгоритма, нумерация блоков.
5. Вход в приложение Visual Basic.
6. Что собой представляет визуальная часть проекта, окно Form?
7. Основные элементы управления, панель General.
8. Размеры и основные свойства объектов управления Properties.
9. Что такое код приложения.
10. Из каких частей состоит проект в Visual Basic?
11. Как из формы перейти в редактор кода?
12. Как быстро перейти от формы к коду и обратно? Окно Project.
13. В чем заключается отладка проекта?
14. Как переносится в отчет о лабораторной работе код процедуры и результат?
15. Поясните каждую строку своей программы.
16. Что такое интегрированная среда разработки – IDE?
17. Основные понятия IDE.
18. Рабочая поверхность IDE или поле окна Visual Basic.
19. Основное окно.
20. Окно конструктора формы – Form.
21. Оператор присваивания.
22. Окно проекта – Project.
23. Окно свойств Properties.
24. Окно вида формы – Form Layout.
25. Панель элементов управления – General.
26. Порядок создания проекта в Visual Basic.
27. Сохранение проекта.
28. Запуск программы на выполнение и ее завершение.
29. Основы языка Visual Basic. Типы данных.
30. Ключевые слова.
31. Константы и переменные. Что общего и в чем разница?
32. Правила записи арифметических выражений, примеры.
33. Арифметические функции в языке Visual Basic.
34. Функции преобразования типов данных Val и Str.
35. Тригонометрические функции в языке Visual Basic.

Контрольные вопросы к лабораторной работе № 3

Структура Развилка

1. Организация ввода данных.
2. Функция ввода – `InputBox(<Приглашение>, <Заголовок>, [По умолчанию])`.
3. Функция `Format`.
4. Организация вывода данных.
5. Панель сообщений в режиме оператора
`MsgBox(<Сообщение>, [<Код1>+<Код2>], [Заголовок])`
6. Панель сообщений в режиме функции `<Имя переменной> = MsgBox(<Сообщение>, [<Код1>+<Код2>], [Заголовок])`
7. Печать результатов на форму – метод `Debug.Print`.
8. Использование в методе `Debug.Print` символов: кавычки, запятая, точка с запятой и &-конкатенация.
9. Использование в методе `Debug.Print` функций `Spс(число)`, `CurrentX=число` и `CurrentY=число`.
10. Операции отношения и их результат, примеры.
11. Логические операции и их результат, примеры.
12. Вычислительные процессы и структуры.
13. Что такое структурное программирование.
14. Структура РАЗВИЛКА и ее разновидности.
15. Классическая и модифицированная развилки и их реализация в языке Visual Basic.
16. Вложенная РАЗВИЛКА и ее реализация в языке Visual Basic.
17. РАЗВИЛКА с одной ветвью и ее запись на языке Visual Basic.
18. Оператор `Select Case . . . End Select`. Пример использования.
19. Оператор безусловного перехода - `Go To <метка>` . Пример использования.
20. Функция `ИФ` (логическое выражение, выражение1, выражение2). Пример использования.

Контрольные вопросы к лабораторной работе № 4

Структура Цикл

1. Что собой представляют циклический вычислительный процесс и структура цикла?
2. Что такое параметр цикла?
3. Что такое тело цикла?
4. Цикл с параметром.
5. Циклы с предусловием и с постусловием.
6. Оператор For ... Next, досрочный выход из цикла.
7. Какие переменные обязательно являются входными в циклических задачах?
8. Поле списка ListBox.
9. В каких случаях в операторе цикла к конечному значению параметра цикла прибавляют половину шага?
10. Как оформляют вывод результатов в виде таблицы с шапкой?
11. Операторы Do While ... Loop и Do ... Loop While.
12. Операторы Do Until ... Loop и Do ... Loop Until.
13. Как организуется цикл для заданного количества аргументов функции, не отличающихся постоянным шагом?
14. Структура цикл в цикле.
15. Итерационные циклы.
16. Итерационный цикл. Метод последовательных приближений.
17. Итерационный цикл. Метод деления отрезка пополам.

Приложение А

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ
ИМПЕРАТОРА АЛЕКСАНДРА I»
Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»
Дисциплина: «Информатика»

О Т Ч Е Т
о лабораторной работе № 1
на тему:
«ТЕКСТОВЫЙ ПРОЦЕССОР MS WORD»
Вариант 14

Выполнил студент
Факультета *БФО*
Группы *XXX-000-з*
Номер зачетной книжки
.....

Иванов И. И.

Санкт-Петербург
2025

Содержание работы

Лабораторная работа состоит из четырех разделов, каждый из которых начинается с заголовка.

11. Работа с текстом
 - 11.1. Подготовить реферат на одну полную страницу по теме варианта (первый абзац начать с Tab, между словами один пробел).
 - 11.2. Внутри текста не должно быть заголовков и списков. В тексте должно быть не менее четырех абзацев.
 - 11.3. На тексте реферата продемонстрировать приемы редактирования и форматирования фрагментов текста, выравнивание (по левому и правому краю, по центру и по ширине), смену шрифта, начертания и размера, изменение цвета текста, фона и стилей.
 - 11.4. Показать отступы и интервалы перед и после абзаца, между строк.
 - 11.5. В текст вставить сноску и ссылки на рисунки из раздела 4.
 - 11.6. Вставку и удаление фрагментов текста показать на защите работы.
12. Работа с таблицами
 - 12.1. Вставить таблицу, состоящую из 3 строк и 5 столбцов.
 - 12.2. Первые две строки заполнить числами. Все числа выравнивать по центру.
 - 12.3. В третьей строке объединить первые две и три последние ячейки.
 - 12.4. В левую ячейку ввести слово «Итого», а в правую – формулу для вычисления суммы всех чисел первой и второй строк.
 - 12.5. Залить нижние ячейки разным цветом.
 - 12.6. Внешние границы таблицы и двух нижних ячеек выделить жирным контуром.
13. Работа со списками.

Все списки должны быть осмысленными, содержательными и относиться к теме реферата.

 - 13.1. Нумерованный список (1, 2, 3, 4, 5).
 - 13.2. Маркированный (*, *, *, *, *).
 - 13.3. Многоуровневый. Три уровня по 2-3 элемента списка в каждом уровне, например:
 4.
 - 4.1.
 - 4.1.1.
 - 4.1.2.
 - 4.1.3.
 - 4.2.
 - 4.2.1.
 - 4.2.2.
 5.
 - 5.1.
 - 5.1.1.
 - 5.1.2.
 - 5.2.
 - 5.2.1.
 - 5.2.2.
 - 5.2.3.
 6.
14. Работа с рисунком
 - 14.1. Вставить три одинаковых рисунка (рисунок должен быть иллюстрацией к реферату из раздела 1)
 - 14.2. У второго рисунка пропорционально изменить размеры (масштаб).

- 14.3. Третий рисунок обрезать.
- 14.4. Рисунки расположить один под другим.
- 14.5. Подписать рисунки.
15. Разбить весь материал на 4 страницы по одному разделу на каждой.
16. В начало документа вставить две пустые страницы для титульного листа и оглавления.
17. Пронумеровать страницы.
18. Вставить верхний колонтитул (Фамилия И. О., группа, номер варианта, название лабораторной работы).
19. Собрать автоматически оглавление.
 - 19.1. Изменить стиль заголовков с «Обычный» на стиль «Заголовок 1».
 - 19.2. Разместить оглавление на второй странице.
20. На первой странице поместить титульный лист. Обратить внимание на то, что на титульном листе не должны отображаться верхний колонтитул и номер страницы.

Оглавление

1. Работа с текстом	95
2. Работа с таблицами	96
3. Работа со списками	97
4. Работа с рисунком	98

1. Работа с текстом

Существует большое количество различных графических пакетов, предназначенных для решения графических задач, — от маленьких и очень простых (например, ориентированных на решение одной небольшой задачи) до огромных и весьма сложных.

Как правило, подобные пакеты ориентированы не на программистов, а на пользователей — художников, модельеров, аниматоров. Большинство операций выполняется с помощью мыши, графического планшета и клавиатуры, а не путем написания кода (хотя подобная возможность часто присутствует и оказывается очень важной для решения некоторых задач). Подобные пакеты, как правило, можно разделить на пакеты для двухмерной и для трехмерной графики, который приведен на рис.1. –рис.3.

Пакеты двухмерной графики обычно ориентированы на работу с двумерными изображениями или их последовательностями. Простейшим примером такого пакета может служить обычный редактор изображений, присутствующий практически в любой операционной системе. При этом сам пакет может быть очень сложным.

Многие подобные пакеты рассматривают изображение как набор полупрозрачных слоев. У каждого слоя есть степень прозрачности (своя для каждого пикселя и общая для всего слоя), к слою могут быть применены различные визуальные эффекты. Итоговое изображение строится как результат комбинирования подобных слоев между собой.

Среди пакетов двухмерной графики можно произвести деление на пакеты растровой графики и пакеты векторной графики (хотя иногда граница между ними оказывается довольно размытой: так, в пакете растровой графики может быть поддержка отдельных векторных возможностей).¹

¹ https://studme.org/156222/informatika/osnovnye_graficheskie_pakety

2. Работа с таблицами

2	3	56	88	2
3	1	67	1	3
Итого		226		

3. Работа со списками

1. Передник;
2. Поварешка;
3. Ложка.

- * Батарейка А;
- * Батарейка АА;
- * Батарейка ААА;

1. Принтер
 - 1.1. Струйный
 - 1.1.1. Цветной
 - 1.1.2. Черно-белый
 - 1.2. Матричный
2. Камера
 - 2.1. Цифровая
 - 2.1.1. Объем
 - 2.1.2. Качество
 - 2.2. Пленочная

4. Работа с рисунком



Рисунок 1 ППП Компас -3D



Рисунок 2 - ППП Компас -3D (измененный масштаб)



Рисунок 4 - ППП Компас -3D (рисунок обрезан снизу)

Приложение Б

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Государственное бюджетное образовательное учреждение
высшего образования

«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ
ИМПЕРАТОРА АЛЕКСАНДРА I»

Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»

Дисциплина: «Информатика»

О Т Ч Е Т
о лабораторной работе № 2

на тему:
СТРУКТУРА СЛЕДОВАНИЕ»

В а р и а н т X

Выполнил студент
Факультета *БФО*
Группы *XXX-000-з*
Номер зачетной книжки
.....

Иванов И. И.

Санкт-Петербург
20__

1. Постановка задачи [б,к]

Найти длину окружности L , заданную радиусом.

Входные данные:

R – радиус окружности, переменная вещественного типа,
 $\pi = 3,14$ – константа вещественного типа.

Выходные данные:

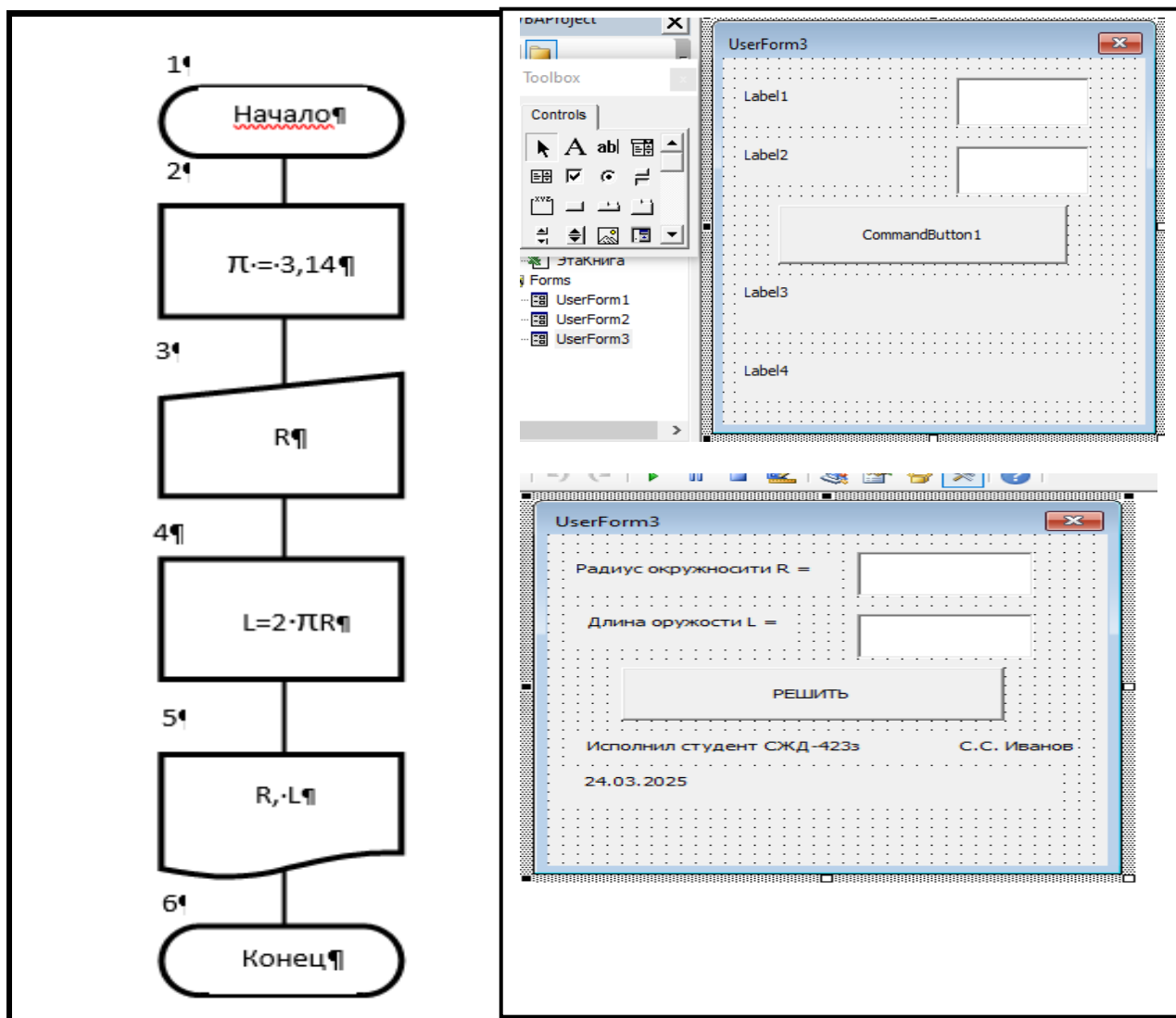
L – длина окружности, переменная вещественного типа.

2. Математическая модель задачи

$$L = 2\pi R$$

3. Разработка алгоритма

4. Разработка визуальной части проекта



5. Код приложения в VBA

```

CommandButton1

Private Sub CommandButton1_Click()
    Dim R As Single, L As Single
    Const Pi As Single = 3.14
    R = Val(TextBox1.Text)
    L = 2 * Pi * R
    TextBox2.Text = Str(L)
End Sub

```

6. Отладка приложения:

Радиус окружности R =

Длина окружности L =

Исполнил студент СЖД-423з С.С. Иванов

24.03.2025

Ручной счет:

При $R = 10$ $L = 62,831$

Решение примера в LibreOffice

```

1 REM ***** BASIC *****
2
3 Sub Main
4
5 Dim R As Single, L As Single
6 Const Pi As Single = 3.14
7 R = Inputbox("Введите R =")
8 L = 2 * Pi * R
9 Print "Длина круга =", L
10 End Sub

```

Введите R =

```

1 REM ***** BASIC *****
2
3 Sub Main
4
5 Dim R As Single, L As Single
6 Const Pi As Single = 3.14
7 R = Inputbox("Введите R =")
8 L = 2 * Pi * R
9 Print "Длина круга =", L
10 End Sub
11
12

```

Предупреждение

Длина круга = 62,799999237

Решение примера в Python

```
1  PI = 3.1415
2  r = float(input('Введите радиус: '))
3  L = 2 * PI * float(r)
4  print ('L =',L)
```

Введите радиус: 10

L = 62.830000000000005

Приложение В

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Государственное бюджетное образовательное учреждение
высшего образования

«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА АЛЕКСАНДРА I»

Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»

Дисциплина: «Информатика»

О Т Ч Е Т
по лабораторной работе № 3
СТРУКТУРА РАЗВИЛКА

Вариант X

Выполнил студент
Факультета *БФО*
Группы *XXX-000*
Номер зачетной книжки
.....

Иванов И. И.

Санкт-Петербург
20__

1. Постановка задачи

Дано действительное число X , являющееся аргументом функции $Y = \sin X$, если оно отрицательно, и функции $Y = X^2$, если положительно.

Входные данные:

X – аргумент функции, переменная вещественного типа

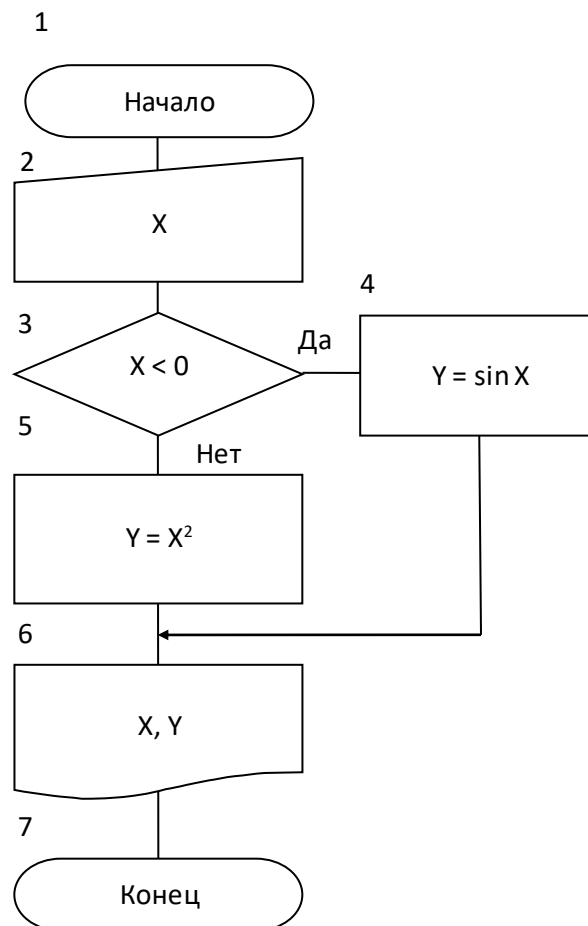
Выходные данные:

Y – значение функции, переменная, вещественного типа

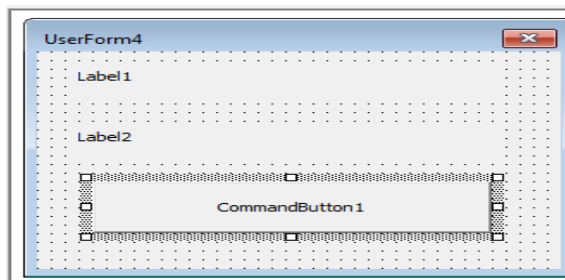
2. Математическая модель задачи

$$Y = \begin{cases} \sin X, & \text{если } X < 0 \\ X^2, & \text{если } X \geq 0 \end{cases}$$

3. Разработка алгоритма



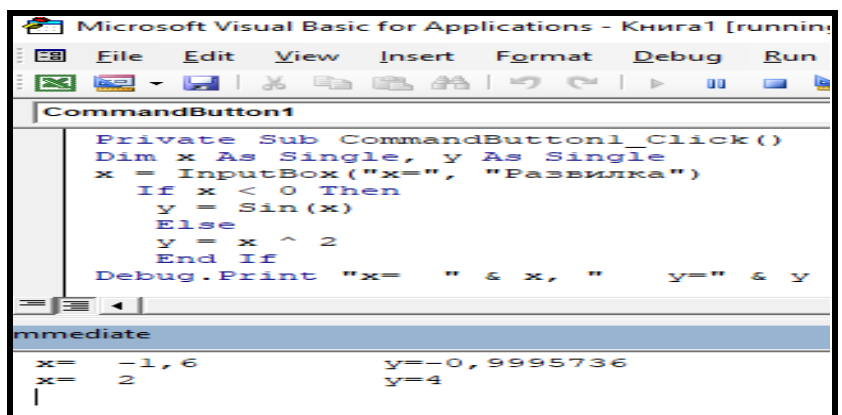
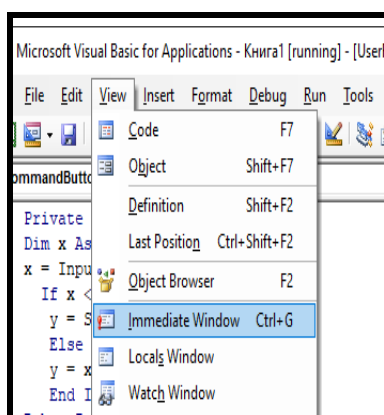
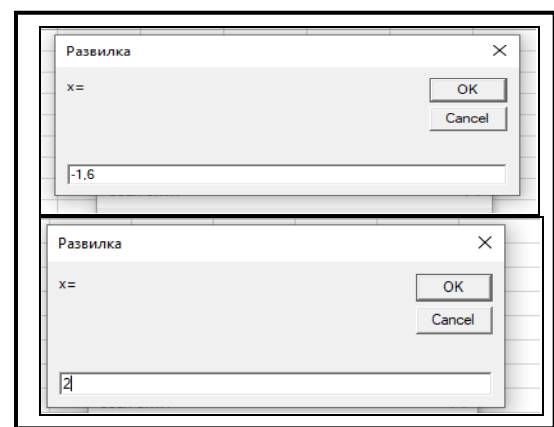
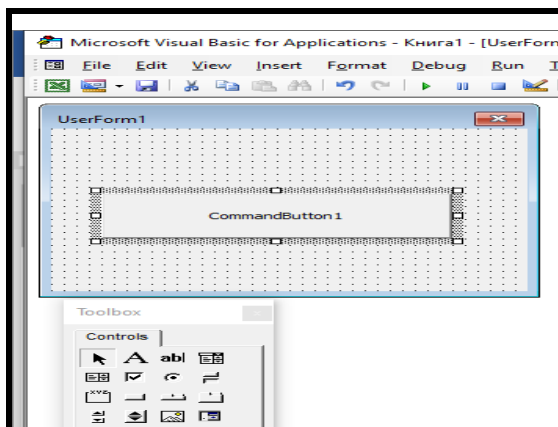
4. Разработка визуальной части проекта



5. Код приложения

```
Private Sub CommandButton1_Click()
    Dim x As Single, y As Single
    x = InputBox("x=", "Развилка")
    If x < 0 Then
        y = Sin(x)
    Else
        y = x ^ 2
    End If
    Debug.Print "x= " & x, " y=" & y
End Sub
```

6. Отладка программы



Результаты ручного счета:

При $X = -1,6$ $Y = \sin X = -0,99$

При $X = 2$ $Y = X^2 = 4$

Решение примера в LibreOffice

```
1 REM ***** BASIC *****
2
3 Sub Main
4 Dim x As Single, y As Single
5 x = InputBox("x=", "Развилка")
6 If x < 0 Then
7   y = Sin(x)
8 Else
9   y = x ^ 2
10 End If
11 Print "x= " & x, " y=" & y
12 End Sub
13
14
```

Развилка

x=

-1,6

OK Отменить

```
1 REM ***** BASIC *****
2
3 Sub Main
4 Dim x As Single, y As Single
5 x = InputBox("x=", "Развилка")
6 If x < 0 Then
7   y = Sin(x)
8 Else
9   y = x ^ 2
10 End If
11 Print "x= " & x, " y=" & y
12 End Sub
13
14
15
```

Предупреждение

x= -1,600000024 y=-0,999573588

OK Отменить

```
1 REM ***** BASIC *****
2
3 Sub Main
4 Dim x As Single, y As Single
5 x = InputBox("x=", "Развилка")
6 If x < 0 Then
7   y = Sin(x)
8 Else
9   y = x ^ 2
10 End If
11 Print "x= " & x, " y=" & y
12 End Sub
13
14
```

Предупреждение

x= 2 y=4

OK Отменить

Решение примера в Python

```
1  import math
2  x = float(input('Введите значение X: '))
3  if x<0:
4      Y = math.sin(x)
5  else:
6      Y = x**2
7  print ('X =', x)
8  print ('Y =', Y)
```

Введите значение X: -1.6

X = -1.6

Y = -0.9995736030415051

Введите значение X: 2

X = 2.0

Y = 4.0

Приложение Г

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Государственное бюджетное образовательное учреждение
высшего образования

«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА АЛЕКСАНДРА I»

Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»

Дисциплина: «Информатика»

О Т Ч Е Т
по лабораторной работе № 4
СТРУКТУРА ЦИКЛ
Вариант X

Выполнил студент
Факультета *БФО*
Группы *XXX-000*
Номер зачетной книжки
.....

Иванов И. И.

Санкт-Петербург
20__

1. Постановка задачи

Вычислить значения функции $Y = \cos X$ для всех X , принимающих значения от 0 до 1,0 с шагом 0,1.

Входные данные: X_0 – начальное значение параметра цикла, вещественная переменная;
 X_k – конечное значение параметра цикла, вещественная переменная;
 dX – шаг изменения параметра цикла, вещественная переменная.

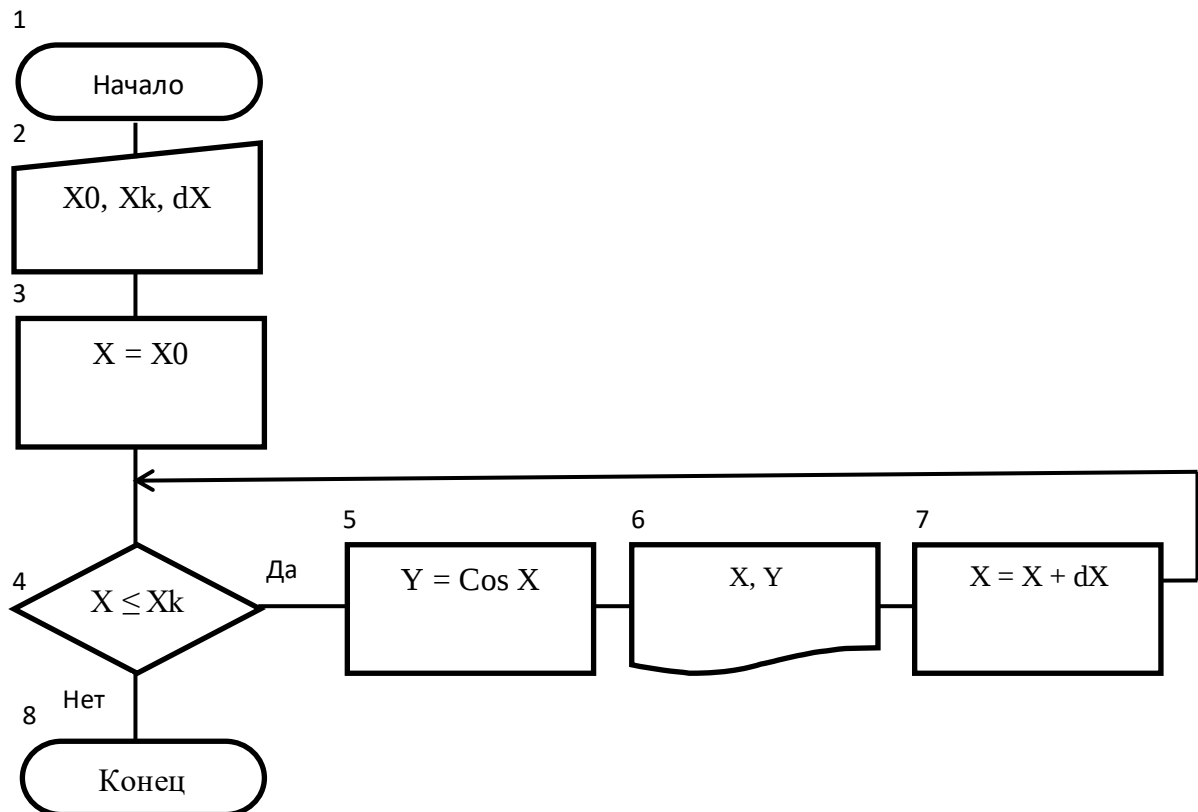
Выходные данные: X – параметр цикла, вещественная переменная;
 Y – значение функции, вещественная переменная.

2. Математическая модель

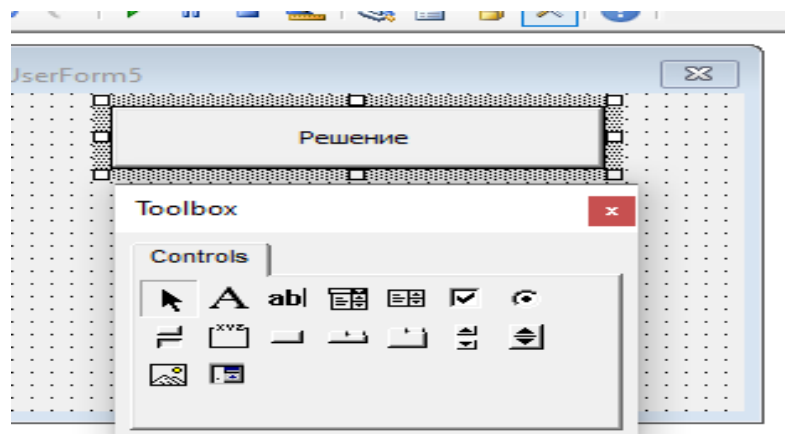
$$Y = \cos X$$

при $X_0 \leq X \leq X_k$ с шагом dX

3. Разработка алгоритма



4. Разработка визуальной части проекта

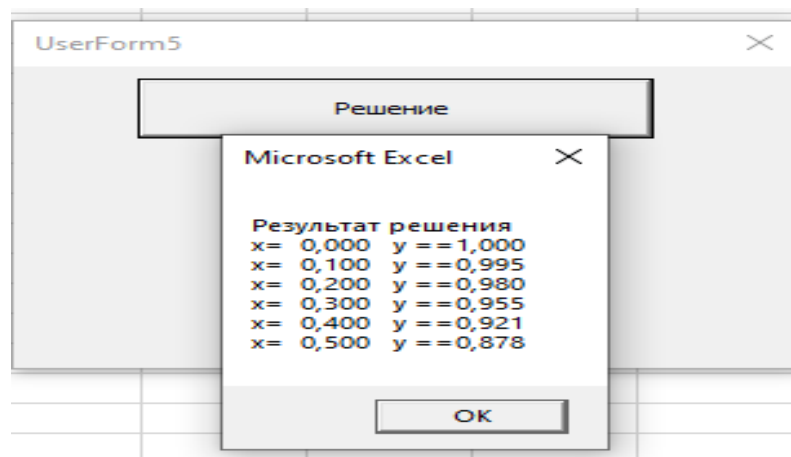


4. Код приложения

```
CommandButton1

Private Sub CommandButton1_Click()
    Dim X As Single, Y As Single
    Dim X0 As Single, Xk As Single, dX As Single
    X0 = InputBox("Введите X0", "Цикл")
    Xk = InputBox("Введите Xk", "Цикл")
    dX = InputBox("Введите dX", "Цикл")
    For X = X0 To Xk + dX / 2 Step dX
        Y = Cos(X)
        Text = Text & "x= " & (Format(X, "0.000") & " " & " y ==" & Format(Y, "0.000")) & Chr(10)
    Next
    MsgBox "Результат решения" & Chr(13) & Text
End Sub
```

6. Отладка программы



Ручной счет:

При $X = 0$, $Y = 1,0$

$X = 0,5$ $Y = 0,877$

$X = 0,8$ $Y = 0,696$

Решение примера в LibreOffice

The image shows two screenshots of the LibreOffice Basic editor. The left screenshot shows the VBA script, and the right screenshot shows the input dialog boxes for the script.

Left Screenshot (VBA Script):

```
1 REM ***** BASIC *****
2
3 Sub Main
4 Dim X As Single, Y As Single
5 Dim X0 As Single, Xk As Single, dX As Single
6 X0 = InputBox("Введите X0", "Цикл")
7 Xk = InputBox("Введите Xk", "Цикл")
8 dX = InputBox("Введите dX", "Цикл")
9 For X = X0 To Xk + dX / 2 Step dX
10 Y = Cos(X)
11 Text = Text & "x= " & (Format(X, "0.000")) & " " & " y ==" & Format(Y, "0.000")) & Chr(10)
12 Next
13 MsgBox "Результат решения" & Chr(13) & Text
14 End Sub
15
```

Right Screenshot (Input Dialogs):

The right screenshot shows two input dialog boxes. The first dialog box is titled "Цикл" and prompts "Введите X0". The second dialog box is titled "Цикл" and prompts "Введите Xk". Both dialog boxes have an "OK" button.

Input Dialog 1:

Цикл

Введите X0

0

OK

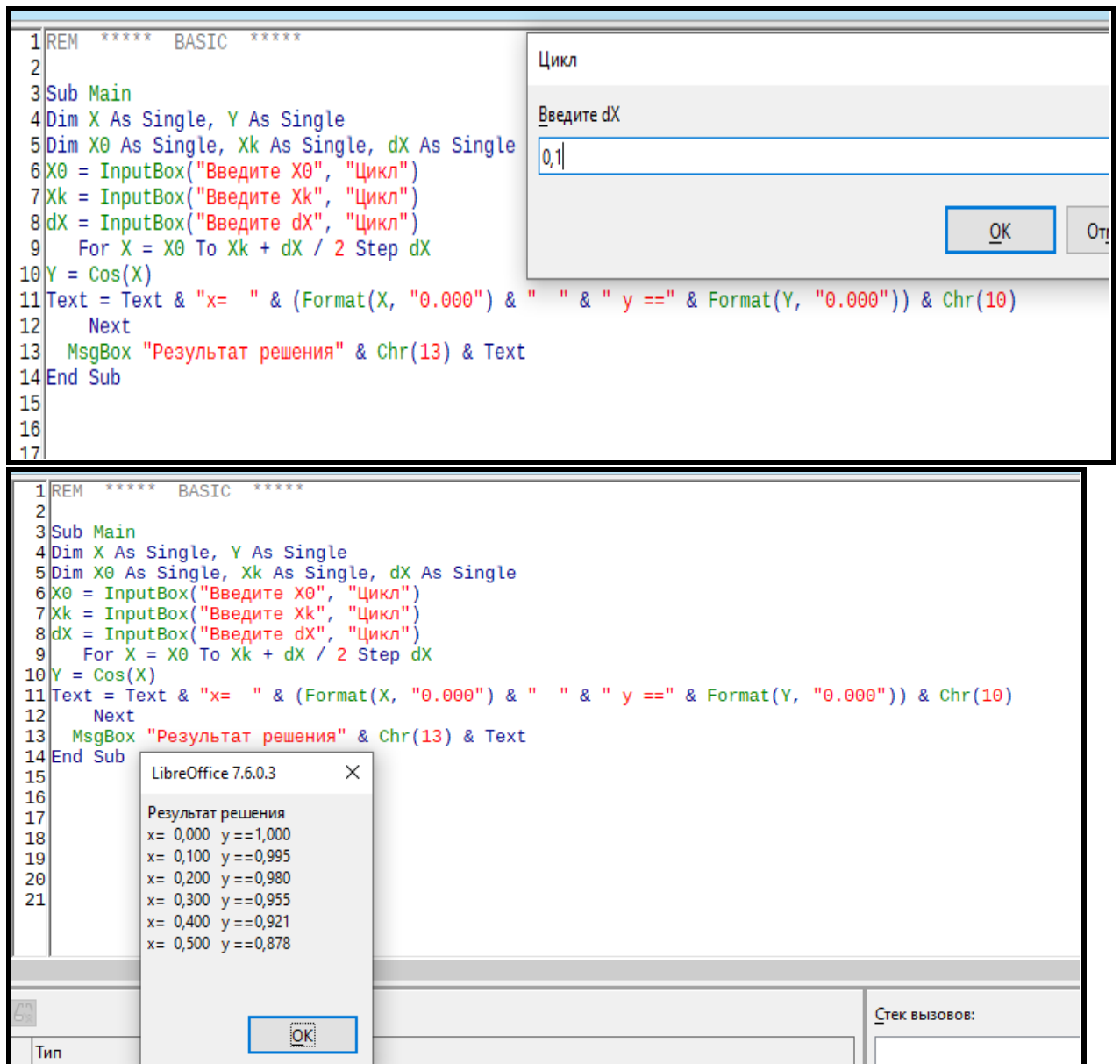
Input Dialog 2:

Цикл

Введите Xk

0,5

OK



Решение примера в Python:

```

1 import math
2 x = float(input('Введите начальное значение x (x): '))
3 xk = float(input('Введите конечное значение x (xk): '))
4 dx = float(input('Введите шаг изменения x (dx): '))
5 while x <= xk:
6     y = math.sin(x)
7     print('x =', x, ' y =', y)
8     x = round(x + dx, 7)

```

Введите начальное значение x (x): 0

Введите конечное значение x (x_k): 0.5

Введите шаг изменения x (dx): 0.1

$x = 0.0$	$y = 0.0$
$x = 0.1$	$y = 0.09983341664682815$
$x = 0.2$	$y = 0.19866933079506122$
$x = 0.3$	$y = 0.29552020666133955$
$x = 0.4$	$y = 0.3894183423086505$
$x = 0.5$	$y = 0.479425538604203$