Частное учреждение образования

«Колледж бизнеса и права»

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРОГРАММНОГО СРЕДСТВА СОРТИРОВКИ И БИНАРНОГО ПОИСКА В МАССИВЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Основы алгоритмизации и программирование»

КП Т.091

Руководитель проекта (Ю.В.Шаляпин)

Учащийся (А.О.Зеневич)

2022

Частное учреждение образования

«Колледж бизнеса и права»

**Отделение** ПОИТ

**Специальность** 2-40-01 01 «Программное обеспечение информационных технологий»

**Группа** Т-091

**ЗАДАНИЕ**

на курсовое проектирование

Учащемуся (-ейся) Зеневичу Александру Олеговичу

1. Тема проекта: Программы для сортировки и бинарного поиска в массиве

2. Срок сдачи законченного проекта: **6 июня 2022 года**

3. Основные функциональные требования к проекту: разработать на языке программирования C++ в среде разработки MS Visual Studio программу для сортировки и бинарного поиска в массиве.

Реализовать: авторизацию и аутентификацию пользователей (роли: администратор, пользователь), локализацию на английском, русском и белорусском языке; возможность выбора цвета фона окна приложения из 3-х вариантов; ввод пользователя данных о размере массивов, ввод их содержимого и обеспечить возможность рандомного заполнения массивов. Реализовать сортировку Хоара и бинарный поиск в таблице символов, основанной на двумерном массиве. Визуализировать работу алгоритма бинарного поиске в виде деревьев для массивов размерами N=17 и N=24. В программе визуализировать массивы и содержимое файлов, обеспечить возможность чтения и записи файлов.

Обеспечить организацию лаконичного и понятного пользователя интерфейса программы средствами создания оконных форм с элементами меню для пользователя. Создать справочную систему приложения и установочных диск. Программа должна сохранять данные в файл (-ы) формата \*.docx, \*.txt и (-или) \*.bin, читать из них данные.

**Состав проекта:**

**А) Расчётно-пояснительная записка (перечень подлежащих разработке вопросов):**

Введение. 1. Постановка задачи. 2. Вычислительная система. 3. Проектирование задачи. 4. Описание программного средства. 5. Методика испытаний. 6. Применение. Заключение. Литература. Приложения (листинги программных модулей, копии экранных форм, отчёты и др.)

**Б) Графическая часть проекта:**

Лист 1. Блок-схема работы алгоритма сортировки Хоара

Лист 2. Блок-схема работы программы (общая)

**5. Календарный график работы на весь период проектирования:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Постановка задачи, определение вычислительной системы | | | 19.04 – 24.04 |
| 2 | Объектно-ориентированный анализ задачи, проектирование задачи | | | 25.04 – 01.05 |
| 3 | Описание программного средства | | | 02.05 – 22.05 |
| 4 | Методика испытаний (функциональное тестирование) | | | 23.05 – 27.05 |
| 5 | Применение | | | 28.05 – 31.05 |
| 6 | Оформление пояснительной записки | | | 01.06 – 05.06 |
| Председатель цикловой комиссии ПОИТ №10 | | | В.Ю.Михалевич | |
| Руководитель проекта | | | Ю.В.Шаляпин | |
| Дата выдачи задания «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 года | | | | |
| Задания принял к исполнению, | | | | |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 года | | Подпись учащегося: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Датаа*

*Лист*

*3*

*КП Т.091*

*Разраб.*

*Зеневич.А.О*

*Провер.*

*Шаляпин.Ю.В.*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Программная реализация программного «Помощник Прораба»*

*Лит.*

*Листов*

*КБиП*

*74*

*у*

# Введение

В ходе выполнении курсового проекта было задействовано множество форумов, сайтов, интернет-ресурсов, справочных материалов, методических указаний.

В первом разделе «Объектно-ориентированный анализ и проектирование системы»

описывается сущность поставленной задачи, предметная область, список задач, которые должны быть автоматизированы за счёт программного средства. Так же перечисляются основные функции программы, выполняется построение информационной модели, которая, в свою очередь, отражает сущности задач вместе с их свойствами.

Во втором разделе «Вычислительная система» описывается аппаратные и операционные ресурсы компьютера, на котором разрабатывалось программное средство.

В третьем разделе «Проектирование задачи» перечисляются основные требования к программному средству, логическую и физическую структуры данных в контексте среды разработки, структуру и описание функций пользователя в рамках среды разрабатываемого программного средства, функции и элементы управления, проектирование справочной системы программного средства.

В четвертом разделе «Описание программного средства» будут перечислены способы реализации функциональной части приложения и её функциональное назначение.

В пятом разделе «Методика испытаний» описывается перечень требований к техническим средствам для проведения тестирования, требования к информационной и программной совместимости. Представляются результаты функционального тестирования.

В шестом разделе «Применение» происходит описание сведений о назначении программного средства и области его применения, ограничениях. Так же описываются требования к техническим и программным ресурсам. Именно в этом разделе приводится структура справочной системы, а также методика её использования.

В заключении будут представлены выводы о созданном программном приложении, определена степень соответствия поставленной задачи и выполненной работы.

На практике бинарные деревья применяются, когда в каждой точке какого-нибудь вычислительного процесса нужно принимать одно из 2-х возможных решений. Существует множество задач, решаемых таким способом. Одна из них – выполнение операции, условно говоря, X с каждым элементом дерева. X рассматривается в качестве параметра обобщённой задачи посещения всех вершин либо задачи обхода дерева. Если рассмотреть такую задачу в качестве единого последовательного процесса, то можно сказать, что отдельные вершины посещаются в определённом порядке, то есть могу считаться линейно расположенными.

Целью данной курсовой работы является создание программного средства для сортировки и бинарного поиска в массиве. Для достижения поставленной цели необходимо выполнить следующие задачи:

* рассмотрение аналогов программных средств;
* изучение алгоритмов правильного распределения нагрузки;
* объектно-ориентированный анализ и проектирование системы;
* выявление требований к аппаратным и операционным ресурсам;
* определение необходимых инструментов для разработки.
* проектирование задачи;
* описание программного средства;
* написание программы;
* проведение испытаний и устранение выявленных ошибок.

При анализе электронных и литературных источников было выявлено два интернет-приложения, отвечающее требованиям к разрабатываемой программе: https://visualgo.net/en/bst и https://www.cs.usfca.edu/~galles/visualization/BST.html

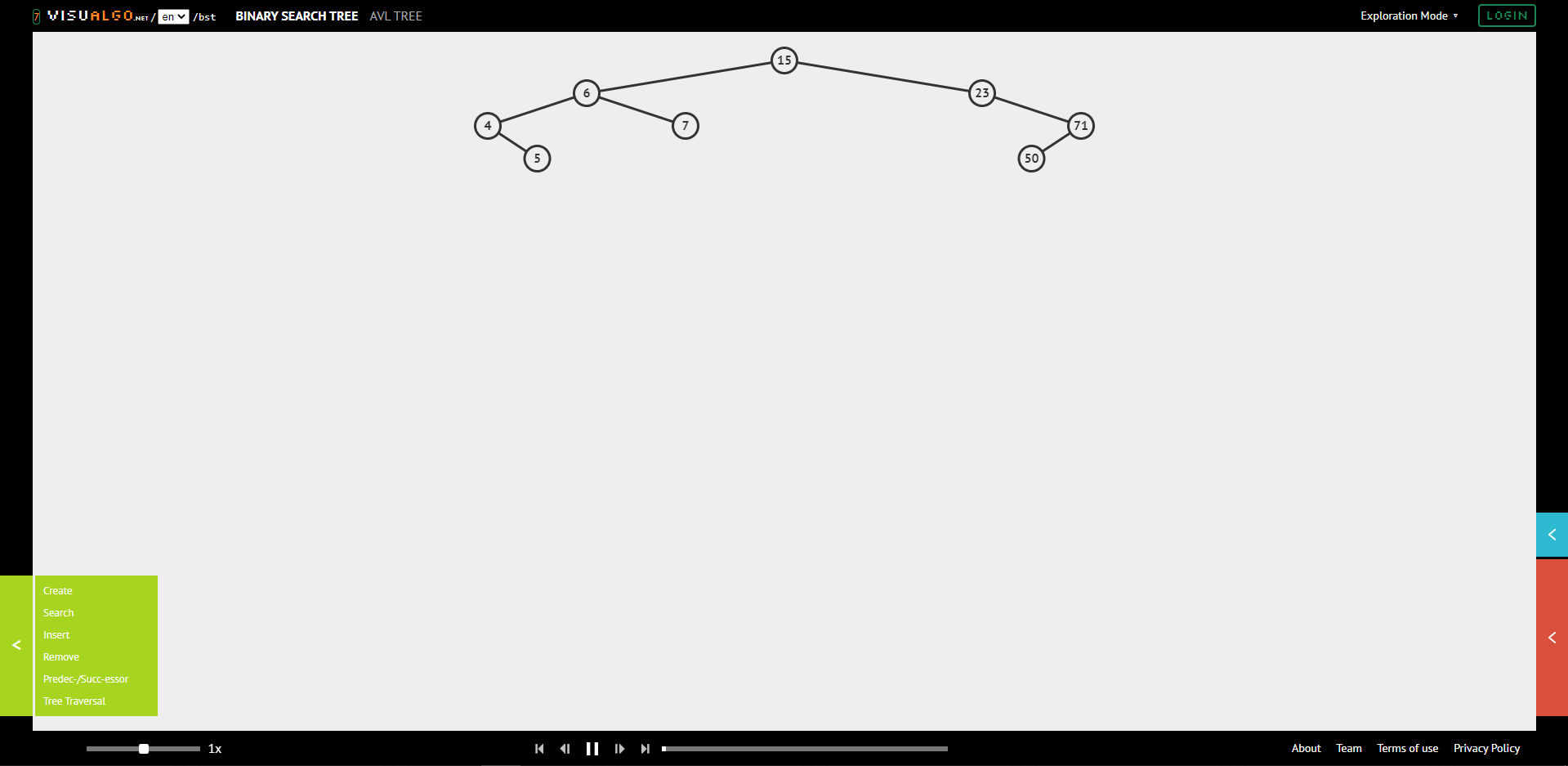


Рисунок 1 - Интернет-ресурс https://visualgo.net/en/bst

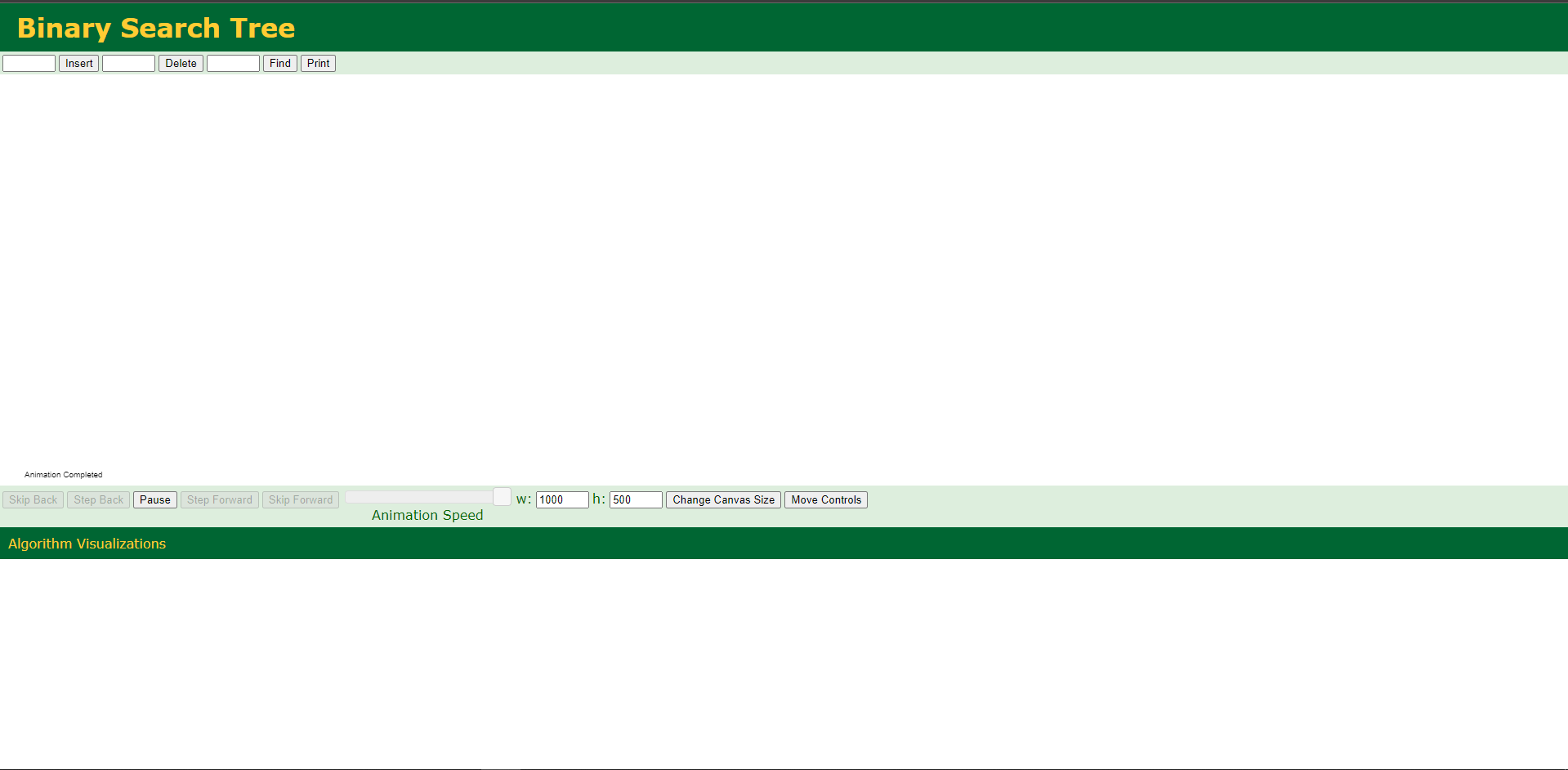


Рисунок 2 - Интернет-ресурс https://www.cs.usfca.edu/~galles/visualization/BST.html

# Объектно-ориентированный анализ и проектирование системы

## Сущность задачи

При любой сортировке происходит сравнение величин (полей сортировки). Массив чисел сортируется по принципу сравнения: если надо отсортировать по возрастанию (неубыванию), то сравниваем значения соседних элементов, и если левый элемент по хранимому в нем значению больше правого, то хранимые в них значения надо обменять местами; если массив надо отсортировать по убыванию (невозрастанию), то сравниваем значения соседних элементов, и если леворасположенный элемент хранит в себе значение меньшее, чем в праворасположенном элементе, то хранимые в них значения надо обменять местами. При более сложных алгоритмах сортировок сравниваться могут не соседние элементы массива, а имеющие между собой другие «промежуточные» элементы, но все равно эти удаленные друг от друга элементы сравниваются и учитывается, какой элемент расположен левее, а какой – правее. Возможно оперирование и сразу последовательностями элементов (группы расположенных рядом элементов, удовлетворяющих критерию сортировки, переставляются с другими группами элементов).

Бинарное дерево — это иерархическая структура данных, в которой каждый узел имеет значение (оно же является в данном случае и ключом) и ссылки на левого и правого потомка. Узел, находящийся на самом верхнем уровне (не являющийся чьим-либо потомком), называется корнем. Узлы, не имеющие потомков (оба потомка которых равны NULL) называются листьями.

Программное средство «Сортировки и бинарного поиска в массиве» позволит легче работать с массивом, без труда его сортировать, а также визуализировать его в виде бинарных деревьев и выполнять поиск.

Потенциальная аудитория – математики, студенты, программисты.

Функции необходимые для автоматизации:

* процесс ввода информации о массиве, необходимой для сортировки и работы бинарного дерева;
* экспорт массива в текстовый документ.

Исходя из всех вышеперечисленных факторов можно сделать вывод что написанное программное средство позволит начинающим программистам, математикам или же студентам показать и визуализировать работу с массивами и бинарными деревьями.

## Проектирование модели

Основное окно программы состоит из 3 вкладок: «Инициализация массива», «Сортировка Хоара» и «Бинарное дерево». В первой вкладке реализована возможность инициализации массива, используя кнопки «рандом размер» для рандомизации размера массива и «рандом массив» для его случайного заполнения и вывода в текстовое окно. Также массив можно создать индивидуально, для этого нужно указать размер массива по строкам и столбцам в текстовое окно, а после нажать кнопку «Создание массива» и начать его дальнейшее заполнение вручную через кнопку «Добавление числа в массив». Вторая вкладка отвечает за сортировку Хоара и его вывод в текстовое окно. При нажатии на кнопку «Сортировать» произойдёт сортировка массива и его вывод в текстовое окно. В третьей вкладке реализована возможность визуализации бинарного дерева при нажатии кнопки «Создать». Ниже находится окошко для бинарного поиска. При вводе ключа в текстовое окно, а после, нажатии кнопки «Поиск», будет реализован бинарный поиск. Кнопка «Очистить» очищает текстовые окна, уничтожает массивы, вызывает функцию очистки бинарного дерева и позволяет начать ввод заново.

Пользователю доступна возможность чтения и записи файлов посредством меню приложения. Также, в меню доступны функции смены темы оформления, локализации, вывода справки программы, а также выход из самой программы.

# Вычислительная система

## Требования к аппаратным операционным ресурсам

Довольно важную роль при разработке программных средств имеет компьютер, на котором ведется разработка. Написание программного средства «Сортировки и бинарного поиска в массиве» будет вестись на персональном компьютере, со следующими характеристиками:

* процессор Intel Core i7-4700HQ, 2013 года выпуска, техпроцесс 22 нанометра, частота 3.4 GHz, 4 ядра, 8 потоков;
* оперативная память 8ГБ, DDR3, 1600 МГц;
* видеокарта Nvidia GeForce GTX 850M 4ГБ;
* операционная система MS Windows 10 Pro, 64-разрядная.

## Инструменты разработки

Для быстрой и эффективной разработки, отладки и развертки приложения были использованы нижеследующие программные средства:

* операционная система MS Windows 10 Pro;
* среда программирования MS Visual Studio 2022 – использовалась для написания кода программы, построения форм приложения, отладки и развертки;
* язык программирования «C++» – был выбран из-за быстродействия, гибкости и

возможности работать с крупными массивами данных;

* paint – программа использовалась для создания иконки приложения в формате «png»;
* веб-приложение www.xiconeditor.com – использовалось для конвертации «png» файла иконки в формат «ico»;
* smart install maker 5.04 – использовалась для создания инсталлятора и деинсталлятора

для готовой программы.

# Проектирование задачи

## Требования к приложению

Основные функциональные требования к проекту

* реализовать авторизацию и аутентификацию пользователей, локализацию на трёх языках (английский, русский и белорусский), а также оформление (возможность выбора цвета фона окна приложения из 3-х вариантов);
* реализовать ввод пользователем данных о размере массивов, ввод их содержимого и обеспечить возможность рандомного заполнения массивов;
* реализовать сортировку Хоара и бинарный поиск;
* реализовать визуализацию работу алгоритма бинарного поиска в виде деревьев
* реализовать справочную систему программы.

Требования к интерфейсу максимально просты, интерфейс должен быть «дружелюбен» и понятен пользователю. Так же интерфейс должен обеспечивать организацию интуитивно-понятного пользования им.

Защиты информации от несанкционированного использования не предусмотрено ТЗКП. Средством защиты может выступать пользователь, который должен предусмотреть список людей, которые будут иметь доступ к данному программному средству.

## Концептуальный прототип

Разработанное программное средство представляет собой файл «.exe».

При двойном нажатии правой кнопкой мыши на файл, запускается окно авторизации и аутентификации пользователей состоящая из главного меню, текстовых окон для заполнения данных и соответствующими кнопками управления, изображённых на Рисунок 3.

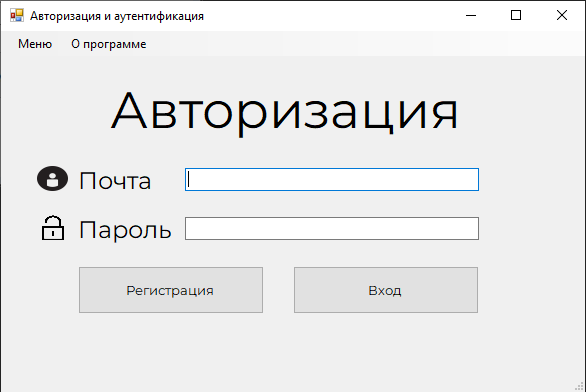


Рисунок 3 - Окно авторизации и аутентификации

В окне регистрации представлены регистрация пользователя, состоящая из главного меню, текстовых окон для заполнения данных и соответствующими кнопками управления, изображённых на Рисунок 4.

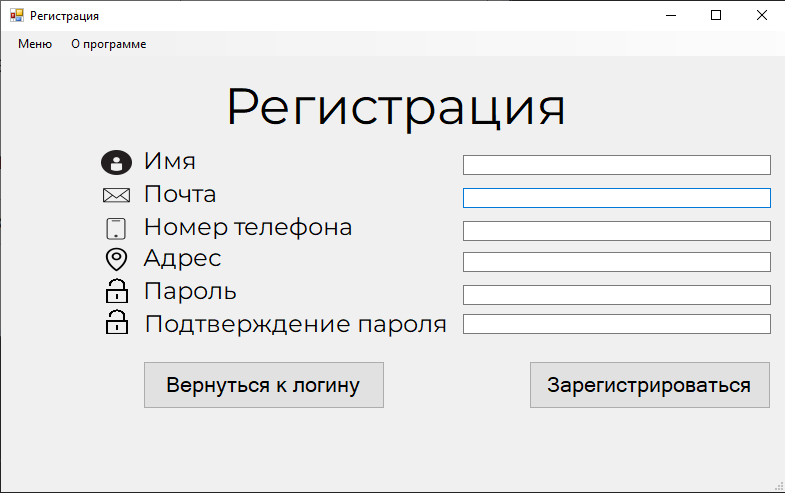


Рисунок 4 - Окно регистрации

Основное окно программы состоит из 3 вкладок: «Инициализация массива», «Сортировка Хоара» и «Бинарное дерево». В первой вкладке реализована возможность инициализации массива, используя кнопки «рандом размер» для рандомизации размера массива и «рандом массив» для его случайного заполнения и вывода в текстовое окно. Также массив можно создать индивидуально, для этого нужно указать размер массива по строкам и столбцам в текстовое окно, а после нажать кнопку «Создание массива» и начать его дальнейшее заполнение вручную через кнопку «Добавление числа в массив». Представлено на Рисунок 5.

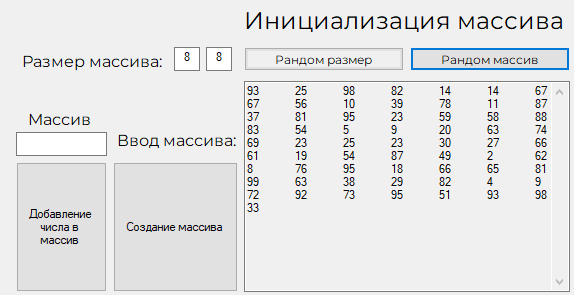


Рисунок 5 - Вкладка инициализации массива

Вторая вкладка отвечает за сортировку Хоара и его вывод в текстовое окно. При нажатии на кнопку «Сортировать» произойдёт сортировка массива и его вывод в текстовое окно. Представлено на Рисунок 6.

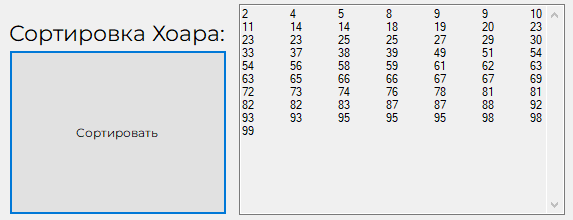


Рисунок 6 - Вкладка Сортировки Хоара

В третьей вкладке реализована возможность визуализации бинарного дерева при нажатии кнопки «Создать». Ниже находится окошко для бинарного поиска. При вводе ключа в текстовое окно, а после, нажатии кнопки «Поиск», будет реализован бинарный поиск. Кнопка «Очистить» очищает текстовые окна, уничтожает массивы, вызывает функцию очистки бинарного дерева и позволяет начать ввод заново. Представлено на Рисунок 7.



Рисунок 7 - Вкладка бинарного дерева и поиска

Пользователю также доступна возможность чтения и записи файлов посредством меню приложения. Представлено на Рисунок 8 и Рисунок 9.

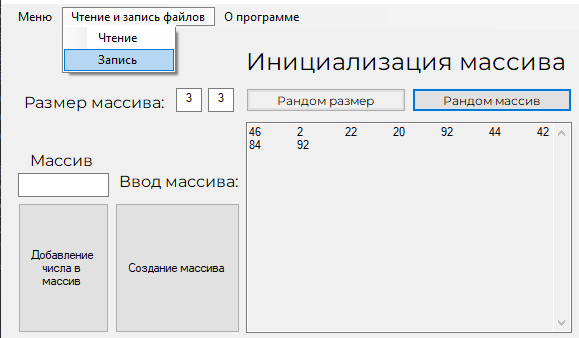


Рисунок 8 - Запись массива в файл

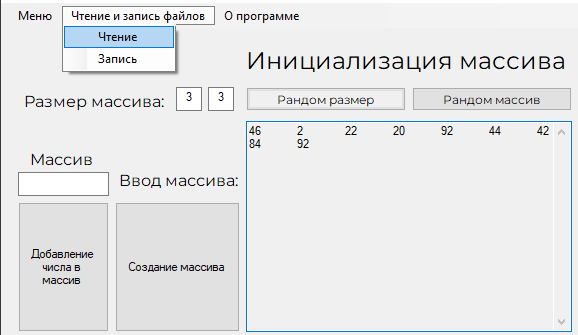


Рисунок 9 - Чтение массива из файла

Также, в меню доступны функции смены темы оформления, локализации, а также выход из самой программы. Представлено на Рисунок 10 и Рисунок 11.

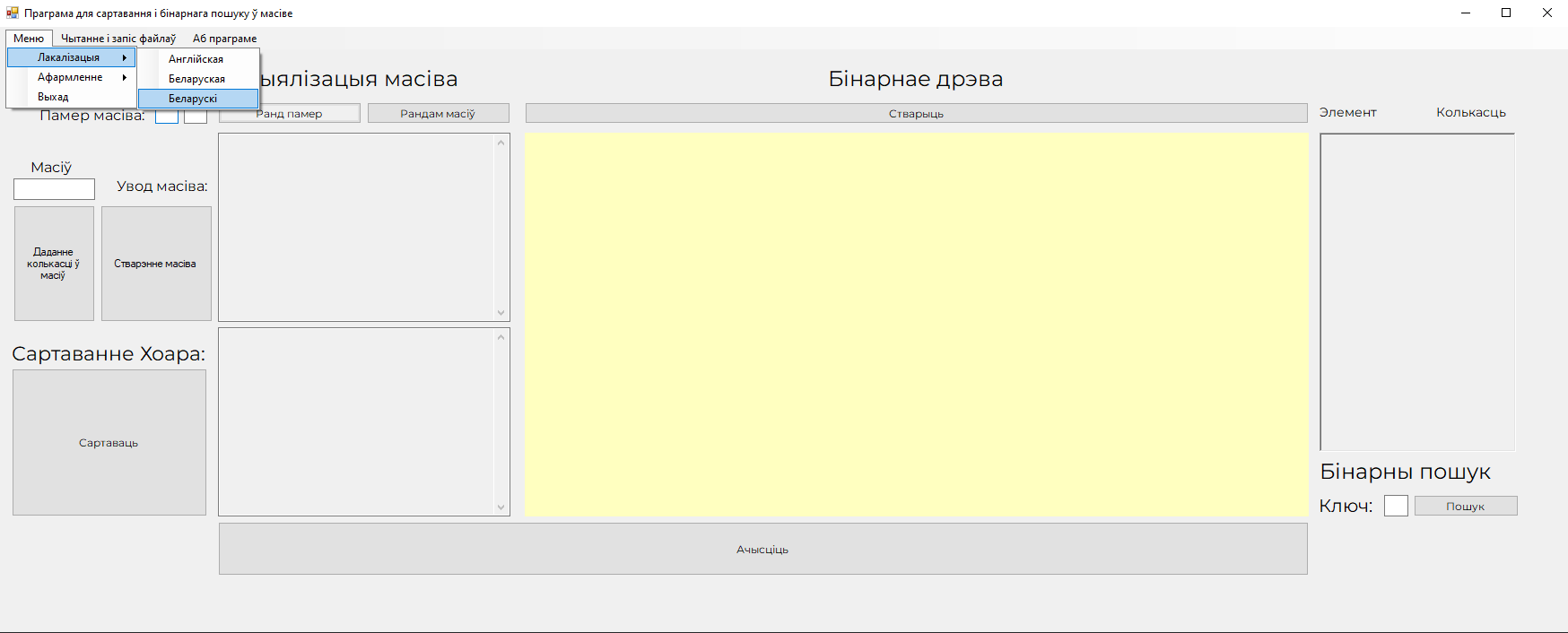


Рисунок 10 - Локализация

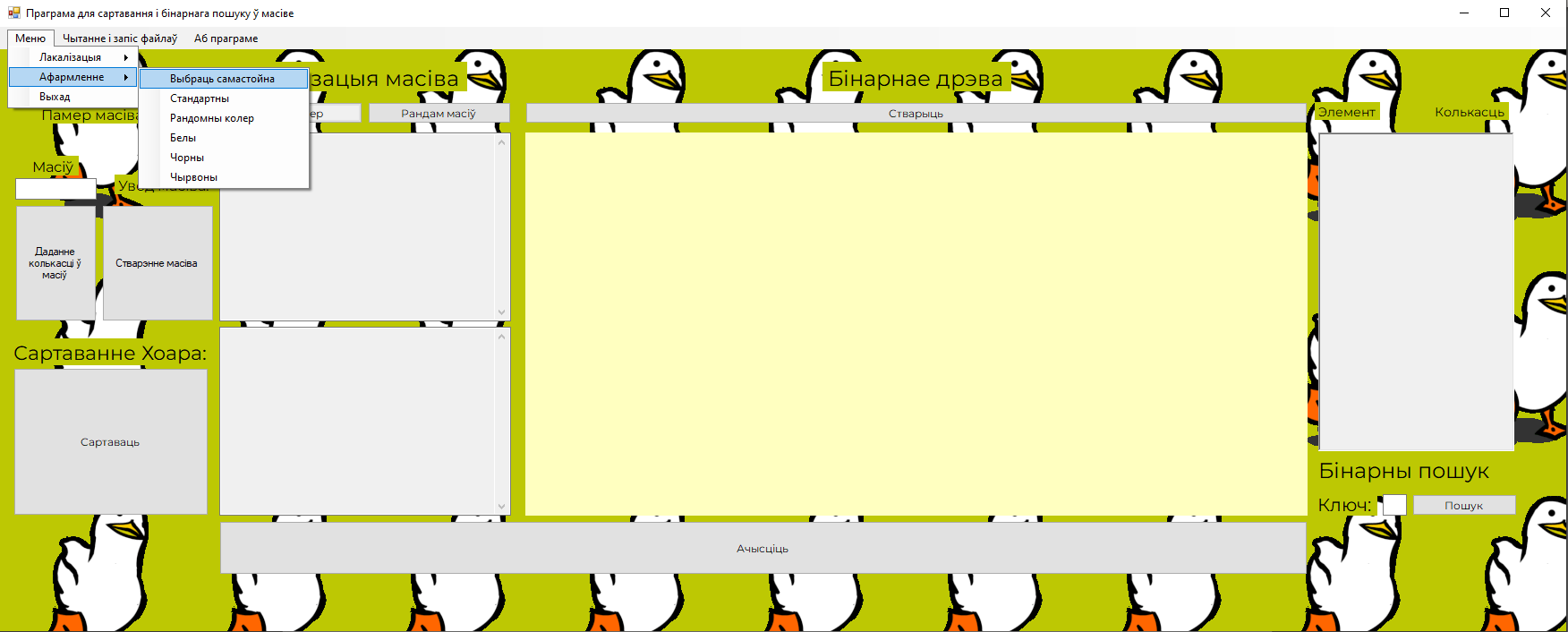


Рисунок 11 - Оформление

В последней кнопки меню реализован вывод краткой справки о программе. Представлено на Рисунок 12.

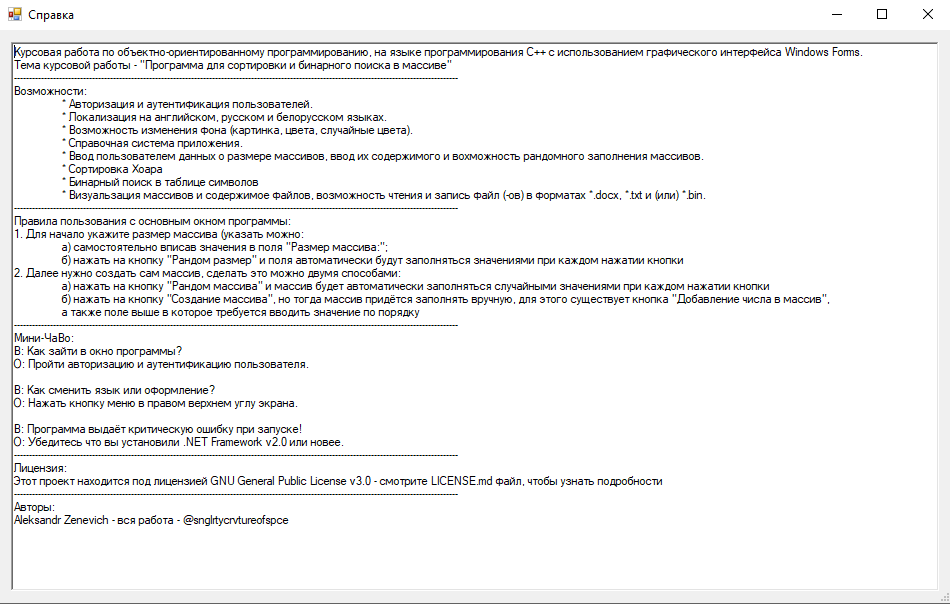


Рисунок 12 – Справка

## Организация данных

| Размещение | Элемент экрана | Класс | Действие |
| --- | --- | --- | --- |
| Форма программы | Рандом размер | Button | Рандомизация размера массива и его отображение в текстовые окна |
| Рандом массив | Button | Рандомизация элементов массива и его отображение в текстовое окно |
| Создание массива | Button | Создание пустого массива и его отображение в текстовое окно |
| Добавление числа в массив | Button | Добавление числа в массив начиная с нулевого элемента и его постоянное обновление в текстовом окне |
| Сортировать | Button | Реализует сортировку массива алгоритмом быстрой сортировки Хоара |
| Создать | Button | Визуализация бинарного дерева в панель |
| Очистить | Button | Очищает текстовые окна, уничтожает массивы, вызывает функцию очистки бинарного дерева и позволяет начать ввод заново. |
| Поиск | Button | Реализует бинарный поиск, ключ берётся с текстового окна. |
| Меню – Локализация | ToolStripMenuItem | Выбор языка приложения |
| Меню – Оформление | ToolStripMenuItem | Выбор темы приложения |
| Меню – Выход | ToolStripMenuItem | Закрытие приложения |
| Чтение | ToolStripMenuItem | Чтение данных из файла и вывод в текстовое окно |
| Запись | ToolStripMenuItem | Запись данных из текстового окна в файл |
| О программе | ToolStripMenuItem | Открытие краткой справки |

Таблица 1 - Элементы управления и визуализации

## Функции: логическая и физическая организация и элементы управления

| Функция | Компонент меню | Название элемента  интерфейса | Реализация |
| --- | --- | --- | --- |
| Смена языка | английскийToolStripMenuItem1\_Click | Щелчок мыши по кнопке «Меню – Локализация – Английский» | Средства С++ |
| Смена языка | русскийToolStripMenuItem2\_Click | Щелчок мыши по кнопке «Меню – Локализация – Русский» | Средства С++ |
| Смена языка | русскийToolStripMenuItem3\_Click | Щелчок мыши по кнопке «Меню – Локализация – Белорусский» | Средства С++ |
| Смена оформление заднего фона картинкой форматов (\*.BMP; \*.JPG; \*.GIF; \*.PNG) | выбратьСамостоятельноToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Выбрать самостоятельно» | Средства С++ |
| Смена цвета заднего фона на стандартный | стандартныйToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Стандартный» | Средства С++ |
| Рандомизация цвета заднего фона | рандомныйЦветToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Рандомный цвет» | Средства С++ |
| Смена цвета заднего фона на белый | белыйToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Белый» | Средства С++ |
| Смена цвета заднего фона на чёрный | чёрныйToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Чёрный» | Средства С++ |
| Смена цвета заднего фона на красный | красныйToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Оформление – Красный» | Средства С++ |
| Закрыть программу | выходToolStripMenuItem\_Click | Щелчок мыши по кнопке «Меню – Выход» | Средства С++ |
| Вывод окна справки программы | оПрограммеToolStripMenuItem\_Click | Щелчок мыши по кнопке «О программе» | Средства С++ |
| Рандомизация элементов массива и его отображение в текстовое окно | button1\_Click | Кнопка «Рандом массив» | Средства С++ |
| Рандомизация размера массива и его отображение в текстовые окна | button2\_Click | Кнопка «Рандом размер» | Средства С++ |
| Очищает текстовые окна, уничтожает массивы, вызывает функцию очистки бинарного дерева и позволяет начать ввод заново. | button3\_Click | Кнопка «Очистить» | Средства С++ |
| Реализует сортировку массива алгоритмом быстрой сортировки Хоара | button4\_Click | Кнопка «Сортировать» | Средства С++ |
| Создание пустого массива и его отображение в текстовое окно | button5\_Click | Кнопка «Создание массива» | Средства С++ |
| Добавление числа в массив начиная с нулевого элемента и его постоянное обновление в текстовом окне | button6\_Click | Кнопка «Добавление числа в массив» | Средства С++ |
| Реализует бинарный поиск, ключ берётся с текстового окна. | button7\_Click | Кнопка «Поиск» | Средства С++ |
| Визуализация бинарного дерева в панель | button8\_Click | Кнопка «Создать» | Средства С++ |
| Чтение данных из файла и вывод в текстовое окно | чтениеToolStripMenuItem\_Click | Щелчок мыши по кнопке «Чтение и запись файлов – Чтение» | Средства С++ |
| Запись данных из текстового окна в файл | записьToolStripMenuItem\_Click | Щелчок мыши по кнопке «Чтение и запись файлов – Запись» | Средства С++ |

# Описание программного средства

## Общие сведения

Детальные сведения о программе и проблеме, которую она решает, приводятся в подразделе 1.1.

Также, в ходе выполнения курсового проекта были задействованы следующие подключаемые модули приложения:

1. «vector»;
2. «string».

«Windows Forms» – интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Имеется возможность разработки на C#, C++, VB.Net, J# и др.

«vector» – это структура данных, которая уже является моделью динамического массива, стандартный шаблон обобщённого программирования и реализующий его.

«string» – класс с методами и переменными для организации работы со строками в языке программирования С++. Он включён в стандартную библиотеку С++. Название образовано от имени строчного типа данных (англ. string; с англ. - «строка»).

## Функциональное назначение

Программа разработана для работы с массивом, без труда его сортировать, а также визуализировать его в виде бинарных деревьев и выполнять поиск. Все ошибки, которые могут возникнуть во время работы программы обработаны с использованием блоков «try-catch».

## Входные и выходные данные

В качестве входных данных используются конвертируемые целочисленные значения из текстовых окон для размера массива, а также если пользователь выберет инициализацию массива самостоятельно. Для бинарного дерева в качестве входных данных используются структуры «Node» класса «BinaryTree».

В качестве выходных данных используются текстовые окна, весь вывод массива, а после и отсортированный также в нём. Также, в свою очередь выходными данными являются данные, записанные в текстовый файл.

# Методика испытаний

## Технические требования

Для полноценного функционирования разработанного программного приложения, качественного выполнения всех поставленных задач, требуется ПК следующей конфигурации:

* процессор с частотой не менее 1 ГГц;
* ОЗУ от 1 ГБ (для 32-разрядной ОС) или 2 ГБ (для 64-разрядной ОС);
* от 16 (для 32-разрядной ОС) или от 20 ГБ (для 64-разрядной ОС) свободного места на жестком диске;
* наличие DirectX 9 и более поздняя версия с драйвером WDDM 1.0.
* .NET Framework 4.0 или новее.
* операционная система: MS Windows 7 или новее;

Для проведения испытаний функций сохранения и загрузки использовались файлы с заранее сохраненными в них данными. Тестирование проводилось на аппаратных базах ноутбуков Asus N750JK и HP Pavilion 15, а также на стационарных ПК с различными техническими конфигурациями. Во время тестирования программы проблем, связанных с аппаратной совместимостью, выявлено не было.

## Функциональное тестирование

2