**Аннотация**

Пояснительная записка включает 27 страниц,12 рисунков,13 источников.

КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ, ЯЗЫК C++. ПРОГРАММИРОВАНИЕ. СОЗДАНИЕ ТЕСТОВ.

Тема курсового проекта: «Разработка программы тестирования знаний».

Объектом курсового проекта является процесс тестирования знаний по дисциплине «Алгоритмизация и программирование».

Предметом является программа реализации алгоритма тестирования знаний с выбираемыми вопросами случайным образом из файла.

Цель работы – разработать алгоритм и программуна языке программирования С++.

Для достижения цели необходимо решить ряд задач:

1. Составить тест по дисциплине «Алгоритмизация и программирование»;
2. Освоить навыки работы с файлами
3. Разработать алгоритм программы;
4. Произвести отладку и тестирование программы..

**Содержание**

[Введение 5](#_Toc42088449)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc42088450)

[1.1 Развернутое описание задачи 6](#_Toc42088451)

[1.2 Область применения программы 7](#_Toc42088452)

[1.3 Описание исходных данных, переменных и констант 7](#_Toc42088453)

[2 ПРИНЦИП ТЕСТИРОВАНИЯ ЗНАНИЙ 9](#_Toc42088454)

[3 АЛГОРИТМ ВЫПОЛНЕНИЯ ПОГРАММЫ 11](#_Toc42088455)

[4 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ 17](#_Toc42088456)

[Заключение 21](#_Toc42088457)

[Список использованных источников 22](#_Toc42088458)

[Приложение А - Текст программы 24](#_Toc42088459)

# Введение

Курсовой проект был разработан в среде С++. Строго типизированный, компилируемый язык программирования общего назначения – это язык С++.

Также С++ поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования.

С++ создавался на основе языка С, и при этом до определенного момента сохранял с ним совместимость. Следовательно, С++ вобрал в себя всю скорострельность языка С. C++ используется во всех сферах деятельности программирования: от высоконагруженных систем до программирования микроконтроллеров. На С++ можно написать как web-сервер, так и игры, любые компьютерные программы, компоненты и так далее.

Дисциплина «Алгоритмизация и программирование» актуально в наше время. Уже с школы ученики изучают её на уроках информатики. На этой дисциплине рассматривается метод описания систем или процессов путём составления алгоритмов функционирования. Также, на занятиях развивается логика. Что касается представления алгоритмов, они бывают: словесные, структурно-стилизованные, графические и программные.

Как и всегда, после изучения чего-либо необходимо проверить уровень знаний после, или во время, прохождения курса. Одной из простых форм проверки знаний является – тестирование. Тестирование представляет собой вопрос, на который необходимо самостоятельно ответить.

Целью курсового проекта по разработке программы тестирования знаний является – разработать алгоритм и программуна языке программирования С++.

Суть курсового проекта: создание программы для тестирования знаний по дисциплине «Алгоритмизация и программирование».

# ПОСТАНОВКА ЗАДАЧИ

# Развернутое описание задачи

Для выполнения курсового проекты был выбран язык программирования С++, поскольку именно он рассматривается в рамках нашего обучения. А также, поскольку этот язык актуален в рабочей сфере.

Программная реализация представляет собой вывод в текстовой форме вопроса по дисциплине «Алгоритмизация и программирование» с вариантами ответа в консольном приложения, для тестирования знаний. Также, при выводе вопроса необходимо выбрать варианта ответа и после ответа на все вопросы, пользователя ждёт вывод о количестве верно/неверно отвеченных вопросов и процент верных ответов на тест.

В ходе работы над курсовым проектом необходимо выполнить такие условия, как:

* написать программу тестирования знаний студентов по дисциплине «Алгоритмизация и программирование»;
* вопросы из файла не должны повторяться;
* вопросы тестирования должны выбираться из файла случайным образом;
* при вводе ответа не входящим в диапазон вариантов ответа дать ещё попытку пользователю для ответа на вопрос;
* после выполнения теста программа должна осуществлять подсчёт верных/неверных ответов и вывести в консольном приложении результаты на экран.

# Область применения программы

Программа разработана для тестирования знаний учеников. Поскольку программа использует вопросы из файла, который можно редактировать, данную программу можно использовать в любой сфере, где необходима тестовая проверка знаний.

Главное, владеть знаниями как заполнять базу данных и какие правки нужно сделать в программе для регулирования количества вопросов в тесте.

# Описание исходных данных, переменных и констант

Описание исходных данных, переменных и констант представлено ниже.

1. srand(time(NULL)) - возвращает количество секунд, которые прошли с момента запуска WindowsXP, для использования случайных чисел;
2. vector<unsignedint>Repetition\_Check - служит для случайного выбора вопросов без повторов;
3. unsignedintQuantity\_Of\_Correct\_Answers - принимает значение 0, это количество правильных ответов;
4. UnsignedintRand\_Question - отвечает за номер вопроса, которые выводится случайным образов;
5. UnsignedintQuantity\_Of\_Questins - количество вопросов в базе;
6. Inti – целочисленная переменная;
7. Charsimbol - служит для навигации по файлу. Чтобы определить, где начинается вопрос, заканчивается и какой ответ. Поочередности хранит в себе каждый символ;
8. ifstreamfin - нужно для вывода текста из файла;
9. unsignedintTempIndex - в этой переменной случайным образом появляется индекс, чтобы выбрать какой из вопрос нужен сейчас;
10. Intendlinecheck - отвечает за ввод данных онлайн;
11. Сharbuffer – локальный буфер;
12. UnsignedintYour\_Answer - вариант отвечает, который выбрал пользователь;
13. UnsignedintCorrect\_Answer - отвечает за вариант ответа, который является верным;
14. FloatQuantity\_Of\_Correct\_Answers - замена типа переменной для расчёта процентов;
15. FloatQuantity\_Of\_Questins - замена типа переменной для расчёта процентов.

# ПРИНЦИП ТЕСТИРОВАНИЯ ЗНАНИЙ

По определению, педагогический тест представляет собой систему взаимосвязанных заданий специфической формы, определенного содержания, возрастающей сложности, позволяющих надежно и валидно оценить знания и другие интересующие педагога характеристики личности.

К целям тестирования относится формирование: знаний, понятий и системы понятий, терминов, законов и гипотез, научных фактов, моделей и методов.

Всего существует несколько форм тестирования, которые представлены на рисунке 1.1.

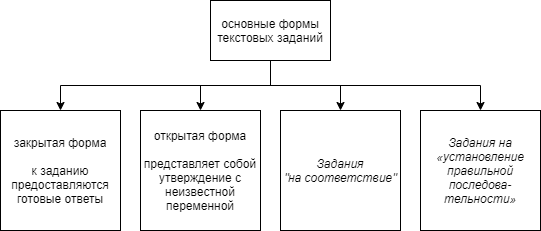


Рисунок 1.1 – Формы тестовых заданий

При составление теста, в основном, была использована закрытая форма тестирования. Основная цель заданий закрытой формы - быстро проверить ориентированность студента в данной учебной дисциплине по профориентированной работе для самопроверки. Потому, закрытая форма удобна при составления теста на базе С++.

Также необходимо отметить, что тест проводится онлайн и все результаты рассчитывает машина. Что влечет за собой ряд преимуществ:

1. Экономия времени. Не нужно писать отдельный тест для каждого курса — вы можете создать экзамен просто выбирая вопросы из уже созданной базы или воспользоваться шаблоном, который будет легко изменить для каждого нового курса.
2. Онлайн тесты эффективны и надежны. Чтобы для каждого ученика оценка была объективной, важно предотвратить вариант списывания. Потому очень удобен вариант создания теста, где вопросы в случайном порядке выводятся на экран из базы данных. Также, многие тестирования не позволяют скопировать текст из диалогового окна, что усложняет списать ответ из интернета.
3. Автоматическая система подсчета баллов. Когда машина сама считает баллы, процент неверного подсчёта баллов близится к нулю. Также, у машины нет каких-то взаимоотношений с учениками, потому машина максимально объективно оценивает ответы студентов.

Говоря о достоинствах, всегда нужно упоминать и недостатки. Согласно исследованиям, результаты прохождения тестов будут ниже у учеников, которые проходят опрос онлайн, по сравнению с теми, кто пользуется бумагой и ручкой. Но эта особенность скорее связана с тем, что многие просто не привыкли к такой форме проверки знаний.

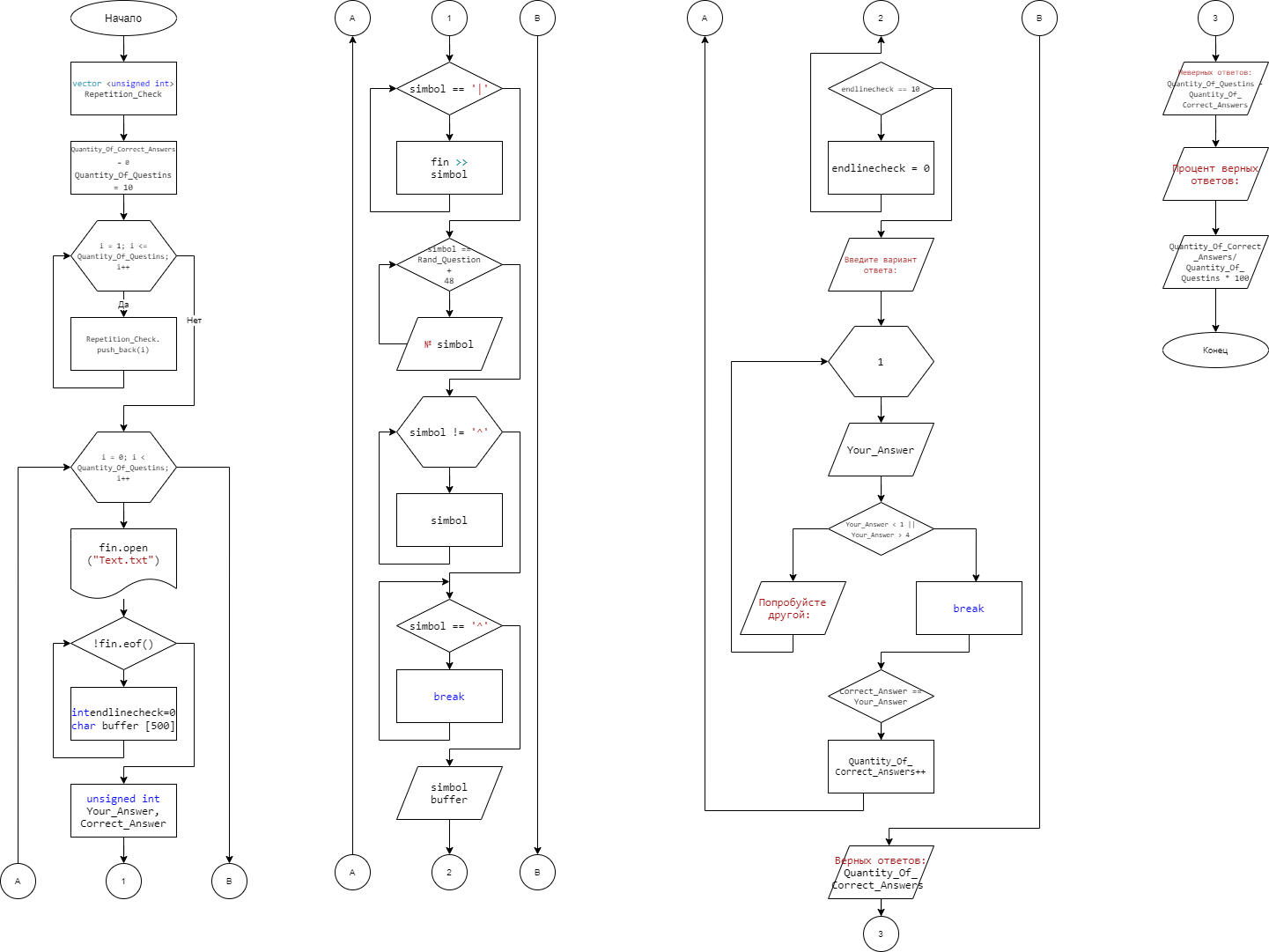
# АЛГОРИТМ ВЫПОЛНЕНИЯ ПОГРАММЫ

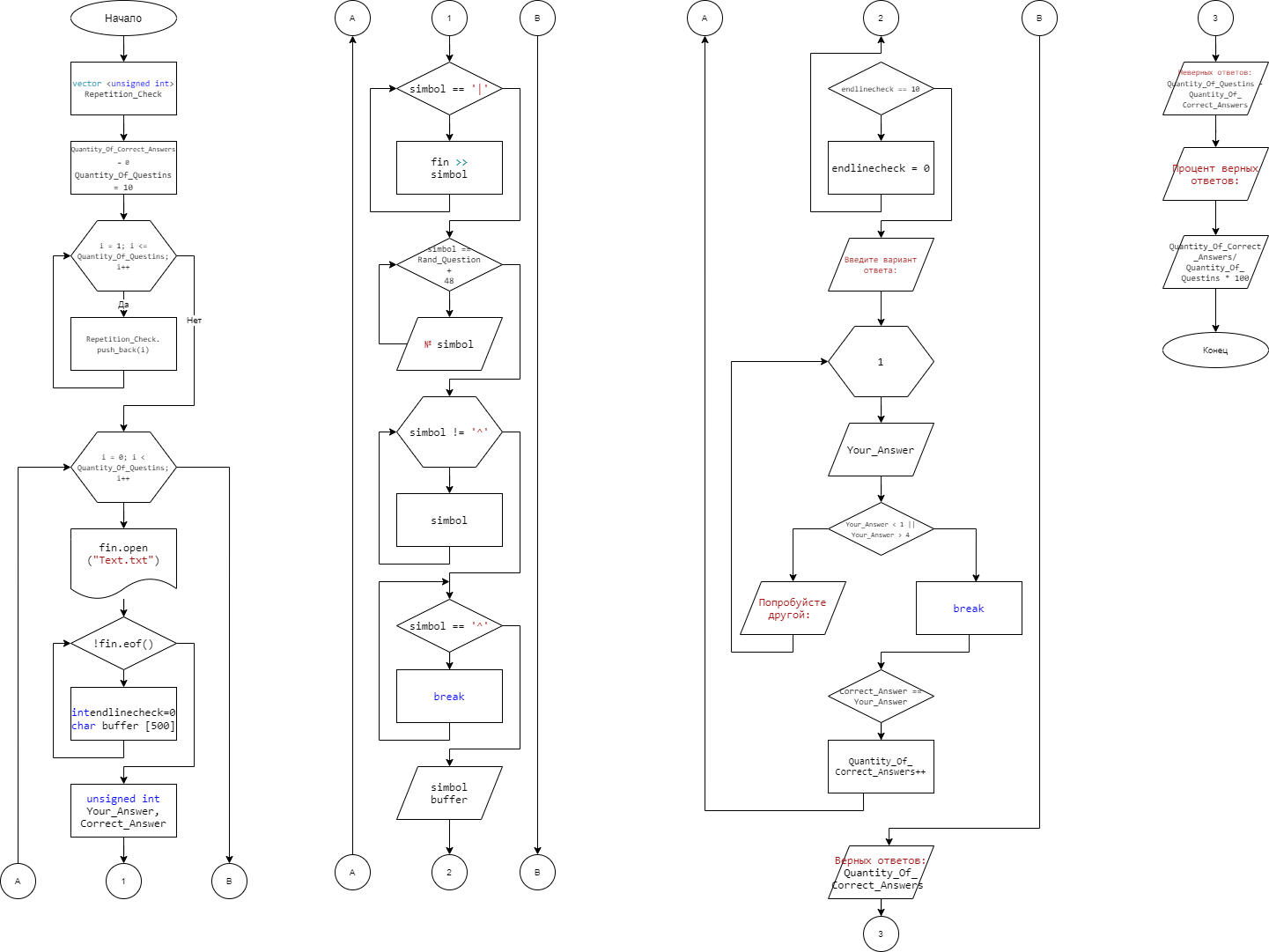
Первым этапом в составлении программы является – составление блок-схем. Блок-схема это наглядный способ того, как работает программа. Для выполнения курсового проекта была составлена блок-схема, которая представлена на рисунке 2.1 - 2.4.

Алгоритм выполнения функции intmain ():

1. Создается вектор, который возвращает количество секунд, которые прошли с момента запуска WindowsXP, для использования случайных чисел.
2. ИнициализацияпеременныхQuantity\_Of\_Correct\_Answers = 0, Rand\_Question, Quantity\_Of\_Questins = 10.
3. Еслиусловиеinti = 1; i<= Quantity\_Of\_Questins; i++, тогдадобавляемвконецвектораRepetition\_Check.push\_back(i), дляпрогонкикаждоговопроса.
4. Открываем текстовый файл «Text.txt».
5. Случайным образом рандомим индекс, для нахождения вопроса в файле. В переменную записываем номер вопроса из листа с вопросами. Удаляем из списка индекс, который до этого зарандомили, что бы ещё раз не вышел этот вопрос.
6. Создание условия while (!fin.eof()). Когда программа читает файл, принимает значение 0, когда файл весь прочитан, становится =1 и тогда цикл закрывается.
7. Создание локального буфера данных размером 500.
8. Создаётся конструкция, для возможности вписывания пользователем ответа. Your\_Answer отвечает за верный вариант ответа, Correct\_Answer за вариант ответа, который выбрал пользователь.
9. if (simbol == '|') поиск символа, программа ищет начало вопроса. После нахождения начала вопроса программы выводит «№» и то, что стояло сразу после '|'. Остальной текст переходит в локальный буфер. И также выводится на консоль.
10. if (simbol == '^') поиск символа, для нахождения в файле верного ответа на вопрос. После знака ^ сразу же, без отступов, пишется номер верного варианта ответа.
11. Ввод ответа пользователя.
12. Если ответ пользователя не входит в диапазон, тогда дается ещё попытка ответить на вопрос, пока пользователь не выберет вариант ответа из предложенных в вопросе.
13. После ввода ответа идёт сравнения ответа пользователя и верный ответ на вопрос. Если эти значения равны, тогда Quantity\_Of\_Correct\_Answers++.
14. Все циклы с пункта 6 повторяются столько раз, сколько есть вопросов в файле.
15. После прочтения всех вопросов в файле программа считает результаты тестирования и выводит их в консоль.
16. Верных ответов:, Неверных ответов:, Процент верных ответов:. После вывода на экран результатов тестирования и нажатия на любую клавишу программа закрывается.

Изучив приложение А, можно ознакомиться с самим текстом программы, в котором прописаны подробные комментарии.

Рисунок 2.1 – Первая часть блок-схемы

Рисунок 2.2 – Вторая часть блок-схемы

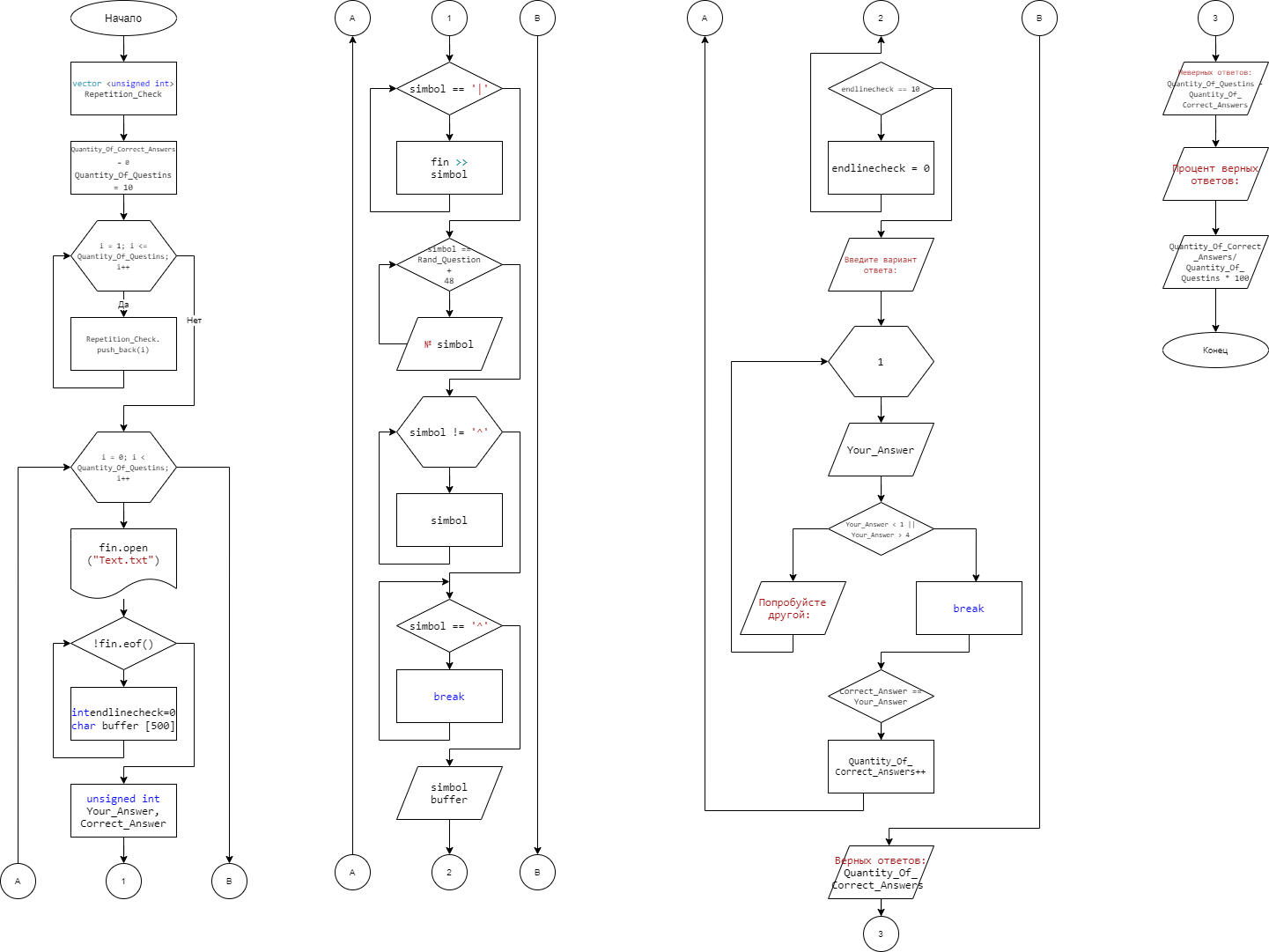
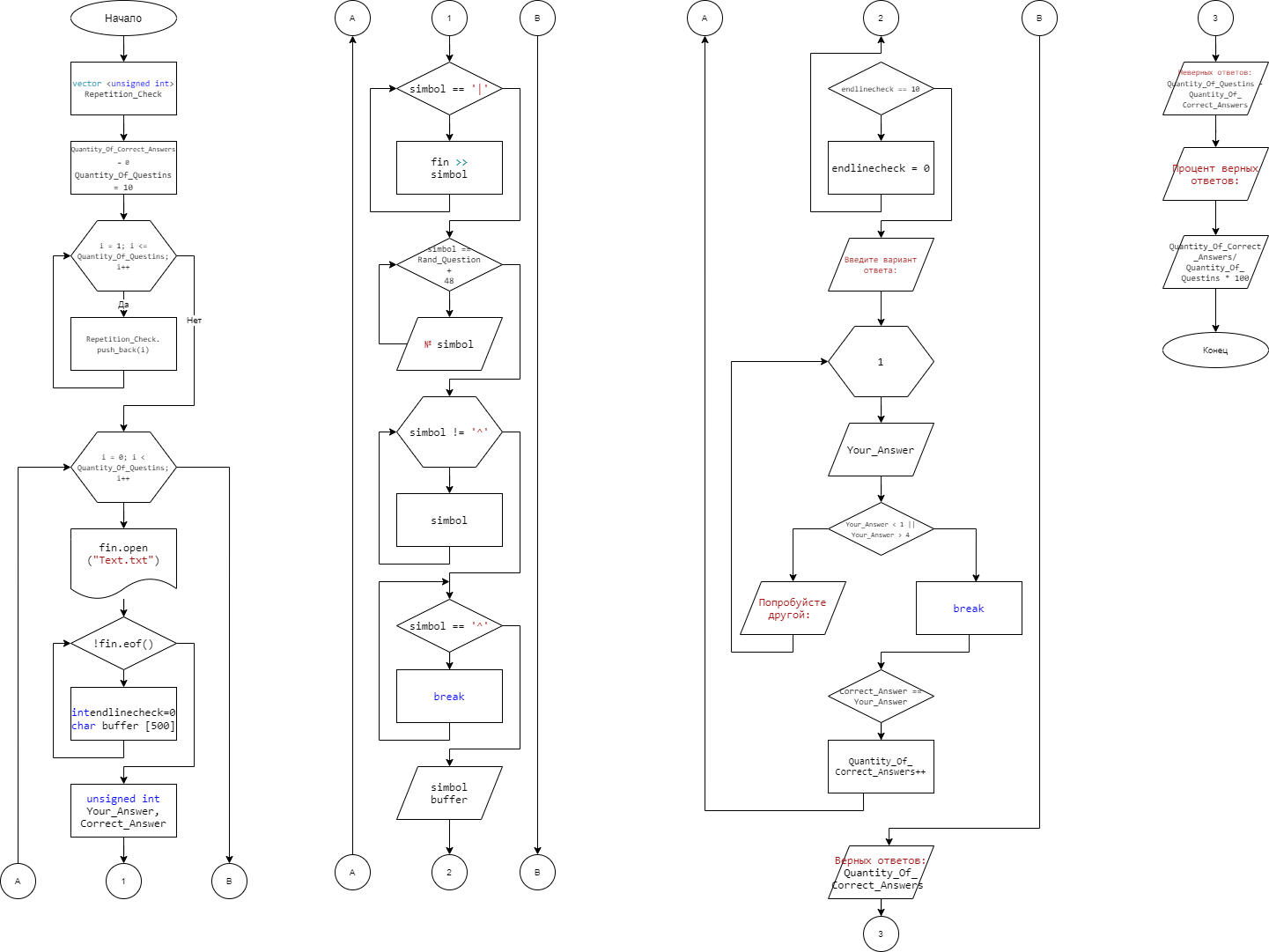


Рисунок 2.3 – Третья часть блок-схемы

Рисунок 2.4 – Четвёртая часть блок-схемы

# ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ

При открытии программы в консольном приложении сразу открывается вопрос. Пример представлен на рисунке 4.1.

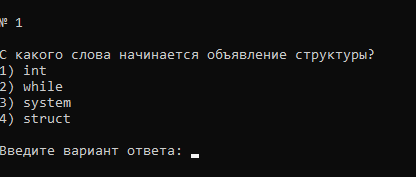


Рисунок 4.1 – Вывод первого вопроса

Далее пользователь вводит ответ, который считает нужным. Ответим сейчас верно. При вводе ответа в консоль появляется сразу же следующий вопрос. Пример представлен на рисунке 4.2.

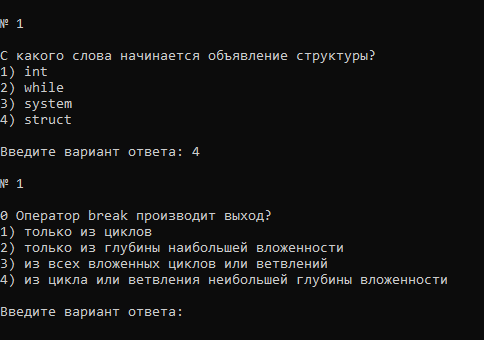


Рисунок 4.2 – Вывод второго вопроса

Всего в каждом вопросе 4 варианта ответа. При вводе варианта ответа входящей в область ответа, вариант ответа уже не изменить и программа выведет на экран следующий вопрос. Если пользователь вводит ответ не входящий в эту область, тогда программа даёт ещё шансы ввести ответ, пока пользователь не введет ответ в диапазоне от 1 до 4. Данный процесс представлен на рисунке 4.3.

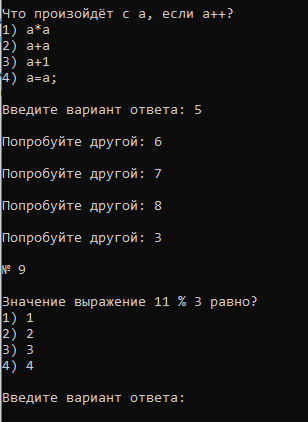


Рисунок 4.3 – Иной ответ

После ответа на все вопросы программа считает, сколько было верных, неверных ответов и процент выполнения теста. Представлено на рисунке 4.4.

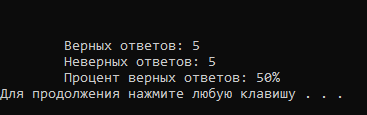


Рисунок 4.4 – Результат тестирования

После проведения отладки и тестирования не было выявлено необходимости в создании временных рамок для ответа пользователя на вопрос.

В тест можно вносить дополнительно свои вопросы. Верно отредактировав документ с вопросами. На рисунке 4.5 представлен пример, как нужно записывать новые вопросы, чтобы программа их понимала, выводила на экран и могла найти верный ответ.

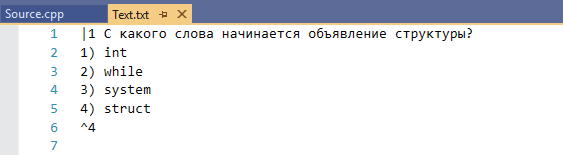


Рисунок 4.5 – Пример записи новых вопросов

Перед началом вопроса обязательно нужно поставить « | », благодаря этому символу программа находит начало вопроса. После этого символа, без отступов, ставится номер вопроса. Дальше, сделав отступ можно писать текст вопроса и варианты ответов. После окончания описания вопроса нужно сказать верный вариант ответа после символа « ^ »,тогда программа сможет найти верный ответ и сравнить его с ответом пользователя.

При добавлении вопросов необходимо заменить несколько значений в самой программе. Эти значения представлены на рисунке 4.6. В данном случае в текстовой файле 10 вопросов, потому эти переменные и приравниваются 10. В случае увеличения или уменьшения вопросов, необходимо заменить на то количество, сколько вопросов есть в текстовом файле.

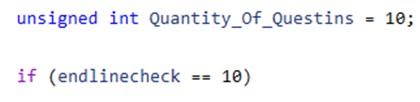


Рисунок 4.6 – Замена значений при изменении количества вопросов

Если идёт изменение количества вариантов ответов, в программе также нужно сделать замену там, где показано на рисунке 4.7. Значение 4 нужно будет заменить, на то количество вариантов ответов, которое есть в программе.



Рисунок 4.6 – Замена значений при изменении количества вариантов

Пример описания разработки кода программы:

4.1 Описание программы

На первом этапе происходит создание файла. Затем указывается размерность массива с интервалами генерации случайных чисел (рисунок 8).

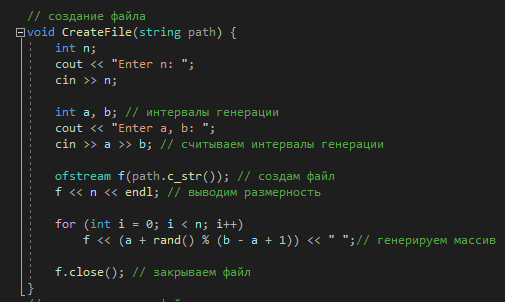


Рисунок 8– Создание файла

Далее считывается размер файла, значения массива из файла (рисунок 9).

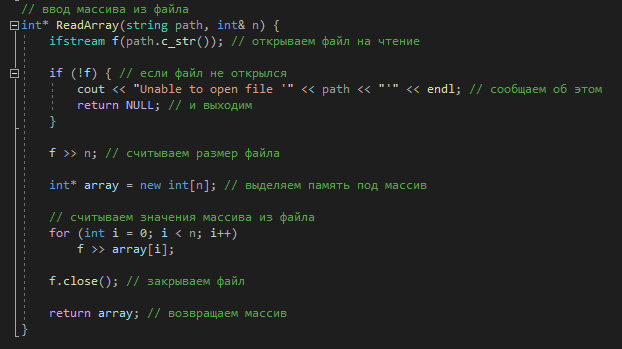


Рисунок 9 – Ввод массива из файла

Далее идет вывод массива и вывод массива в файл (рисунок 10).

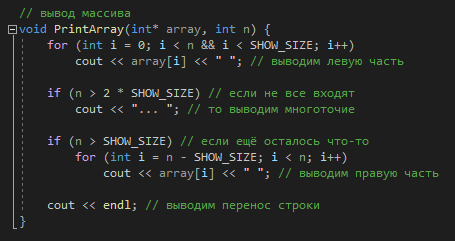


Рисунок 10– Вывод массива

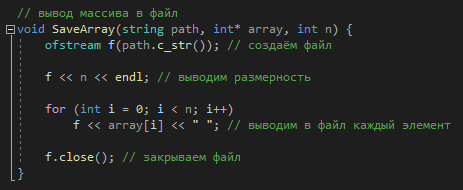


Рисунок 11 – Вывод массива в файл

Затем оба массива сортируются двумя методами (перестановкой, выбором), замеряя время сортировки. Затем выводятся отсортированные массивы и затем освобождается выделенная память (рисунок 12).

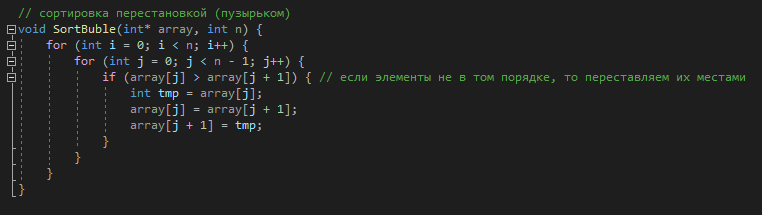


Рисунок 12– Сортировка перестановкой

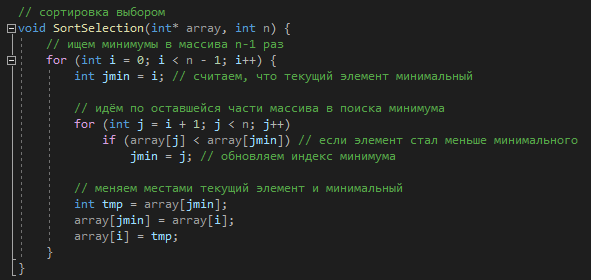


Рисунок 13– Сортировка выбором

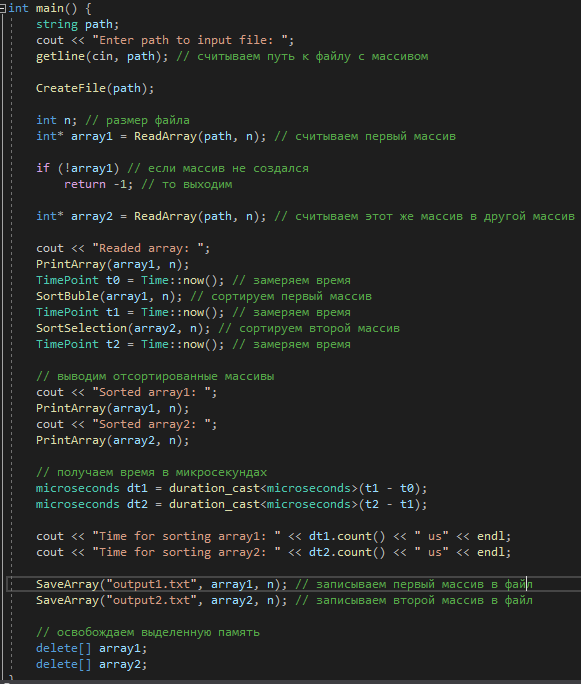


Рисунок 14 – Основная программа

## 5 ОТЛАДКА И ТЕСТИРОВАНИЕ ПРОГРАММЫ

5.1 Отладка и тестирование программы

На первом этапе вводим название файла и создаем его, будем использовать массив из 20 элементов (рисунок 15).

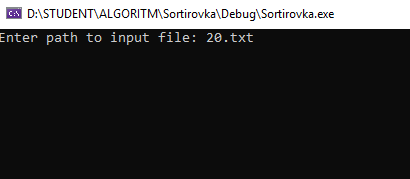


Рисунок 15 – Создание файла

На втором этапе вводим количество элементов массивов (рисунок 16).

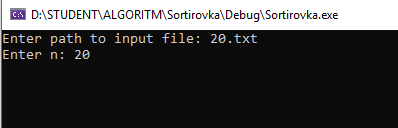


Рисунок 16 – Размерность

В третьем этапе мы указываем интервалы генерации случайных чисел массива (рисунок 17).

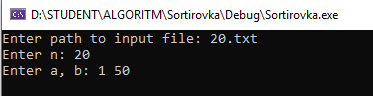


Рисунок 17 – Интервалы

После этого программа сортирует оба массива двумя методами сортировки (перестановкой, выбором), при этом указывает время, отведенное на каждую из сортировок (рисунок 18).

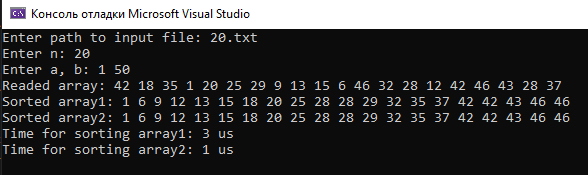


Рисунок 18 – Отсортированные массивы

Файлы с отсортированными массивами создаются в папке с нашим cpp файлом (рисунок 19).

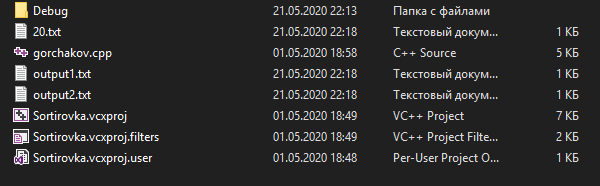


Рисунок 19 – Операция создания файла

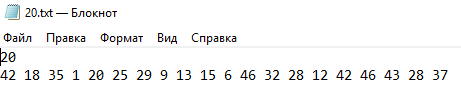


Рисунок 20 – Исходный массив

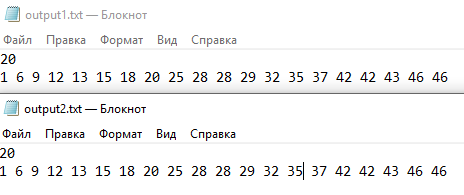


Рисунок 21 – Отсортированные массивы двумя методами

ответов

# Заключение

В результате работы, курсового проекта, были использованы навыки поиска необходимой информации, изучения и анализа полученных материалов, нахождения путей решения поставленных задач.

В результате выполнения курсового проекта были выполнены следующие задачи:

* изучены и составление теста по дисциплине «Алгоритмизация и программирование»;
* разработаны алгоритма программы и вывод результата в консольном приложении;
* отлаженапрограмма какая?;
* протестирована программа Какая?.

# Список использованных источников

1. Структура, содержание, оформление и защита самостоятельных работ студентов [Электронный ресурс] : методические указания / СибАДИ, Кафедра "Прикладная информатика в экономике" ; сост.: Л. И. Остринская, С. Ю. Пестова
2. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. – СПб.: Питер, 2011. – 461 с.
3. Программирование и основы алгоритмизации: Учеб.пособие./ Макаров В.Л. - СПб.: Питер, 2013. – 130 с.
4. Муртузалиева А.С. Условия оптимизации алгоритмической подготовки будущего педагога профессионального обучения (на примере отрасли «информатика, вычислительная техника и компьютерные технологии»).- Дис. насоиск. канд. пед. наук, Махачкала, 2011.
5. ПРОГРАММИРОВАНИЕ. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ/ Парфилова Н.И., учебник для студентов высших учебных заведений, обучающихся по направлению подготовки "Информатика и вычислительная техника"] / Н. И. Парфилова, А. Н. Пылькин, Б. Г. Трусов ; под ред. Б. Г. Трусова. Москва, 2012.
6. Алгоритмизация и программирование/ Трофимов В.В., Павловская Т.А.Учебник / Москва, 2018.
7. Калинина В.В., Пушкарева Т.П., Степанова Т.А. Развитие алгоритмического стиля мышления при обучении программированию в вузе // Теоретические и практические аспекты психологии и педагогики. Уфа: Аэтерна, 2015.
8. Стась А.Н. , Долганова Н.Ф. Развитие алгоритмического мышления в процессе обучения будущих учителей информатики// Вестн. Томского гос. пед. ун-та, 2012.
9. MicrosoftBuild. Написание модульных тестов для C/C++ в VisualStudio [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/visualstudio/test/writing-unit-tests-for-c-cpp?view=vs-2019> (дата обращения: 12.05.2020)
10. Первушина Н.А. Успешность визуализации информации в процессе обучения // Научно-педагогическое обозрение, 2013.
11. Методика составлений тестовых заданий [Электронный ресурс]. – Режим доступа: <http://elar.rsvpu.ru/bitstream/123456789/20133/1/Mamay_2001.pdf> (дата обращения: 20.04.2020)
12. Положение о самостоятельной работе студентов [Электронный ресурс]. – Режим доступа: <http://www.sibadi.org/about/officialdocument/36/262/?sphra-se_id=71060> (дата обращения 10.04.20)
13. MyOwnConference[Электронный ресурс]. – Режим доступа:<https://myownconference.ru/blog/index.php/online-tesirovanie/> (дата обращения 27.05.2020)

# Приложение А - Текст программы

#include<iostream> //Библиотека ввода и вывода

#include<fstream> //Библиотека для работы с файловыми потоками

#include<ctime>

//Библиотека для использования системного времени, чтобы выходило случайное число

#include<vector> //Библиотека для использования векторов

usingnamespacestd;

intmain() //Точкавхода

{

srand(time(NULL));

//Для случайной последовательности случайных чисел в определённом промежутке

setlocale(LC\_ALL, "Rus"); //Перевод на русский

vector<unsignedint>Repetition\_Check; //Выборслучайноговопросабезповтора

unsignedintQuantity\_Of\_Correct\_Answers = 0, Rand\_Question, Quantity\_Of\_Questins = 10;

//Инициализация переменных: кол.верн. ответов, номер вопроса

(который выбирает рандом), кол.воп.

//При изменении количества вопросов, изменить значение Quantity\_Of\_Questinsна

кол-во вопросов

for (inti = 1; i<= Quantity\_Of\_Questins; i++)

Repetition\_Check.push\_back(i);

//Добавляем в конец вектора i, для прогонки каждого вопроса

charsimbol; //Служит для записи символов

for (inti = 0; i<Quantity\_Of\_Questins; i++)

//Алгоритм для открытия текстового файла и работы с ним

{

ifstreamfin; //Служит для открытия текстового файла

fin.open("Text.txt"); //Открытие текстового файла

unsignedintTempIndex = 0 + rand() % (Repetition\_Check.size());

//Случайным образом рандомим индекс, для нахождения его в файле

Rand\_Question = Repetition\_Check[TempIndex];

//В переменную записываем номер вопроса из листа с вопросами

Repetition\_Check.erase(Repetition\_Check.begin() +TempIndex);

//Удаляет из списка индекс, который до этого зарандомили,

что бы ещё раз не вышел этот вопрос

while (!fin.eof())

//Констурция нужна, чтобы когда программа прочла файл приняла значение =1

и вышла из цикла

{

intendlinecheck = 0;

//Отвечает за то, чтобы мы могли прописывать онлайн ответы

char buffer[500]; //Созданиелокальногобуфера

unsignedintYour\_Answer, Correct\_Answer;

//Your\_Answer отвечает за верный вариант ответа

//Correct\_Answer за вариант ответа, который выбрал пользователь

fin>>simbol;

if (simbol == '|') //Поиск символа, программа ищет начало вопроса

{

fin>>simbol;

if (simbol == Rand\_Question + 48)

//+48 чтобы получить char

{

cout<<endl<<"№ ";

cout<<simbol<<endl<<endl;

//Выписывает номер вопроса в консоль

while (simbol != '^')

//Поиск символа, программа ищет ответ на вопрос

{

fin>>simbol;

if (simbol == '^')

break;

cout<<simbol;

//Выводсимвола

fin.getline(buffer, 500);

//Оставщуюсястрокузаписываемвбуфер

cout<<buffer<<endl;

//Вывод в консоль самого вопроса

endlinecheck++;

//Прибавляет +1 когда вписывают ответ

if (endlinecheck == 10)

//Считает сколько раз вписывает пользователь ответ

{

cout<<endl;

endlinecheck = 0;

}

}

fin>>Correct\_Answer;

//Вариант ответа, который выберет пользователь

cout<<endl<<"Введите вариант ответа: ";

while (1)

{

cin>>Your\_Answer;

if (Your\_Answer< 1 || Your\_Answer> 4)

//Если ответ не входит в диапазон вариантов, тогда:

cout<<endl<<"Попробуйстедругой: ";

else

break;

}

if (Correct\_Answer == Your\_Answer)

//Сравнивание ответа пользователя и верного ответа

Quantity\_Of\_Correct\_Answers++;

//Если пользователь ответил верно, тогда прибавляется +1

}

}

}

}

cout<<endl<<endl<<endl<<"\tВерныхответов: "<<Quantity\_Of\_Correct\_Answers<<

endl<<"\tНеверныхответов: "<<Quantity\_Of\_Questins -Quantity\_Of\_Correct\_Answers<<

endl<<"\tПроцентверныхответов: "<<float(Quantity\_Of\_Correct\_Answers) / float(Quantity\_Of\_Questins) \* 100 <<"%"<<endl;

//Вывод результатов тестирования

system ("pause");

}