МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образование учреждение высшего образования

«Курганский государственный университет»

Кафедра «Программное обеспечение автоматизированных систем»

РФ КГУ 09.03.04. КР24.360091 06

**КУРСОВАЯ РАБОТА**

**по дисциплине**

**«Алгоритмы и структуры данных»**

**Руководство программиста**

Листов 13

# АННОТАЦИЯ

Документ содержит сведения о программе, функциональное назначение, описание логической структуры, используемых технических средств, входных и выходных данных.

Документ предназначен для программистов.

# СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc184467918)

[СОДЕРЖАНИЕ 3](#_Toc184467919)

[1 ОБЩИЕ СВЕДЕНИЯ 4](#_Toc184467920)

[1.1 Программное обеспечение 4](#_Toc184467921)

[1.2 Языки программирования 4](#_Toc184467922)

[2 ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ 5](#_Toc184467923)

[3 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ 6](#_Toc184467924)

[4 ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ 7](#_Toc184467925)

[5 ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ 8](#_Toc184467926)

[6 ЛИСТИНГ АЛГОРИТМА 9](#_Toc184467927)

[7 ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММНОГО КОМПЛЕКСА 11](#_Toc184467928)

[8 ОПИСАНИЕ СТРУКТУР ДАННЫХ 12](#_Toc184467929)

# ОБЩИЕ СВЕДЕНИЯ

## **Программное обеспечение**

Данная программа выполняется под управлением операционной системы Windows. Для ее выполнения необходимо запустить файл Coursework\_AaSD-repos.exe

## **Языки программирования**

Для реализации программного продукта был выбран язык программирования C++

# ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

Разработанный программный продукт предназначен для формализации метода сортировки случайного массива с помощью Timsort

Программа предназначена для решения элементарных задач обработки данных, таких, как редактирование и сортировка.

Программа осуществляет следующие возможности:

1. Отображение титульного листа с кнопками навигации и справкой
2. Создание массива случайных чисел с помощью обратной функции Брэдфорда
3. Сортировка массива методом Timsort
4. Отображение отсортированного массива
5. Вывод количества времени выполнения сортировки, количества перестановок и сравнений.
6. Ввод количество выборок пользователем для анализа уравнения связи.
7. Автоматическая генерация массивов для каждой итерации размером от 9000 до 50000 элементов и выполнение сортировки
8. Вывод данных в таблицу DataGridView. Номер, Время, Размер массива.
9. Построение точечного графика зависимости времени сортировки от размера массива
10. Определение коэффициентов уравнения регрессии методом наименьших квадратов
11. Расчет коэффициента корреляции, детерминации, эластичности и бета-коэффициента
12. Построение линии регрессии на графике.

# ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

На рисунке 1 приведено описание логической структуры

**InfoForm**

buttonExit\_Click()

linkLabel1\_LinkClicked()

**MainForm**

buttonSort\_Click()

buttonEqConnection\_Click()

buttonInfo\_Click()

buttonExit\_Click()

**MHKForm**

buttonExit\_Click()

buttonExit\_Click()

buttonMHK\_Click()

buttonClear\_Click()

buttonCalc\_Click()

sample\_size\_TextChanged()

EqForm\_Load()

buttonExit\_Click()

buttonSort\_Click()

buttonClear\_Click()

textBoxN\_TextChanged()

SortForm\_Load()

**SortForm**

**EqForm**

Рисунок 1 – Логическая структура.

# ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ

Входные данные:

1. Массив случайных чисел, создаваемый с помощью обратной функции Брэдфорда
2. Количество элементов массива.
3. Коэффициенты функции Брэдфорда
4. Количество выборок для проведения эксперимента.

Выходные данные:

1. Отсортированный массив случайных чисел
2. Количество элементов
3. Время выполнения алгоритма
4. Количество сравнений.
5. Количество перестановок
6. Таблица исходных статистических данных, отображающая размер массива и время выполнения сортировки.
7. График зависимости времени выполнения сортировки от размера массива.
8. Рассчитанные статистические коэффициенты
9. Линия регрессии, построенная на основе данных эксперимента и отображенная на графике зависимости.

# ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ

Для функционирования программы требуется ЭВМ со следующими характеристиками:

1. Операционная система: Windows 7 или более новые версии, Windows 8.1, Windows 10, Windows 11
2. Процессор Intel Pentium 4 или эквивалентный AMD с тактовой частотой не менее 3.0 ГГц.
3. Наличие не менее 2 Гб оперативной памяти
4. Свободное пространство на жёстком диске 500 Мб для файлов приложения.

# ЛИСТИНГ АЛГОРИТМА

#include "math.h"

#include <stdlib.h>

#include <ctime>

#include <algorithm>

double get\_Bradford(double A, double B, double C) {

int r = rand();

double u = (double)r / ((double)RAND\_MAX + 1);

double par = ((B - A) \* (pow((C + 1), u) - 1)) / C;

double y = A + par;

return y;

}

void get\_rand(double N, double A, double B, double C, double \*arr)

{

srand((unsigned)time(0));

for (int i = 0; i < N; i++)

{

arr[i] = get\_Bradford(A, B, C);

}

}

void insertionSort(double \*arr, int left, int right, int &c, int &s)

{

for (int i = left + 1; i <= right; i++)

{

double temp = arr[i];

int j = i - 1;

while (j >= left && arr[j] > temp)

{

c++;

arr[j + 1] = arr[j];

j--;

s++;

}

c++;

arr[j + 1] = temp;

s++;

}

}

void merge(double \*arr, int l, int m, int r,int &c, int &s)

{

int len1 = m - l + 1, len2 = r - m;

double\* left = new double[len1];

double\* right = new double[len2];

for (int i = 0; i < len1; i++)

left[i] = arr[l + i];

for (int i = 0; i < len2; i++)

right[i] = arr[m + 1 + i];

int i = 0, j = 0, k = l;

while (i < len1 && j < len2)

{

c++;

if (left[i] <= right[j]) {

arr[k] = left[i];

i++;

}

else {

arr[k] = right[j];

j++;

}

s++;

k++;

}

while (i < len1) {

arr[k] = left[i];

k++;

i++;

s++;

}

while (j < len2) {

arr[k] = right[j];

k++;

j++;

s++;

}

}

void TimSort(double\* arr, int n, int &c, int &s)

{

for (int i = 0; i < n; i += RUN)

insertionSort(arr, i, min((i + RUN - 1), (n - 1)),c,s);

for (int size = RUN; size < n; size = 2 \* size) {

for (int left = 0; left < n; left += 2 \* size) {

int mid = left + size - 1;

int right = min((left + 2 \* size - 1), (n - 1));

if (mid < right)

merge(arr, left, mid, right,c,s);

}

}

}

# ОПИСАНИЕ СТРУКТУРЫ ПРОГРАММНОГО КОМПЛЕКСА

На рисунке 2 приведена структура программного комплекса.

Диалоговые окна

Интерфейс пользователя

Меню

Ввод-вывод данных

Графическое представление

Бизнес-логика

Обработка данных

Оценивание коэффициентов уравнения связи

Расчет коэффициентов корреляции и детерминации

Расчет среднеквадратичной ошибки

Таблицы

Хранение данных

Графики

Рисунок 2 - Структура программного комплекса

# ОПИСАНИЕ СТРУКТУР ДАННЫХ

Таблица 1 – Описание структур данных

|  |  |  |
| --- | --- | --- |
| **MainForm Главное меню** | | |
| buttonExit\_Click() | Выход из программы. | |
| buttonInfo\_Click() | Открывает модальное окно с информацией о программе | |
| buttonSort\_Click() | Открывает модальное окно с сортировкой данных с помощью метода сортировки Timsort | |
| buttonEqConnection\_Click() | Открывает модальное окно с расчётом уравнения связи | |
| **InfoForm О программе** | | |
| buttonExit\_Click() | Возвращает в главное меню | |
| linkLabel1\_LinkClicked() | При нажатии открывает ссылку в браузере на репозиторий GitHub | |
| **SortForm Метод Timsort** | | |
| buttonExit\_Click() | | Возвращает в главное меню |
| buttonSort\_Click() | | Заполнение массива случайными числами и его сортировка |
| buttonClear\_Click() | | Очищает поля ввода и таблицы и выходные данные |
| textBoxN\_TextChanged() | | Делает кнопки buttonSort  и buttonClear активными при правильном вводе данных |

Продолжение таблицы 1.

|  |  |
| --- | --- |
| SortForm\_Load() | Отключает кнопки “Сортировка” и “Очистка” при запуске формы |
| **EqForm Уравнение связи. Расчетные данные и графики** | |
| buttonExit\_Click() | Возвращает в главное меню |
| buttonMHK\_Click() | Открывает модальное с теоретическим обоснованием |
| buttonClear\_Click() | Очищает все поля и таблицу и график |
| buttonCalc\_Click() | Провести анализ и рассчитать парный коэффициент корреляции Y и X, совокупный коэффициент детерминации , средняя квадратическая ошибка объема выборки X |
| sample\_size\_TextChanged() | Делает кнопки buttonSort  и buttonClear активными при правильном вводе данных |
| EqForm\_Load() | Отключает кнопку buttonSort при открытии модального окна |
| **MHKForm** | |
| buttonExit\_Click() | Возвращает к уравнению связи |