

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Казанский государственный энергетический университет»

**КУРСОВАЯ РАБОТА
ПО ДИСЦИПЛИНЕ «АЛГОРИТМИЗАЦИЯ И ПРО-
ГРАММИРОВАНИЕ»**

Методические указания

**Для студентов, обучающихся
по образовательной программе
направления подготовки
09.01.03 Информатика и вычислительная техника
09.03.03 Прикладная информатика**

Казань
2024

УДК 378.14
ББК 74.58
К93

К93 Курсовая работа по дисциплине «Алгоритмизация и программирование»: методические указания / сост. Петрова Н.К., С.М. Куценко. – Казань: Казан. гос. энерг. ун-т, 2024. – 51 с.

Приведены основные рекомендации по выбору темы, объему, структуре, оформлению курсовой работы по дисциплине «Алгоритмизация и программирование».

Предназначены для студентов всех форм обучения по образовательной программе направления подготовки 09.01.03 Информатика и вычислительная техника, направленность (профиль) «Технология разработки программного обеспечения», 09.03.03 Прикладная информатика, направленность (профиль) «Прикладная информатика»..

УДК 378.14
ББК 74.58

ВВЕДЕНИЕ

Данные методические указания содержат основные рекомендации по выбору темы, объему, структуре, оформлению курсовой работы по дисциплине «Алгоритмизация и программирование». Курсовая работа предусмотрена рабочим учебным планом для обучающихся направления 09.03.03 Прикладная информатика

Выполнение курсовой работы направлено на формирование у студента следующих компетенций: знания современных информационных технологий и программных средств, в том числе отечественного производства при решении задач профессиональной деятельности; умения выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности; владения навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности; знания основных языков программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий, умения применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ, владения навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач, способности разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач; способности применять на практике новые научные принципы и методы исследований.

В результате студент должен:

знать:

- синтаксис и семантику языка программирования ,C++
- основные алгоритмы обработки массивов данных;
- принципы структурного программирования;
- принципы объектно-ориентированного программирования назначение и виды информационных систем;

уметь:

- выбирать алгоритм решения задачи и записывать его на языках программирования C++;

- иметь навыки оформления программы в стиле структурного программирования в виде набора пользовательских функций;
- уметь применять основные алгоритмы обработки данных;
- применять графический интерфейс приложений;
- внедрять апплеты и применять сервлеты в web-приложениях.

владеть:

- навыками записи алгоритмов на языках программирования C++;
- навыками использования массивов, файлов, более сложных структур данных в алгоритмах

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Целью курсовой работы (КР) является углубление знаний и расширение навыков по разработке алгоритмов и их реализации на персональном компьютере, грамотного оформления полученных результатов, умения представить результаты своей работы в виде научного доклада и защитить их в последующей дискуссии.

Выполняя курсовую работу, студент приобретает опыт и навыки использования теоретических знаний в решении практических задач, также расширяет практические навыки и опыт работы со специальной литературой.

Основными задачами при выполнении курсовой работы является повышение эффективности рассматриваемых процессов и приближение студента к практической деятельности.

Итогом выполнения курсовой работы является создание готового программного продукта путем обобщения теоретического материала по предложенной теме.

Критерием оценки курсовой работы является:

- 1) правильный алгоритм и код программы;
- 2) соответствие формы представления курсовой работы всем требованиям, предъявляемым к оформлению.

2. ТЕМЫ КУРСОВОЙ РАБОТЫ

Тематика курсовой работы утверждается на заседании кафедры. Ниже приведен перечень тем курсовой работы:

1. Дана действительная квадратная матрица порядка $2n$ ($n > 2$). Получить новую матрицу, переставляя ее блоки размером $n \times n$: а) крест-накрест; б) по часовой стрелке (левый верхний блок становится правым верхним, правый верхний – правым нижним и т.д.).
2. Составить программу, которая заполняет квадратную матрицу порядка n числами, заданными функцией $F(k)$, записывая их в нее «по

спирали» а) по часовой стрелке, б) против часовой стрелки. Разработать проект для случая, когда $F(k)$ может быть разной (функция, как параметр).

Например, для $n=5$ и $F(k) = k$ получаем следующие матрицы:

а) 1	2	3	4	5	б) 1	16	15	14	13
16	17	18	19	6	2	17	24	23	12
15	24	25	20	7	3	18	25	22	11
14	23	22	21	8	4	9	20	21	10
13	12	11	10	9	5	6	7	8	9

3. Дан одномерный целочисленный массив размера N . Разработать не менее двух способов шифрования/дешифрования элементов массива с применением побитовых операций.

4. Дана вещественная матрица размером $(N \times M)$. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (или один из них) оказался в левом верхнем углу.

5. Разработать проект, который обеспечивает выполнение операций сложения, вычитания и умножения над матрицами целых чисел. Выбор выполняемой операции осуществляет пользователь. Для арифметических операций над матрицами использовать перегрузку операторов.

6. Из массива удалить самую длинную цепочку четных элементов. Пример, из массива $A[8]$: 4 1 4 2 1 2 4 6 должен получиться массив $A[5]$: 4 1 4 2 1 (самая длинная цепочка четных чисел включает элементы с 6 по 8: 2 4 6). Проверить полноту программы на длинных массивах с большим набором последовательностей целых чисел. Рассмотреть ситуацию, когда подобных цепочек нет или нескольких цепочек одинаковой максимальной длины. .

7. Из массива A удалить те элементы, которые встречаются и в массиве A и в массиве B по крайней мере по n раза. Пример для $n = 2$, массив $A[8]$: 3 3 4 5 2 3 5 9, массив $B[7]$: 1 2 3 4 5 2 5. По 2 раза в обоих массивах встречается только элемент, равный 5. Массив A после удаления примет вид: $A[6]$: 3 3 4 2 3 9.

8. Из массива A удалить те цепочки четных элементов, в которых есть хотя бы один элемент из массива B . Пример, массив $A[9]$: 3 2 4 5 2 3 2 6 5, массив $B[6]$: 1 3 4 7 8 9. Массив A после удаления примет вид: $A[7]$: 3 5 2 3 2 6 5.

9. Массивы A и B представляют собой два числовых множества из n и m элементов, соответственно. Разработать программы для получения объединения, пересечения и разности указанных множеств, создав структурную переменную множество и используя перегрузку для операторов объединения (+), разности (-) и т.д.

10. Массив A представляет собой матрицу коэффициентов линейного

уравнения n -го порядка. Разработать программу для нахождения решения системы методом Гаусса.

11. Массив A представляет собой матрицу коэффициентов линейного уравнения n -го порядка. Разработать программу для нахождения решения системы методом Крамера.

12. Массив A представляет собой матрицу коэффициентов линейного уравнения n -го порядка. Разработать программу для нахождения решения системы методом Зейделя.

13. Массив A представляет собой матрицу коэффициентов линейного уравнения n -го порядка. Разработать программу для нахождения решения системы методом обратной матрицы.

14. Построить функциональную матрицу $F(x,y) = \sqrt{x} + \sqrt[3]{y} - \frac{1}{\sqrt{x^2 + y^2}}$ в n

точках для любого заданного диапазона x, y . Найти координаты (x_{\min}, y_{\min}) (x_{\max}, y_{\max}) экстремальных точек функции, если таких точек несколько указать координаты всех точек. Построить наиболее «короткий» путь от любой минимальной к любой максимальной точке. Под «коротким» путём понимается минимальное кол-во клеток, через которые от минимума можно пройти к максимуму.

15. Дан файл, содержащий текст на русском языке. Подсчитать количество слов, начинающихся и заканчивающихся на одну и ту же букву и выдать эти буквы с указанием соответствующего количества слов.

16. Написать программу, которая позволяет зашифровывать и расшифровывать сообщение с помощью «Кодового слова»: используется кодовое слово (любое слово без повторяющихся букв). Данное слово вставляется впереди алфавита и остальные буквы по порядку дописываются, исключая те, которые уже есть в кодовом слове. Пример: кодовое слово - NOTEPAD. Исходный: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Замена: N O T E P A D B C F G H I J K L M Q R S U V W X Y Z.

17. Текст программы на C++ хранится в файле на диске. Составить программу обработки текста программы: 1) определить максимальную степень вложенности циклов в программе; 2) определить общее количество строк и количество символов, отличных от пробела; 3) удалить из текста программы все комментарии.

18. Написать подпрограмму–функцию $Form(S, X, Y)$, где S – строка, X и Y – вещественные переменные. В строке записано арифметическое выражение, содержащее переменные X и Y , константы (целые или вещественные), операции $+$, $-$, $*$, $/$. Порядок операций определен скобками. Подпрограмма–функция возвращает значение арифметического выражения при заданных значениях X и Y .

19. Написать подпрограмму–функцию $Form(S, X)$, где S – строка, X – вещественная переменная. В строке записано арифметическое выражение, содержащее переменную X , константы (целые или вещественные), операции $+$, $-$, $*$, $/$. Порядок операций определен скобками. Подпрограмма–функция возвращает значение арифметического выражения при заданном значении X .

20. Разработать проект, в котором реализуются стандартных операции над трёхмерными векторами $\vec{a} = x\vec{i} + y\vec{j} + z\vec{k}$. Создать соответствующие структуры, функции с ними и реализовать перегрузку операторов.

21. Разработать проект, в котором реализуются стандартных операции над комплексными числами в алгебраической форме $z = A + iB$. i – комплексная $1 = \sqrt{-1}$. Создать соответствующие структуры, функции с ними и реализовать перегрузку операторов.

22. Разработать проект, в котором реализуются стандартные операции над комплексными числами в экспоненциальной форме $z = e^{x+iy}$. i – комплексная $1 = \sqrt{-1}$. Создать соответствующие структуры, функции с ними и реализовать перегрузку операторов.

23. Разработка проекта тестирования по курсу «Программирование на C++». Проект должен включать в себя вопросы открытого и закрытого типа. Вопросы, их тип и варианты ответов должны быть записаны в файл. Диалог с тестируемым должен быть организован через консоль. Можно использовать Windows Forms, которые поддерживаются в версиях, начиная с 2022.

24. Разработка компьютерную игру «Угадай число». Суть игры: Компьютер моделирует случайное число на основе заданного диапазона. Игроку случайным образом выделяется от n до m жизней. Игрок задаёт число, если не угадал, ПК сообщает ему: меньше или больше загаданного введённое число. ПК предлагает сделать ещё одну попытку с потерей жизни. Этот процесс продолжается до тех пор, пока А) игрок не угадает число; Б. не закончились жизни; С) Игрок не хочет больше играть. Все ситуации необходимо снабдить соответствующими комментариями.

25. Разработка компьютерную игру «Детское лото». Суть игры: Компьютер случайным образом моделирует две карты «лото» по 5 чисел в диапазоне от 1 до 20 на каждой карте (для 2-х игроков). При этом надо предусмотреть, чтобы числа на одной карте не повторялись. Затем компьютер выбрасывает «бочонок» - случайное неповторяющееся число из диапазона от 1 до 20. Выдаётся сообщение «Выпало число N». Высвечивается карта 1-го игрока и запрашивается есть ли у него такое число? Игрок выбирает – да/нет. Если он сказал «Да», а такого числа на карте нет, то ему – штрафное очко, а если число на карте есть, то оно удаляется с карты. Побеждает тот игрок, у которого карта обнулится первая. Однако, если есть K штрафных очков у первого игрока и L штрафных очков у второго

игрока, второй игрок имеет право ещё (K-L) раз выкинуть бочонок. Если его карта обнулится раньше, чем исчерпаются штрафные очки, он – победитель. Если штрафных очков осталось 0, то – ничья. В конце игры сообщается имя победителя.

26. Задан закон распределения дискретной случайной величины рядом распределения:

x_i	x_1	x_2	\dots	x_n
y_i	y_1	y_2	\dots	y_n

При этом плотность распределения y_i удовлетворяет условию $\sum_{i=1}^n y_i = 1$.

Составить программный проект, который для любого ряда распределения

вычисляет математическое ожидание $m=M(x)=\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n y_i}$, дисперсию

случайной величины $D(x)=\sum_{i=1}^n y_i (x_i - m)^2$, среднее квадратичное отклонение

$\sigma_x = \sqrt{D(x)}$. Перед вычислением, проверить, удовлетворяет ли ряд распределения свойствам плотности распределения.

27. Непрерывная случайная величина описывается функцией плотности вероятности $f(x)$, такой, что $\int_{-\infty}^{+\infty} f(x)dx = 1$. Составить программный

проект, который для любой $f(x)$ вычисляет математическое ожидание $m=M(x)=\int_{-\infty}^{+\infty} xf(x)dx$, дисперсию случайной величины

$D(x)=\int_{-\infty}^{+\infty} (x_i - m)^2 f(x)dx$, моду случайной величины – то значение, при

котором плотность распределения максимальна. Рассчитать указанные параметры для двух вариантов плотности распределения:

$$f_1(x) = \begin{cases} 0, & \text{при } x < 0, \\ x - x^2/4, & \text{при } x \in [0; 2], \\ 0, & \text{при } x > 2, \end{cases} \quad f_2(x) = \begin{cases} 0, & \text{при } x < 0, \\ \frac{1}{2} \sin(x), & \text{при } x \in [0; \pi], \\ 0, & \text{при } x > \pi, \end{cases}$$

Перед вычислением, проверить, удовлетворяет ли функция $f(x)$ свойствам плотности распределения.

3. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

При выполнении курсовой работы студент должен:

- разработать структуры данных и алгоритмы;
- реализовать функции и основную программу;
- разработать тесты и выполнить тестирование программы;
- оформить пояснительную записку и приложения.

В ходе выполнения работы должны быть осуществлены следующие этапы:

3.1 Постановка задачи и разработка структуры входных и выходных данных;

Решение задачи начинается с ее постановки. Постановка задачи - точная формулировка решения задачи на компьютере с описанием входной и выходной информации. Входная информация по задаче – это данные, поступающие на вход задачи и используемые для ее решения. Выходная информация может быть представлена в виде документов, кадров на экране монитора, информации в базе данных, выходного сигнала устройству управления.

В данном разделе дается точное описание исходных данных, условий задачи и целей ее решения. На этом этапе условия задачи, записанные в форме различных словесных описаний, необходимо выразить на формальном языке математики. Обычно математическая модель – это набор уравнений, неравенств и ограничений, приближенно описывающих задачу. При построении математической модели отбрасываются некоторые свойства реальной задачи, мало влияющие на решение.

Описываются ограничения, накладываемые на исходные данные, необходимая разрядность и точность представления исходных данных и результатов решения. Указываются возможные пределы изменения входных параметров задачи.

Используемые структуры данных должны быть представлены в виде схем данных.

3.2 Разработка алгоритма решения задачи

Дается обобщенное словесное описание алгоритма решения поставленной задачи, излагаются основные требования к алгоритму и пути их реализации. Приводится схема алгоритма, состоящая из укрупненных модулей. Дается пояснение назначения и состава каждого модуля.

Обобщенный алгоритм обычно использует обозначения и термины исходной задачи.

На следующем этапе каждый модуль детализируется. Выделяются укрупненные команды, реализуемые по вспомогательным алгоритмам. Тот же подход применяется при разработке вспомогательных алгоритмов.

Разработанный алгоритм должен быть оформлен в виде блок-схемы, отвечающей принципам структурного программирования. Структурное программирование - методология разработки программного обеспечения, предложенная в 70-х годах XX века [Дейкстрой](#) и разработанная и дополненная [Виртом](#). В соответствии с данной методологией любая программа представляет собой структуру, построенную из трёх типов базовых конструкций: последовательное исполнение — однократное выполнение операций в том порядке, в котором они записаны в тексте программы;

ветвление — однократное выполнение одной из двух или более операций, в зависимости от выполнения некоторого заданного условия;

цикл — многократное исполнение одной и той же операции до тех пор, пока выполняется некоторое заданное условие (условие продолжения цикла).

В программе базовые конструкции могут быть вложены друг в друга произвольным образом, но никаких других средств управления последовательностью выполнения операций не предусматривается. Повторяющиеся фрагменты программы (либо не повторяющиеся, но представляющие собой логически целостные вычислительные блоки) могут оформляться в виде так называемых подпрограмм (процедур или функций). В этом случае в тексте основной программы вместо помещённого в подпрограмму фрагмента вставляется инструкция вызова подпрограммы. При выполнении такой инструкции выполняется вызванная подпрограмма, после чего исполнение программы продолжается со следующей за командой вызова подпрограммы инструкции.

3.3 Графическая визуализация алгоритма с помощью блок-схем


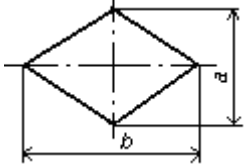
Перечень, наименование, обозначение и размеры обязательных символов и отображаемые ими функции в алгоритме и программе обработки данных, в идеале, должны соответствовать указанным в ГОСТ 19.003-80. Например, в табл. 1 приведён лишь фрагмент обозначений символов и соответствующих им функций.

Оформление блок-схем по ГОСТ — это длительная и кропотливая работа, требующая специальных навыков. Поскольку курсовая работа носит учебный характер, выполняется в весьма ограниченные сроки и её результаты не проходят производственную проверку, при составлении блок-схем разрешается упрощённый вариант их отображения — Табл. 2.

Размеры блоков можно менять в зависимости от содержания, но соблюдая пропорции (это нарушение ЕСПД, но в данном, учебном, варианте мы можем обойти это требование стандарта). Текст внутри блоков должен быть напечатан чётко, размер шрифта должен быть подобран так, чтобы его легко было прочесть.

Таблица 1

ПЕРЕЧЕНЬ, НАИМЕНОВАНИЕ, ОБОЗНАЧЕНИЕ СИМВОЛОВ И
ОТОБРАЖАЕМЫЕ ИМИ ФУНКЦИИ

Наименование	Обозначение и размеры в мм	Функция
1. Процесс		Выполнение операций или группы операций, в результате которых изменяется значение, форма представления или расположение данных
2. Решение		Выбор направления выполнения алгоритма или программы в зависимости от некоторых переменных условий

Блок-схема должна отражать стратегию решения: в неё **не включают-ся** операторы описания переменных, в том числе и динамических и прочих массивов, не детализируется инициализация переменных операторами присваивания (если это присваивание не несёт на себе алгоритмической нагрузки), не детализируется вид печати данных.

Если завершается функция с возвращаемым значением, то в блоке конца необходимо писать «Возврат имя_переменной», для функций типа void пишется просто «Возврат» и только для *main* – «Конец».

Блоки начала функций должны отражать прототип функции без описания типов. Если один формальным параметром является массив, его нужно писать со скобками:

Print(A[], n, m)

Если задачей функции является ввод или вывод данных, то такая функция помещается в соответствующий блок ввода/вывода:

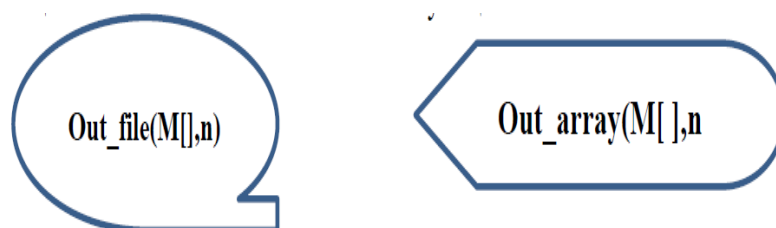
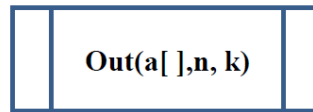


Таблица 2.

ОСНОВНЫЕ БЛОКИ ДЛЯ ФОРМИРОВАНИЯ БЛОК-СХЕМЫ

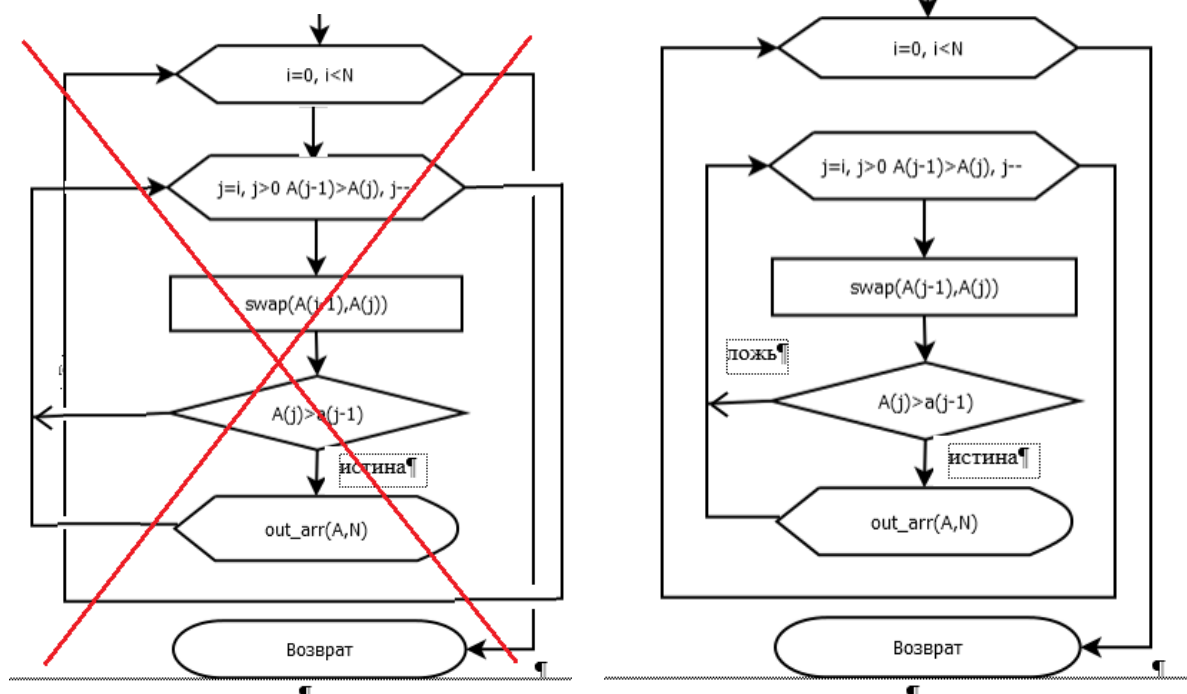
	Начало/Конец функции
	Явная инициализация переменных и констант
	Ввод с консоли
	Вывод на экран
	Вывод на бумагу
	Ввод/вывод в файл
	Блок расчётов (процессорный блок)
	Блок проверки условия
	Счётный оператор цикла в формате VBA: i=1 to 3 step 2 или C: s=0,i=0;i<n;i++,s+=ds
	Вызов функции

Если в функции предусмотрен тип ввода/вывода – на консоль или в файл, в зависимости от заданного флага k , то она помещается в стандартный блок вызова функции:



Для эффективного построения блок-схем рекомендуется использовать онлайн-приложение <https://www.drawio.com/> . Если по каким-либо причинам доступ к этому ресурсу отсутствует, то можно пользоваться элементами *Вставка/Фигуры* приложения MS Word.

При оформлении вложенных циклов и операторов необходимо следить, чтобы линии соединения блоков НЕ ПЕРЕКРЕЩИВАЛИСЬ.



3.4 Разработка структуры программы, написание текста программы

Разработка программы ведётся пошагово, методом "сверху вниз". Сначала пишется текст основной программы, в котором вместо каждого связного логического фрагмента текста вставляется вызов подпрограммы, которая будет выполнять этот фрагмент. Вместо настоящих, работающих подпрограмм, в программу вставляются "затычки", которые ничего не делают.

Все функции должны иметь строгую спецификацию. Разработанная программа должна быть протестирована, при этом следует уделить внимание подбору тестов и обоснованию их полноты.

При программной реализации алгоритма должен использоваться структурный (функционально-модульный) подход.

На этапе разработки рабочего проекта необходимая степень детализации алгоритмов обычно выбирается такой, чтобы предписания разработанных алгоритмов могли записываться на языке программирования, выбранном для составления текста программы. При детализации алгоритма необходимо перейти к обозначениям, принятым для разработки программ на алгоритмическом языке. При этом имена следует выбирать таким образом, чтобы они отражали сущность используемых параметров.

Кодирование должно быть простым. Изохренное программирование может обойтись слишком дорого при отладке и модификации программы. Необычное кодирование (например, использование скрытых возможностей машины) часто препятствует отладке программы и затрудняет ее использование другими программистами.

Программа должна быть по возможности универсальной. Универсальные программы обеспечивают независимость программы от конкретного набора данных. Например, универсальная программа использует в качестве параметров переменные, а не константы, обрабатывает вырожденные случаи и т. д.

Универсальность программы экономит время по дальнейшей работе над ней и обеспечивает широкие возможности по использованию. Разрабатывая такие программы, можно предвидеть будущие изменения в спецификациях этой программы.

Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ.

При написании программы следует применять операторы, позволяющие использовать основные алгоритмические структуры.

При написании программ не следует забывать о хорошем стиле программирования. После заголовка процедуры или функции записывается комментарий, содержащий поясняющий текст, а именно: назначение подпрограммы; перечень и назначение формальных параметров, их тип. Комментариями должны быть снабжены и основные смысловые блоки программы или подпрограммы.

Для облегчения чтения текста программы отдельные операторы программы записываются с отступом.

3.5 Разработка тестов

Перечисляются требования, подлежащие проверке при испытании программы, а также порядок и методика их контроля. Приводятся исходные данные для решения контрольного примера и ожидаемые результаты.

Прилагается распечатка решения контрольного примера и снимки экрана с результатами тестирования.

В приложения выносятся:

- листинги программ;
- руководство пользователю; (по желанию)
- текст контрольного примера и результаты тестирования (если их очень много или они слишком большие для размещения в тексте пояснительной записки).

3.6 Отладка программы

Полученная программа проверяется и отлаживается. После того, как программист убедится, что подпрограммы вызываются в правильной последовательности (то есть общая структура программы верна), подпрограммы-"затычки" последовательно заменяются на реально работающие, причём разработка каждой подпрограммы ведётся тем же методом, что и основной программы. Разработка заканчивается тогда, когда не останется ни одной "затычки", которая не была бы удалена. Такая последовательность гарантирует, что на каждом этапе разработки программист одновременно имеет дело с обозримым и понятным ему множеством фрагментов и может быть уверен, что общая структура всех более высоких уровней программы верна. При сопровождении и внесении изменений в программу выясняется, в какие именно процедуры нужно внести изменения, и они вносятся, не затрагивая непосредственно не связанные с ними части программы. Это позволяет гарантировать, что при внесении изменений и исправлении ошибок не выйдет из строя какая-то часть программы, находящаяся в данный момент вне зоны внимания программиста.

3.7 Написание отчета о работе

В соответствии с выполнением каждого этапа рекомендуется параллельно оформлять пояснительную записку курсовой работы.

Отчет о выполнении курсовой работы составляется студентом к моменту окончания семестра и сдается на проверку за 2 недели до зачетной недели.

Консультации проводятся на кафедре в установленные часы, либо в дистанционном формате с применением приложения Яндекс.Телемост. Во

время консультаций руководитель проектирования не дает студенту готовых решений, а путем постановки наводящих вопросов помогает студенту понять допущенные ошибки и найти правильный путь к решению вопроса. Руководитель дает указания лишь после того, как убедится, что студент достаточно ознакомился с данным вопросом и понял его сущность. Основной целью при этом является развитие у студента творческих навыков, умения обосновать и доказать наиболее важные положения проекта.

Положительная оценка выставляется при условии, что программа отлажена и вся сопутствующая документация оформлена верно. Представление только текста отлаженной программы без сопутствующей или неправильно оформленной документации не позволяет претендовать на положительную оценку. Небрежность и погрешности оформления снижают итоговый балл.

Оценка за курсовую работу выставляется на титульном листе пояснительной записки, в зачетной книжке студента и аттестационной ведомости по пятибалльной системе.

4. ТРЕБОВАНИЯ К СОДЕРЖАНИЮ И ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

4.1 Содержание и структура курсовой работы

Перед студентом ставится задача разработать программу на языке C++ с целью решения конкретной задачи. Результатом решения является:

- а) пояснительная записка, составленная с учетом требования стандартов Единой системы программной документации (ЕСПД);
- б) код программы в текстовом формате;
- в) скриншоты результатов работы программы, демонстрирующих соответствие контрольным примерам и обосновывающих полноту программы.

Пояснительная записка к работе оформляется в соответствии со стандартом и должна содержать:

- 1) титульный лист (пример оформления титульного листа приведен в Приложении 1);
- 2) содержание (структура содержания приведена в Приложении 2);
- 3) перечень принятых сокращений (при наличии);
- 4) основная часть (разработка алгоритмов и структур данных, программная реализация алгоритмов, тестирование разработанного программного обеспечения);
- 5) библиографический список (Приложение 3);
- 6) приложения.
- 7) отзыв руководителя отдельным файлом (Приложение 4).

Основная часть должна содержать разделы:

- 1) назначение и область применения (теоретическая, практическая, научная);
- 2) постановка задачи (все шаги, предполагаемые в результате работы компьютерной программы, функции, получаемые результаты);
- 3) используемые методы (стандартные и нестандартные модули и процедуры языка программирования);
- 4) описание алгоритма программы с описанием функций составных частей и связи между ними (блок-схема, алгоритмическая запись решения задачи);
- 5) описание входных и выходных данных (описание информации, как вводятся данные, тип данных, где и какого вида выводится информация);

Результаты не проходят производственную проверку, поэтому при составлении блок-схем разрешается упрощённый вариант их отображения – Табл. 2. Размеры блоков можно менять в зависимости от содержания, но соблюдая пропорции (это нарушение ЕСПД, но в данном, учебном, варианте мы можем обойти это требование стандарта). Текст внутри блоков должен быть прописан чётко, размер шрифта должен быть подобран так, чтобы его легко было прочесть.

4.2. Оформление пояснительной записки

1. Оформление текст пояснительной записки делается в приложении Word и предполагает наличие следующих составляющих: титульный лист, содержание, [список обозначений и сокращений], основной текст, заключение, список литературы, приложения;
2. *Заключение* должно включать список выполненных вами действий: в работе рассмотрены методы того-то и того-то, разработан алгоритм, использующий приемы структурного программирования, в частности, разбиение задачи на указанное число функций, алгоритм визуализирован словесно-блочной схемой, разработан контрольный пример.
Программа отлажена (или не отлажена и почему). Скриншот результатов работы представлен.
2. Объем пояснительной записки не ограничивается, форматы страниц следующие: шрифт Times New Roman, кегль 14 пт, междустрочный интервал – «множитель» 1,2, выравнивание текста – по ширине, нумерация страниц в *нижнем* колонтитуле. Форматирование документа делается с помощью диалогового окна «Абзац» так, как показано ниже:

Абзац

Отступы и интервалы Положение на странице

Общие

Выравнивание: По ширине ✓

Уровень: Основной текст ☐ Свернуты по умолчанию

Отступы

Слева: 0 см ✓

Справа: 0 см ✓

☐ Зеркальные отступы

Первая строка: ☒ Отступ на: 1,25 см

Интервал

Перед: 0 пт ✓

После: 0 пт ✓

☒ Не добавлять интервал между абзацами одного стиля

Междустрочный: ☒ Множитель значение: 1,2 ✓

Образец

Предидущий абзац Предидущий абзац Предидущий абзац Предидущий абзац Предидущий абзац Предидущий абзац
 абзац Предидущий абзац Предидущий абзац Предидущий абзац Предидущий абзац Предидущий абзац
 Консультации проводятся на кафедре в установленные часы, либо в дистанционном формате с
 применением приложения Яндекс.Телемост. Во время консультаций руководитель проектирования не дает сту-
 денту готовых решений, а путем постановки наводящих вопросов помог
 Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац
 Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац
 Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац
 Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац Следующий абзац

Табуляция... По умолчанию ОК Отмена

3. На титульном листе печатается: название курсовой работы, Фамилия И.О. исполнителя, курс, группа; Фамилия И.О. – для руководителя.
4. Заголовки глав и параграфов выделяются жирным шрифтом, центрируются, и делается 2 отступа от предыдущего текста и 1 перед последующим.
5. Содержание пояснительной записки должно полностью отражать все пункты пояснительной записки с указанием соответствующих страниц.
6. Список использованных источников - **не менее 3-х**, полное указание выходных данных для книжных и периодических изданий, адреса сайтов с которых заимствован материал, по тексту пояснительной записки должны быть ссылки на источники (по крайней мере, интернет-сайты);
7. В тексте пояснительной записки необходимо делать ссылки на источники.
8. Титульный лист НЕ нумеруется и НЕ входит в общее количество КР. «Содержание» нумеруется номером 1. Далее по порядку страницы пояснительной записки. Отзыв руководителя - отдельная страница, без номера, не входит в текст КР.

В список источников включают только наименования источников, которые использованы при написании текста и на которые в тексте есть ссылки. В работе рекомендуется использовать нумерованный список источников,

но допускается и алфавитное расположение фамилий авторов и заглавий книг и статей (если автор не указан).

Перед выполнением курсовой работы по собственной теме студент обязан написать *Заявление*, образец которого представлен в Приложении 5, и обосновать целесообразность её выполнения.

Образец всей пояснительной записки, начиная с титульного листа и заканчивая приложениями, приведён в Приложении 6.

5. КОНСУЛЬТАЦИИ И КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ

Консультации проводятся на кафедре в установленные часы, либо в письменном виде. Во время консультаций руководитель курсовой работы не дает студенту готовых решений, а путем постановки наводящих вопросов помогает студенту понять допущенные ошибки и найти правильный путь к решению вопроса. Руководитель дает указания лишь после того, как убедится, что студент достаточно ознакомился с данным вопросом и понял его сущность. Основной целью при этом является развитие у студента творческих навыков, умения обосновать и доказать наиболее важные положения проекта.

Положительная оценка выставляется при условии, что программа отлажена и вся сопутствующая документация оформлена верно. Представление только текста отлаженной программы без сопутствующей или неправильно оформленной документации не позволяет претендовать на положительную оценку. Небрежность и погрешности оформления снижают итоговый балл.

Оценка за курсовую работу выставляется на титульном листе пояснительной записки, в зачетной книжке студента и аттестационной ведомости по пятибалльной системе.

Максимальное количество баллов, которое можно получить за выполненную курсовую работу – 100.

В табл. 1 приведено примерное распределение баллов по каждому критерию. Общее количество баллов, получаемое за выполнение курсовой работы, вычисляется как сумма всех баллов по критериям оценивания с учетом особенности тематики курсовой работы – итоговый балл. Минимальное количество критериев оценивания курсовой работы – не менее трех.

После подсчета итоговый балл переводится в оценку на основании таблицы соответствия (табл. 2).

Таблица 1

Примерные варианты структуры оценки курсовой работы по критериям

№	Критерий оценки курсовых работ	Баллы
1	Текст программы	20
2	Блок-схема	15
3	Составленный контрольный пример	15
3	Правильно оформленная документация	20
4	Защита курсовой работы	30

Таблица 2

Шкала перевода баллов

Оценка	Итоговый балл
«3» удовлетворительно	55–75
«4» хорошо	76–84
«5» отлично	85–100

Оценка «отлично» выставляется за курсовую работу, которая грамотно оформлена имеет грамотно изложенный теоретический раздел, характеризуется логичным и последовательным изложением материала с соответствующими выводами и обоснованными предложениями по практическому применению результатов исследования. Такая работа должна иметь положительный отзыв научного руководителя. При ее защите студент показывает глубокие знания вопросов темы, свободно оперирует данными исследования, вносит обоснованные предложения по практическому применению результатов исследования, четко отвечает на поставленные вопросы.

Оценка «хорошо» выставляется за работу, которая в целом имеет положительный отзыв научного руководителя, но содержит ряд незначительных замечаний. При ее защите студент показывает знание вопросов темы, оперирует данными исследования, без особых затруднений отвечает на поставленные вопросы.

Оценка «удовлетворительно» выставляется за работу, которая носит в большей степени описательный характер. Работа имеет теоретический раздел, базируется на практическом материале, но характеризуется непоследовательностью в изложении материала. Представленные выводы автора необоснованны. В отзыве научного руководителя имеются серьезные замечания по содержанию работы и методике анализа. При ее защите студент проявляет неуверенность, показывает слабое знание вопросов темы, не дает полного аргументированного ответа на заданные вопросы.

Оценка «неудовлетворительно» выставляется за работу, которая не носит исследовательского характера и не отвечает требованиям, изложенным в данных методических указаниях по выполнению курсовых работ.

В курсовой работе нет выводов, либо они носят декларативный характер. В отзыве научного руководителя имеются серьезные замечания. При защите курсовой работы студент затрудняется отвечать на поставленные вопросы по теме, не знает теории вопроса, при ответе допускает существенные ошибки. Оценка «неудовлетворительно» может быть также выставлена студенту, представившему на защиту чужую курсовую работу, написанную и уже защищенную в другом вузе или на другой кафедре.

Приложение 1
Образец Титульного листа

Макет титульного листа на курсовую работу / проект



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГЭУ»)

(полное название института)

(полное название кафедры)

КУРСОВАЯ РАБОТА/КУРСОВОЙ ПРОЕКТ

по дисциплине «_____»,

тема «_____»

Выполнил:

_____ (Ф.И.О.)

обучающийся _____ курса

группы _____

(подпись)

Руководитель работы:

(Ф.И.О., должность, кафедра)

Работа выполнена и

защищена с оценкой _____ Дата защиты _____

(подпись руководителя)

Члены комиссии: _____

(должность) (подпись) (И.О. Фамилия)

(должность) (подпись) (И.О. Фамилия)

(должность) (подпись) (И.О. Фамилия)

Казань, 20__ г.

Образец оформления содержания**Содержание**

Введение.....	3
1. Цели и задачи курсовой работы.....	4
2. Темы курсовой работы.....	4
3. Выполнение курсовой работы.....	8
4. Требования к содержанию и оформлению курсовой работы.....	16
4.1. Содержание и структура курсовой работы.....	16
4.2. Оформление пояснительной записки	17
5. Критерии оценивания курсовой работы.....	19
Библиографический список.....	11
Приложение.....	22

Приложение 3 Образец оформления
библиографического списка

Образец оформления библиографического списка

Законодательные материалы

Российская Федерация. Конституция (1993). Конституция Российской Федерации [Текст] : офиц. текст. – М. : Проспект, 2000. – 48 с.

Российская Федерация. Законы. О милиции [Текст] : федер. закон : [принят 18 апреля 1991 г. № 1026–1 : по состоянию на 25 июля 2002 г.]. – 4-е изд. – М. : ИНФРА-М, 2002. – 34 с. – (Сер. «Федеральный закон»).

Российская Федерация. Законы. Семейный кодекс Российской Федерации [Текст] : [федер. закон : принят Гос. Думой 8 дек. 1995 г. : по состоянию на 1 мая 2004 г.]. – СПб. : ПИТЕР, 2004. – 93 с.

Книги с 1-м автором

Просветов, Г.И. Математические модели в экономике [Текст] : учеб.-метод. пособие / Г.И. Просветов. – 2-е изд., доп. – М. : РДЛ, 2006. – 160 с.

с 2-мя авторами

Настич, В.П. Управление качеством холоднокатаных полос [Текст] / В.П. Настич, А.И. Божков. – М. : ИНТЕРМЕТ ИНЖИНИРИНГ, 2006. – 214 с.

с 3-мя авторами

Нарижный, И.Ф. Роль саморегулируемых бизнес-сообществ в повышении конкурентоспособности малых предприятий [Текст] / И.Ф. Нарижный, В.В. Московцев, Д.М. Любавин. – Липецк : ЛЭГИ, 2006. – 136 с.

4 автора и более

Насосы. Вентиляторы. Кондиционеры [Текст] : справочник / Е.М. Росляков [и др.] ; под ред. Е.М. Рослякова. – СПб. : Политехника, 2006. – 822 с.

Наследственное право [Текст] : учеб. пособие для вузов / под ред. Н.А. Волковой. – М. : ЮНИТИ-ДАНА, 2006. – 192 с.

Народы и конфессии Приволжского Федерального округа [Текст] : справочник для гос. служащих. – М. : Мир, 2003. – 568 с. 49

Книги на иностранном языке

Cagianut, F. Unternehmenssteuerrecht [Текст] = Налоговое право в предпринимательстве / F. Cagianut, E. Hohn. – Bern : P. Haupt Verl., 1993. – 855 с.

Переводные издания

Крайг, Г. Психология развития [Текст] : пер. с англ. / Грейс Крайг, Дон Бо-кум. – 9-е изд. – СПб. : Питер, 2006. – 940 с.

Microsoft Access [Текст] : [пер.]. – М. : Мир книги, 2004. – 288 с. – (Компьютер – это просто!).

Сборники трудов

Право: история, теория, практика [Текст] : межвуз. сб. науч. тр., дек. 2006 г. – Липецк : ЛГТУ, 2006. – 191 с.

Прогрессивные технологии и оборудование в машиностроении и металлургии [Текст] : сб. науч. тр. междунар. науч.-техн. конф., посвящ. 50-летию ЛГТУ, 11-12 мая 2006 г. В 2-х ч. – Липецк : ЛГТУ, 2006.

Наука в Липецкой области: истоки и перспективы [Текст] : сб. докл. и тезисов обл. науч.-практ. конф., февр. 2004 г. В 3-х ч. Ч. 2. Гуманитарное направление / отв. за вып. Кисенко В.П., Скопинцев В.А. – Липецк : ЛГТУ, 2004. – 199 с.

Продолжающиеся сборники

Философия в XXI веке [Текст] : междунар. сб. науч. тр. / под общ. ред. проф. О.И. Кирикова. – Вып. 11. – Воронеж : ВГПУ, 2006. – 142 с.

Сборник научных трудов НГТУ [Текст]. – Вып. 1 (43). – Новосибирск : НГТУ, 2006. – 196 с.

Обзорные информации

Тарасова, Н.В. Теоретические и методологические основы модульно-компетентностной технологии обучения [Текст] / Тарасова Н.В., Смирнов С.А. – М., 2007. – 60 с. – (Аналит. обзоры по основным направлениям развития высш. образования / ФИРО ; Вып. 3).

Методические указания

Галушкин, А.М. Концепции современного естествознания [Текст] : метод. указ. с планами семинарских занятий для студентов I-IV курсов очной и очно-заочной форм обучения / сост. А.М. Галушкин. – Липецк : ЛГТУ, 2006. – 44 с.

Козлов, А.М. Разработка планировки цехов (участков) [Текст] : метод. указ. к практ. занятиям по курсу «Проектирование машиностроит. пр-ва» и выполнению диплом. проектирования / А.М. Козлов, И.И. Шацких. – Липецк : ЛГТУ, 2006. – 35 с.

Многотомные издания

Савельев, И.В. Курс общей физики [Текст] : учеб. пособие для вузов : в 5 т. / И.В. Савельев. – М. : АСТ : Астрель, 2005.

Персональный компьютер для всех [Текст] : в 4 кн. / под ред. А.Я. Савельева. – М. : Высш. шк., 1991.

Отдельный том

Савельев, И.В. Курс общей физики [Текст]. В 5 т. Т. 2. Электричество и магнетизм : учеб. пособие для вузов / И.В. Савельев. – М. : АСТ : Астрель, 2005. – 336 с.

Стандарты

ГОСТ 20799 – 88. Масла индустриальные. Технические условия [Текст]. – Введ. 1990 – 01 – 01. – М. : Стандартинформ, сор. 2006. – 5 с.

ГОСТ 8.021 – 2005. Государственная поверочная схема для средств измерений массы [Текст]. – Взамен ГОСТ 8.021 – 84 ; введ. 2007 – 01 – 01. – М. : Стандартинформ, 2006. – 7 с. – (Государственная система обеспечения единства измерений).

ГОСТ Р 7.0.4 – 2006. Издания. Выходные сведения. Общие требования и правила оформления [Текст]. – Введ. 2007 – 01 – 01. – М. : Стандартинформ, 2006. – 43 с. – (Система стандартов по информации, библиотечному и издательскому делу).

Сборники стандартов

Система стандартов безопасности труда : [сборник]. – М. : Изд-во стандартов, сор. 2003. – 191 с. – (Государственные стандарты). – Содерж.: 30 док.

Промышленная чистота [Текст] : (сб. стандартов, утв. до 1 июня 2004 г.). – М. : Изд-во стандартов, 2004. – 164 с. – (Национальные стандарты). – Содерж.: 13 док.

Патентные документы

Пат. 2187888 Российская Федерация, МПК Н 04 В 1/38, Н 04 J 13/00. Приемопередающее устройство [Текст] / Чугаева В.И. ; заявитель и патентообладатель Воронеж. науч.-исслед. ин-т связи. - № 2000131736/09 ; заявл. 18.12.00 ; опубл. 20.08.02, Бюл. № 23 (II ч.). – 3 с.

Заявка 1095735 Российская Федерация, МПК В 64 G 1/00. Одноразовая ракета-носитель [Текст] / Тернер Э.В. (США) ; заявитель Спейс Системз/Лорал. инк. ; пат. поверенный Егорова Г.Б. - № 2000108705/28 ; заявл. 07.04.00 ; опубл. 10.03.01, Бюл. № 7 (I ч.) ; приоритет 09.04.99, № 09/289,037 (США). – 5 с.

А.с. 1007970 СССР, МКИ В 25 J 15/00. Устройство для захвата неориентированных деталей типа валов [Текст] / В.С. Ваулин, В.Г. Кемайкин (СССР). - № 3360585/25–08 ; заявл. 23.11.81 ; опубл. 30.03.83, Бюл. № 12. – 2 с.

Промышленные каталоги

Оборудование классных комнат общеобразовательных школ [Текст] : каталог / М-во образования РФ, Моск. гос. пед. ун-т. – М. : МГПУ, 2002. – 235 с.

Машина специальная листогибочная ИО 217М [Текст] : листок-каталог : разработчик и изготовитель Кемер. з-д электромонтаж. изделий. – М., 2002. – 3 л.

Пример отзыва научного руководителя на курсовую работу (проект)

КГУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
 Федеральное государственное бюджетное образовательное
 учреждение высшего образования
 «КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
 (ФГБОУ ВО «КГУ»)

 (полное название института)

 (полное название кафедры)

Отзыв руководителя на курсовую работу/курсовой проект
 Обучающегося (ейся) _____
 (фамилия, имя, отчество)

Группа _____

 На тему: _____

Показатели	Критерии оценивания	Рейтинговая оценка (от 0 до 100 баллов)
1. Самостоятельность выполнения работы	Работа написана самостоятельно	
	Работа носит частично самостоятельный характер	
	Работа носит не самостоятельный характер	
2. Содержание работы	Полностью соответствует выбранной теме	
	Частично соответствует выбранной теме	
	Не соответствует теме	
3. Элементы исследования	Определены цели и задачи исследования, сформулированы объект и предмет исследования, показана история и теория вопроса	
	Определены цели и задачи исследования, не четко определены объект и предмет исследования, частично показана история и теория вопроса	
	Не определены цели и задачи исследования, не сформулированы объект и предмет исследования, не показана история и теория вопроса	
4. Цитирование и наличие ссылочного материала	Достаточно	

5. Наличие собственных выводов, рекомендаций и предложений, собственной позиции и ее аргументации	Да	
	Нет	
6. Оформление работы	Соответствует полностью требованиям	
	Соответствует частично требованиям	
	Не соответствует требованиям	
7. Библиография по теме работы	Актуальна и составлена в соответствии с требованиями	
	Актуальна и частично соответствует требованиям	
	Не соответствует требованиям	
Итоговый балл		

Отмеченные достоинства _____

Отмеченные недостатки _____

Заключение _____

Руководитель _____

(фамилия, имя, отчество, должность, учинил stamp, учинил печать)

Дата: « _____ » _____ 20 ____ г.

Подпись _____

Приложение 5 Образец Заявления
на тему курсовой работы

Образец заявления обучающегося на тему курсовой работы (проекта)

Заведующему кафедрой _____

Фамилия, имя, отчество зав. кафедрой

Фамилия, имя, отчество обучающегося в род. падеже

обучающего(ей)ся в группе _____
направления подготовки (специальности) _____

направленность (профиль) ОП _____

заявление.

Прошу утвердить мне тему курсовой работы (проекта) по дисциплине
_____ предложенную в инициативном порядке _____

в связи с (заполняется в целях обоснования целесообразности ее разработки)

Тема с руководителем согласована _____ / _____
подпись руководителя ФИО руководителя

_____ / _____
подпись обучающегося ФИО обучающегося

« ____ » _____ 20 ____ г.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГУ»)

Институт цифровых технологий и экономики

(полное название института)

Информационные технологии и интеллектуальные системы

(полное название кафедры)

КУРСОВАЯ РАБОТА

по дисциплине «Алгоритмизация и программирование»
та тему «Компьютерное моделирование функций Робота по заданному
алгоритму»

Выполнил:

Петров Иван Иванович
студент I курса группы ###-#-##

(подпись)

Руководитель работы:

Петрова Н.К., доцент ИТИС
(ФИО, должность)

Работа выполнена и

защита с оценкой

Дата защиты

##.##.##

(подпись
руководителя)

Члены комиссии:

(должность)

(подпись)

(И.О. Фамилия)

(должность)

(подпись)

(И.О. Фамилия)

Казань 202#

Содержание

Содержание.....	2
Определения, обозначения и сокращения	3
Введение.....	4
Разработка контрольного примера.....	5
Словесный алгоритм программы	6
Визуализация алгоритма задачи с помощью блок-схем.....	9
Анализ решения – сравнение с контрольным примером.....	12
Массовые расчёты.....	13
Заключение	14
Библиографический список	15
ПРИЛОЖЕНИЕ 1. Листинг программы	16

Определения, обозначения и сокращения

КР – Курсовая работа

W – белая краска (White)

B – черная краска (Black)

$A[M][N]$ – двумерный массив размера $N \times N$

$A[]$ – передаётся массив в функцию

VS – Visual Studio

Введение

Применение компьютеров в научных исследованиях и производственных процессах является необходимым условием изучения сложных систем, прогнозирования каких-либо событий, явлений, обеспечения работоспособности реальных технических систем. Принципы компьютерного моделирования – это современная эффективная процедура, обеспечивающая возможность целостного изучения поведения наиболее сложных систем как естественных, так и создаваемых для проверки теоретических гипотез.

Методами компьютерного моделирования пользуются специалисты практически всех отраслей и областей науки и техники, поскольку с их помощью можно прогнозировать и даже имитировать явления, события или проектируемые предметы в заранее заданных параметрах.

В рамках данной курсовой работы требуется разработать проект, который позволит моделировать функционал Робота по заданному алгоритму. *Суть задачи следующая.*

Робот движется по полю размером $M \times N$ кв.м., размеченного на квадраты размером $1\text{ м} \times 1\text{ м}$. У него два баллона с черной и белой краской. Его задача раскрасить поле в шахматном порядке. Робот проходит по полю в ленточном порядке: сначала слева направо проходит первый ряд, переходит на второй и обратным движением справа налево движется до левой границы поля и так до последней клетки. В емкостях находится W и B кг краски, соответственно, белого и чёрного цвета. Покраска начинается с белого цвета и идёт до тех пор, пока хотя бы в одном баллоне остаётся более 10г краски. В зависимости от номера клетки, количество краски на каждый квадратный метр определяется по формуле:

$$а) \text{ белая краска } K_w = A_w |\cos(2n)|, \text{ б) черная краска: } K_B = A_b |\sin(n)|, \quad (1)$$

здесь n – это номер клетки по порядку прохождения поля, т.е. первая клетка имеет номер 1, вторая 2 и т.д. Если рассчитываемое кол-во краски $K_B < 0,01$ кг, $K_w < 0,01$ кг, то потратить 0,010 кг.

Смоделировать процесс расхода краски на поле заданного размера (N , M) и заданными параметрами расхода краски A_b кг и A_w кг. Определить:

1. Достаточно ли выданного объёма на покрытие всего поля?
2. Если достаточно, то подсчитать сколько краски осталось в емкостях.
3. Если краски не хватило, то определить, на какой клетке она закончилась (для белой и черной отдельно).

4. Найти клетку, в которой ушло больше всего и меньше всего белой и черной краски, соответственно.

Разработка контрольного примера

Для понимания сути разрабатываемого приложения и формирования алгоритма разработаем контрольные тестовые примеры, позволяющие решить поставленную задачу с учётом удовлетворения требования полноты предстоящего решения.

Допустим, что исходные параметры следующие:

$M=3$ м, $N=2$ м.

$W=1$ кг, $B=1$ кг

$A_b = 0,5$ кг; $A_w = 0,6$ кг

Раскраску начнем с белой краски. Воспользуемся средой Excel для визуализации задачи. С помощью стандартных формул Excel получаем необходимые результаты.

Пример 1. Расход краски большой и заданного количества коаски не хватило на покраску всего поля. Найдены минимум и максимум расхода, а также клетки, оставшиеся не покрашенными.

	A	B	C	D	E	F	G	H	I	J
1	Исходные параметры задачи									
2	M=	3	N=	2		Оставшаяся краска				
3	W=	1	B=	1		W=	0,31	"=B3-B18-D18-C19"		
4	Aw	0,5	Ab=	0,6		B=	0,00	"=D3-B19-C18-D19"		
5										
6	Номера клеток в порядке прохождения с учётом индексации C++									
7		0	1	2						
8	0	1	2	3						
9	1	6	5	4						
10										
11	Заполнение поля краской									
12		0	1	2						
13	0	0,21	0,55	0,48		"=\$B\$4*ABS(COS(2*B9))"				
14	1			0,45		"=\$D\$4*ABS(SIN(B10))"				
15										
16	Учёт того, что в клетке может быть меньше 10 г									
17		0	1	2						
18	0	0,21	0,55	0,48		"=ЕСЛИ(B14<0,01;0,01;B14)"				
19	1			0,45						
20										
21		min_w=	0,21							
22		max_w=	0,48							
23						Белая краска закончилась в ячейке:		5		
24		min_b=	0,17			Черная краска закончилась в ячейке:		6		
25		max_b=	0,55							

Рис. 2. Результаты работы для случая Примера 1.

Пример 2. Расход краски небольшой и заданного количества краски хватило на покраску всего поля. Проведен анализ по поиску минимакса.

	A	B	C	D	E	F	G	H	I	J
1	Исходные параметры задачи									
2	M=	3	N=	2		Оставшаяся краска				
3	W=	1	B=	1		W=	0,86		"=B3-B18-D18-C19"	
4	Aw	0,1	Ab=	0,2		B=	0,67		"=D3-B19-C18-D19"	
5										
6	Номера клеток в порядке прохождения с учётом индексации C++									
7		0	1	2						
8	0	1	2	3						
9	1	6	5	4						
10										
11	Заполнение поля краской									
12		0	1	2						
13	0	0,04	0,18	0,10		"=\$B\$4*ABS(COS(2*B9))"				
14	1	0,06	0,08	0,15		"=\$D\$4*ABS(SIN(B10))"				
15										
16	Учёт того, что в клетке может быть меньше 10 г									
17		0	1	2						
18	0	0,04	0,18	0,10		"=ЕСЛИ(B14<0,01;0,01;B14)"				
19	1	0,06	0,08	0,15						
20										
21		min_w=	0,04							
22		max_w=	0,10							
23						Белая краска закончилась в ячейке:				5
24		min_b=	0,06			Черная краска закончилась в ячейке:				6
25		max_b=	0,18							

Рис. 2. Результаты работы для случая Примера 2.

Словесный алгоритм программы

На основе анализа контрольного примера, продумаем из каких шагов будет состоять решение задачи и наметим заголовки функций, реализующие соответствующие операции.

Поле клеток изобразим в виде двумерного массива $A[M][N]$. Чтобы зря не расходовать память под промежуточный массив (ячейки B8:D9), будем сразу заполнять его по ходу движения робота. Алгоритмически было бы удобно сначала заполнить клетки белой краской, затем – черной. Но это не соответствует процессу моделирования работы Робота, т.к. он не перескакивает через клетку, а идёт последовательно.

Разработаем необходимые «устройства» в работе Робота, которые будут реализованы соответствующими функциями.

Функция-координатор - ответственна за движение Робота: робот за один такт переходит из одной клетки в другую, функция будет вести нумерацию

индексов массива (квадратных метров поля) и порядковые номера n клеток. Заголовок функции:

double** Coordinator(int M, int N, double& W, double &B);

Она вернёт указатель на двумерный массив, примет его размеры N, M , а также объёмы краски в обоих баллонах – W, B . Последние два параметра передаются по ссылке, т.к. по мере продвижения Робота по полю эти объёмы будут изменяться.

Функция-манипулятор - ответственна за распыление краски: следит за наполняемостью баллонов краской, за количеством и цветом распыляемой краски. Для работы этой функции нужно передать текущие объёмы обоих баллонов и передать указание, по какому алгоритму распылять краску. Кол-во распыленной на клетке краски dv – возвращаемое значение, которое заносится в текущий элемент массива в вызывающей функции. Если та или иная краска закончилась, то манипулятор возвращает 0. Здесь потребуются две вспомогательные переменные: n – счётчик номера клетки, изменяющийся в ленточном порядке, и логическая переменная *color*: *true* – белый цвет, *color*: *false* -черный. Значение *color* изменяется на противоположное при переходе в следующую клетку. Если в баллоне краски меньше, чем dv , то так же возвращается 0 – распыления не происходит.

Заголовок функции:

double Manipulator(int n, double & V, double (*pf)(int));

Среди формальных параметров функции стоит указатель *(*pf)* на одну из функций распыления (1).

Функция-анализатор - ответственна за определение клетки с минимальными и максимальными расходами красок: зависит от цвета краски. Поскольку покраска начинается с белого цвета, то за него можно взять использовать булеву переменную *color*: 1 – белый, 0 – черный. Для определения индексов клеток, на которых краска закончилась, вводим в теле функции 4 переменные: $i0w, j0w, i0b, j0b$, ответственные за координаты ячейки массива, на которой краска закончилась. Аналогично, для определения индексов экстремальных значений матрицы вводим ещё 4 переменных: $imin, jmin, imax, jmax$. Анализатор работает уже после того, как Робот закончил работу, поэтому его можно вызвать после работы Координатора.

Заголовок функции:

void Analyzer(double** A, int M, int N, bool color);

Функции-распылители - аналог двух «микросхем», определяющих интенсивность и плотность краски от номера ячейки n , для каждого цвета своя «микросхема» – формулы (1а, 1б). Результат зависит от «настройки» амплитуды.

Заголовки функций:

double Spray_W(int n);//распылитель для белой краски - 1а

double Spray_B(int n);//распылитель для чёрной краски – 1б

Для того, чтобы не перегружать список формальных параметров, амплитуды A_w и A_b объявим на уровне глобальных переменных – это аналог настраиваемых до запуска Координатора микросхем Распылителя.

Функция-визуализатор – позволяет распечатать полученное распределение краски по полю. C++ позволяет печатать символы в цвете, но это сложно реализовать в среде VS Studio. Поэтому мы печатаем номер клетки с индексом: w – белая, b – чёрная.

Заголовок функции:

void Visualizer(double** A, int M, int N);

По мере того, как мы начали писать код, у нас возникла необходимость разработать ещё одну функцию, которая бы определяла по индексам массива номер соответствующей ему клетки. Для этого составили контрольный пример и по индексам клетки вывели формулу для вычисления номера n :

$M=3, N=4$

$i \backslash j$	0	1	2	3
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11

допустим, надо определить номер клетки с $i=1$ и $j=2 \rightarrow$ это 7

$n = i * N + j$, где j – индекс строки

для $i=1, j=2$ – номерная строка

$n = 1 * 4 + (2-1) = 4$

для $i=2, j=0$ – первая строка

$n = 2 * 4 + 0 + 1 = 9$

Рис. 3. Контрольный пример для определения номера клетки по индексам элемента массива

Функция – номер клетки имеет следующий прототип:

int n_cell(int i, int j, int N);

После разработки основных структурных элементов можно перейти к построению общего алгоритма решения задачи и детализации каждого модуля.

Визуализация алгоритма задачи с помощью блок-схем

Стратегию решения алгоритма представим в виде блок-схем некоторых, наиболее сложных или интересных с алгоритмической точки зрения функций (рис. 4-7). Начнём с представления главной программы.

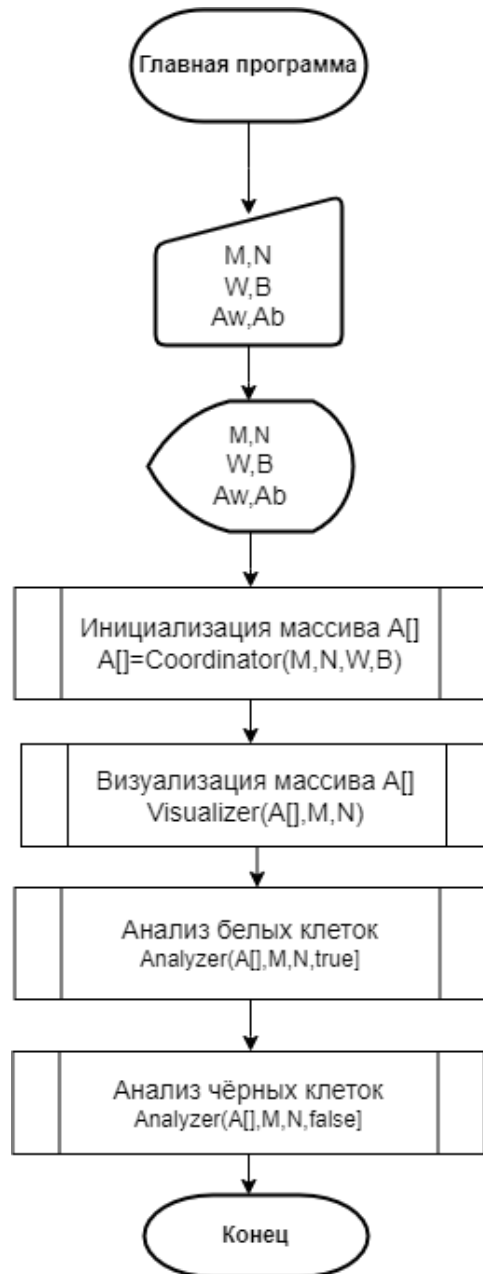


Рис. 4. Блок-схема главной программы задачи

Благодаря структурному подходу, главная программа имеет линейный алгоритм.

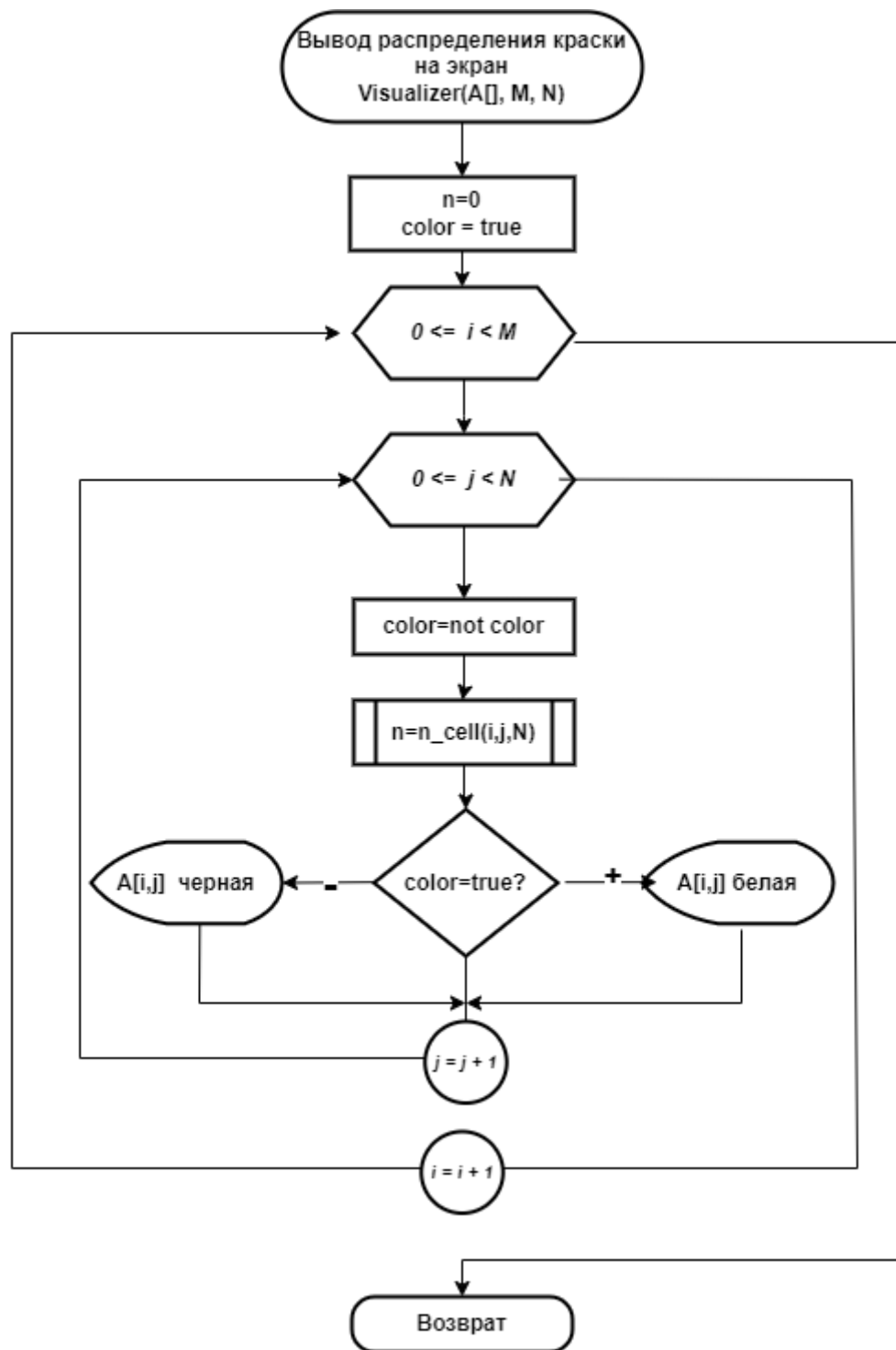


Рис. 5. Блок-схема функции визуализатора. Работает с переданным массивом A[] распределения краски. У неё пустой возврат, она после завершения работы просто передаёт управление вызывающей функции. Реализуется вывод на консоль данных о плотности и цвете краски в каждой клетке поля.

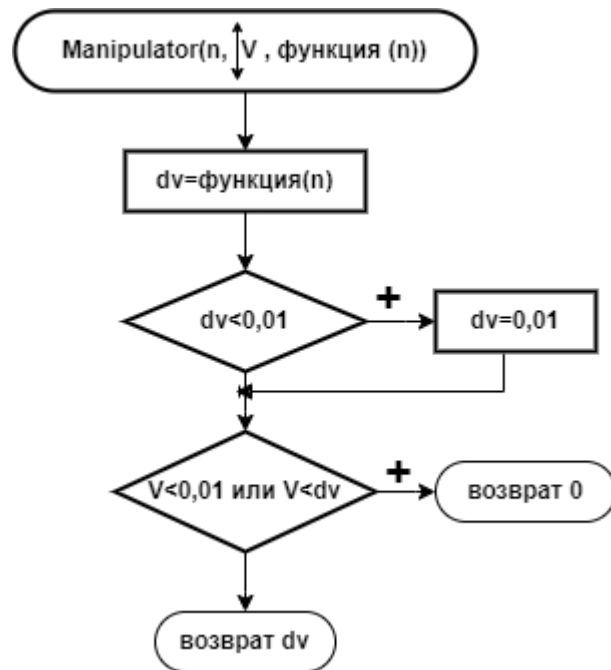


Рис. 6. Блок-схема функции манипулятора. У неё не пустой возврат: она возвращает в вызывающую функцию количество потраченной краски dv через оператор *return*, а также с помощью передачи по ссылке величины V - уменьшившийся объем краски в баллоне.

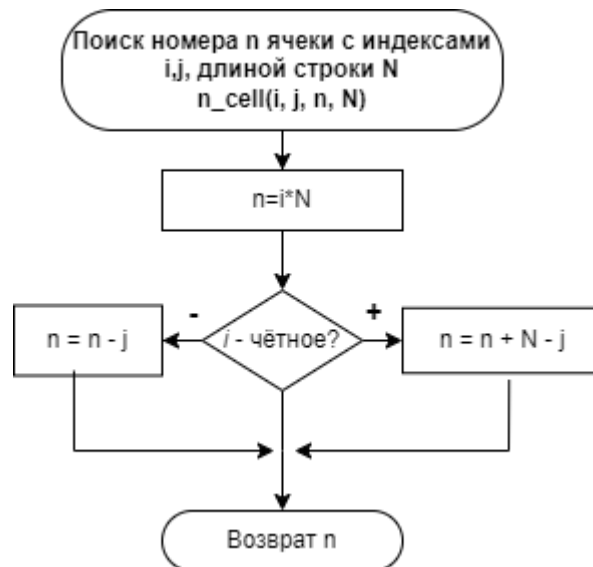


Рис. 7. Блок-схема определения номера ячейки n по индексами массива.

Алгоритм остальных функций построен на линейных алгоритмах, либо на классических алгоритмах поиска минимакса. Разработка кодов функций проекта производилась в среде Visual Studio 2022, их коды представлены в Приложении 1.

Анализ решения – сравнение с контрольным примером

Отладка и анализ разработанных программ проведен путем сравнения результатов работы программы с контрольными примерами, представленным на рис. 1 - 2. Скриншот решения, соответствующие и примеру 1 (Рис. 8), и примеру 2 (Рис. 9), показывает, что результаты совпадают.

Исходные данные задачи:		
Размеры поля 2х3		
Объемы белой и черной красок W= 1.0,B= 1.0		
Амплитуда расхода краски Aw= 0.50,Ab= 0.60		
Результат работы Робота		
n= 1w: 0.21	n= 2b: 0.55	n= 3w: 0.48
n= 6b: 0.00	n= 5w: 0.00	n= 4b: 0.45
После распыления в баллонах осталось:		
0.31кг белой краски и 0.00кг чёрной краски		
Белая краска: результаты анализа		
Минимальное значение 0.21 в ячейке n= 1		
Максимальное значение 0.48 в ячейке n= 3		
Краска закончилась на ячейке n= 5		
Черная краска: результаты анализа		
Минимальное значение 0.45 в ячейке n= 4		
Максимальное значение 0.55 в ячейке n= 2		
Краска закончилась на ячейке n= 6		

Рис. 8 Тестирование по контрольному примеру 1

Настройка параметров Робота		
Ввести размер поля - M N :2 3		
Ввести объемы баллонов - W B :1 1		
Ввести амплитуды для распылителей - Aw Ab :0.1 0.2		
Исходные данные задачи:		
Размеры поля 2х3		
Объемы белой и черной красок W= 1.0,B= 1.0		
Амплитуда расхода краски Aw= 0.10,Ab= 0.20		
Результат работы Робота		
n= 1w: 0.04	n= 2b: 0.18	n= 3w: 0.10
n= 6b: 0.06	n= 5w: 0.08	n= 4b: 0.15
После распыления в баллонах осталось:		
0.78кг белой краски и 0.61кг чёрной краски		
Белая краска: результаты анализа		
Минимальное значение 0.04 в ячейке n= 1		
Максимальное значение 0.10 в ячейке n= 3		
Черная краска: результаты анализа		
Минимальное значение 0.06 в ячейке n= 6		
Максимальное значение 0.18 в ячейке n= 2		

Рис. 9 Тестирование по контрольному примеру 2

Массовые расчёты

Массовые расчёты – это применение построенного нами алгоритма программы к таким входным данным, которые отличаются от контрольного примера. Массовые расчёты показывают, что, разработанный алгоритм решения задачи применим для некоторого класса задач, различающихся только вводимыми пользователем данными.

В *массовых расчётах* показано несколько вариантов исходных данных, которые проверяют *полноту программы*: при всех исходных значениях, результаты получаются верными. Следовательно, применяемый нами алгоритм решения удовлетворяет свойству массовости и построенный нами проект может быть применён для любых строк.

Вариант 1. Меняем размеры поля $M = 4, N=3$

Исходные данные задачи:

Размеры поля 4x3

Объемы белой и черной красок $W= 1.0, B= 1.0$

Амплитуда расхода краски $A_w= 0.10, A_b= 0.20$

Результат работы Робота

n= 1w: 0.04	n= 2b: 0.18	n= 3w: 0.10
n= 6b: 0.06	n= 5w: 0.08	n= 4b: 0.15
n= 7w: 0.01	n= 8b: 0.20	n= 9w: 0.07
n=12b: 0.11	n=11w: 0.10	n=10b: 0.11

После распыления в баллонах осталось:

0.60кг белой краски и 0.20кг чёрной краски

Белая краска: результаты анализа

Минимальное значение 0.01 в ячейке n= 7

Максимальное значение 0.10 в ячейке n= 11

Черная краска: результаты анализа

Минимальное значение 0.06 в ячейке n= 6

Максимальное значение 0.20 в ячейке n= 8

Рис. 10. Скриншот решения для варианта 1.

Программа работает и для других размеров поля.

Вариант 2. Изменяем кол-во краски в баллонах и амплитуды распылителей:

Исходные данные задачи:

Размеры поля 5x5

Объемы белой и черной красок $W= 2.0, B= 1.5$

Амплитуда расхода краски $A_w= 0.70, A_b= 0.50$

Результат работы Робота

n= 1w: 0.29	n= 2b: 0.45	n= 3w: 0.67	n= 4b: 0.38	n= 5w: 0.59
n=10b: 0.00	n= 9w: 0.00	n= 8b: 0.49	n= 7w: 0.10	n= 6b: 0.14
n=11w: 0.00	n=12b: 0.00	n=13w: 0.00	n=14b: 0.00	n=15w: 0.11
n=20b: 0.00	n=19w: 0.00	n=18b: 0.00	n=17w: 0.00	n=16b: 0.00
n=21w: 0.00	n=22b: 0.01	n=23w: 0.00	n=24b: 0.00	n=25w: 0.00

После распыления в баллонах осталось:

0.25кг белой краски и 0.02кг чёрной краски

Белая краска: результаты анализа

Минимальное значение 0.10 в ячейке n= 7

Максимальное значение 0.67 в ячейке n= 3

Краска закончилась на ячейке n= 9

Черная краска: результаты анализа

Минимальное значение 0.01 в ячейке n= 22

Максимальное значение 0.49 в ячейке n= 8

Краска закончилась на ячейке n= 10

Рис. 11. Скриншот решения для варианта 2.

В варианте 2 мы не только задали квадратное поле, но видим, что при меньшей амплитуде черной краски, она закончилась раньше- на 9-й клетке, а белая – лишь на 16-й. Запас в баллонах у белой был больше.

Таким образом, мы продемонстрировали, что при любом наборе исходных данных разработанная программа работает правильно.

Заключение

В данной работе была создана программа, с помощью которой моделируется функционал Робота по заданному алгоритму. Исследование состояло из нескольких этапов.

Для понимания сути задачи и для формирования алгоритма было составлено несколько контрольных примеров: были рассмотрены разные исходные данные для начальной загрузки баллонов с краской, а также разные скорости их распыления. Контрольный пример составлялся с помощью стандартных функций Excel, с помощью которых легко пересчитывать и проверять результаты.

На основе анализа контрольного примера была продумана стратегия решения задачи и те алгоритмические приёмы, которые можно применить для её выполнения.

Выбранная стратегия решения позволила разбить алгоритм на отдельные функциональные модули и реализовать преимущества структурного подхода.

Описание разрабатываемых функций включало следующие характеристики: назначение программы, список формальных параметров с описанием их назначения, возвращаемое значения и, кратко, логика работы программы.

Разработанный алгоритм позволил записать код программы на C++.

Завершающим этапом исследования было тестирование программы на основе контрольных примеров. Оно прошло успешно и позволило нам провести массовые расчёты, демонстрирующие и корректность, и полноту разработанного нами программного комплекса.

Таким образом, все поставленные в курсовой работе задачи были выполнены. Программа отлажена и может применяться на практике.

Библиографический список

Петрова Н.К. Конспект лекций «Алгоритмизация и программирование на C++». Электронный документ <https://lms.kgeu.ru/course/view.php?id=4695>.

Павловская Т. А. «Программирование на языке C++»: учебник для вузов / Национальный Открытый Университет «ИНТУИТ», 2016. Электронный документ <https://e.lanbook.com/book/100409?category=1557>.

Курсовая работа по дисциплине «Алгоритмизация и программирование»: методические указания / сост. Петрова Н.К., С.М. Куценко. – Казань: Казан. гос. энерг. ун-т, 2019. – 51 с.

Сидорина, Татьяна Самоучитель Microsoft Visual Studio C++ и MFC / Татьяна Сидорина. - М.: "БХВ-Петербург", 2014. - 848 с.

Линейная алгебра: программа, методические указания для студентов заочной формы обучения в сокращенные сроки направления подготовки 080100.62 "Экономика", профиля подготовки "Экономика предприятий и организаций" / сост. : Т. А. Григорян, Е. В. Липачева. - Казань : КГЭУ, 2013. - 64 с. - URL: <https://lib.kgeu.ru>. - 4625.

Высшая математика в упражнениях и задачах : учебное пособие для вузов ; в 2 ч / П. Е. Данко, А. Г. Попов, Т. Я. Кожевникова. - 6-е изд. - М. : ОНИКС: Мир и образование, 2005. - ISBN 5-488-00113-1. - Ч. 1, Ч.2. - 2005. - 416 с.

ПРИЛОЖЕНИЕ 1. Листинг программы

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cmath>
#include <windows.h>
#include <cstring>
using namespace std;
double** Coordinator(int M, int N, double& W, double &B); //Координатор
double Manipulator(int n, double & V, double (*pf)(int)); // Манипулятор
double Spray_W(int n); //распылитель для белой краски
double Spray_B(int n); //распылитель для чёрной краски
void Analyzer(double** A, int M, int N, bool color); // Анализатор
void Visualizer(double** A, int M, int N); // Визуализатор
int n_cell(int i, int j, int N); // определение номера ячейки по индексам массива

double Aw, Ab; // амплитуды распылителя

-----

int main()
{
    // Выделение памяти под необходимые переменные
    int N, M, iw, jw, ib, jb; // размер поля и координаты ячейки, на которой закончилась
    краска
    iw = jw = ib = jb = -1; // Начальные значения индексов незакрашенных клеток
    double **A, W, B, minW, maxW, minB, maxB; // Массив клеток поля, объёмы баллонов
    с белой и чёрной краской,
    //переменные для экстремальных значений
    распыленной краски
    setlocale(LC_CTYPE, "rus"); // вызов функции настройки локали для кириллицы

    // инициализация исходных данных
    printf("\n-----");
    printf("\n Настройка параметров Робота\n");
    cout << "Ввести размер поля - M N :"; cin >> M >> N;
    cout << "Ввести объёмы баллонов - W B :"; cin >> W >> B;
    cout << "Ввести амплитуды для распылителей - Aw Ab :"; cin >> Aw >> Ab;

    printf("\n-----");
    printf("\nИсходные данные задачи:\n");
    printf(" Размеры поля %ldx%ld\n", M, N);
    printf(" Объёмы белой и черной красок W=%4.1f,B=%4.1f\n ", W, B);
    printf(" Амплитуда расхода краски Aw=%5.2f,Ab=%5.2f ", Aw, Ab);
    printf("\n-----");
    // Запускаем Координатор
    A = Coordinator(M, N, W, B);
    Visualizer(A, M, N);
    printf("\nПосле распыления в баллонах осталось:\n %5.2fкг белой краски и %5.2fкг
    чёрной краски \n", W, B);
    Analyzer(A, M, N, true);
    Analyzer(A, M, N, false);
    printf("\n-----\n\n");
    system("pause");
    return 0;
}

-----

double** Coordinator(int M, int N, double & W, double & B)
{
    double ** T = new double * [M]; // Выделяем память под временный массив
    for (int i = 0; i < M; i++)
        T[i] = new double[N];
    int n=0;
    int color = -1; //black
    for (int i = 0; i < M; i++)
```

```

{
    if (!(i % 2))// для чётных строк
    {
        for (int j = 0; j < N; j++)
        {
            n++;
            color = -color;
            if (color > 0)
                T[i][j] = Manipulator(n, W, Spray_W);//заполняем белую
клетку
            else
                T[i][j] = Manipulator(n, B, Spray_B);//заполняем
чёрную клетку
            //printf(" i=%2d, j=%2d, T=%6.2f", i, j, T[i][j]);
            //cout << endl;
        }
    }
    else // для нечётных строк
    {
        for (int j = N - 1; j >= 0; j--)
        {
            n++;
            color = -color;
            if (color > 0)
                T[i][j] = Manipulator(n, W, Spray_W);//заполняем белую
клетку
            else
                T[i][j] = Manipulator(n, B, Spray_B);//заполняем
чёрную клетку
            //printf(" i=%2d, j=%2d, T=%6.2f", i, j, T[i][j]);
            //cout << endl;
        }
    }
    return T;
}

```

```

double Manipulator(int n, double & V, double (*pf)(int))//функция реализует
"распыление" краски в
клетку n по алгоритму, передаваемому через
//в
//указатель на функцию (*pf), корректируя кол-во
//
краски в баллоне V
{
    double dv = pf(n);
    if (dv < 0.01) dv = 0.01;
    if (V < 0.01 or V<dv) return 0; // в баллоне не хватает краски
    V -= dv;
    return dv;
}

```

```

void Visualizer(double** A, int M, int N)
{
    int n = 0;//номер ячейки
    cout << " \nРезультат работы Робота\n";
    bool color = 0;//black
    for (int i = 0; i < M; i++)
    {
        for (int j = 0; j < N; j++)
        {
            color = !color;
            n = n_cell(i, j, N);
            if (color)

```

```

        printf("\tn=%2dw:%5.2f",n, A[i][j]);
    else
        printf("\tn=%2db:%5.2f",n, A[i][j]);
    }
    cout << endl;
}
return;
}

-----

double Spray_W(int n)
{
    return Aw * abs(cos(2. * n));
}
double Spray_B(int n)
{
    return Ab * abs(sin(double(n)));
}

-----

void Analyzer(double** A, int M, int N, bool color)
{
    int in, jn, i0w, j0w, i0b, j0b, imin, jmin, imax, jmax, n;
    double minA, maxA;

    string sw = " Белая ", sb = " Черная ", s;
    if (color)
    {
        in = jn = 0; // выбор начальной клетки для анализа
        s = sw; // выбор цвета
    }
    else
    {
        in = 0; jn = 1; s = sb;
    }
    bool flag1 = 1, flag2 = 1; // флаг для вычисления первой пустой:
                                // опускается, если такая клетка есть
    minA = 1e32; maxA = 0;
    int j = -1;
    for (int i = in; i < M; i++)
    {
        if (j >= N) jn = j - N;
        for (j = jn; j < N; j += 2)
        {
            if (A[i][j] < minA and A[i][j] > 0)
            {
                minA = A[i][j]; imin = i; jmin = j;
            }
            if (A[i][j] > maxA and A[i][j] > 0)
            {
                maxA = A[i][j]; imax = i; jmax = j;
            }
            if ((flag1 or flag2) and abs(A[i][j]) < 0.01) // флаг включен, пока
не дошли до пустой клетки
            {
                if (flag1) {
                    flag1 = !flag1; // флаг выключаем и запоминаем индексы
первой пустой клетки
                    i0w = i; j0w = j;
                }
                if (flag2) {
                    flag2 = !flag2; // флаг выключаем и запоминаем индексы
первой пустой клетки
                    i0b = i; j0b = j;
                }
            }
        }
    }
}

```

```

    }
}

printf("\n %s краска: результаты анализа\n", s.c_str());
n = n_cell(imin, jmin, N); // определение номера ячейки
//вывод строковой переменной реализуется через метод c_str()
printf("\t Минимальное значение %6.2f в ячейке n= %d\n", A[imin][jmin], n);
n = n_cell(imax, jmax, N);
printf("\t Максимальное значение %6.2f в ячейке n= %d\n", A[imax][jmax], n);
if (color and !flag1)
{
    n = n_cell(i0w, j0w, N);
    printf("\t Краска закончилась на ячейке n= %d\n", n);
}
if (!color and !flag2)
{
    n = n_cell(i0b, j0b, N);
    printf("\t Краска закончилась на ячейке n= %d\n", n);
}

}

-----

int n_cell(int i, int j, int N)
{
    int n = (i * N);
    if (!(i % 2)) n += j + 1; else n += N - j;
    return n;
}

-----

```


Учебное издание

КУРСОВАЯ РАБОТА
ПО ДИСЦИПЛИНЕ «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»

Методические указания

Для студентов, обучающихся
по образовательной программе направления подготовки
09.03.03 Прикладная информатика

Составители

Петрова Наталья Константиновна
Куценко Светлана Мунавировна

Кафедра информатики и информационно-управляющих систем КГЭУ

Редактор И.В. Краснова
Компьютерная верстка И.В.Краснова

Подписано в печать 27.05.2019.
Формат 60x84/16. Гарнитура «Times». Вид печати РОМ.
Усл.-печ. л. 0,87. Уч.-изд. л. 0,43. Заказ № 245/эл.

Редакционно-издательский отдел КГЭУ,
420066, Казань, Красносельская, 51