

Введение

На курсовое проектирование была поставлена задача, разработать программу на тему: Электронный учебный тренажёр «Динамические массивы».

Цель курсового проекта заключается в знакомстве пользователя с принципами работы и методами обработки динамических массивов.

Создаваемая программа будет рассчитана пользователей изучающих основы программирования. Применить данную программу смогут не только люди, изучающие основы программирования, но и просто люди, заинтересовавшиеся в данной теме.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

«Реализация задачи» – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		4

пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе «Применение» будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В «Литературе» будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		5

1 Анализ задачи

1.1 Постановка задачи

Темой данного курсового проекта является «Динамические массивы».

Динамическим называется массив, размер которого, при необходимости, может меняться во время исполнения программы.

Динамические массивы дают возможность более гибкой работы с данным, так как позволяют не прогнозировать хранимые объёмы данных, а регулировать размер массива в соответствии с реально необходимыми объёмами.

В отличие от динамических массивов существуют статические массивы и массивы переменной длины.

Размер статического массива определяется на момент компиляции программы.

Размер массива переменной длины определяется во время выполнения программы. Отличием динамического массива от массива переменной длины является автоматическое изменение размеров, что не трудно реализуется в случаях его отсутствия, поэтому часто не различают массивы переменной длины с динамическими массивами.

Динамические массивы являются основополагающим элементом в программировании и применяются в различных областях для хранения и управления данными.

Динамические массивы могут увеличиваться или уменьшаться в размере во время выполнения программы. Это достигается за счет перераспределения памяти и копирования существующих элементов в новую область памяти.

Элементы в динамическом массиве могут быть быстро доступны по их индексу, что делает их очень эффективными для операций, требующих случайного доступа к данным.

Динамические массивы могут хранить данные различных типов, включая примитивные типы (целые числа, числа с плавающей запятой и т.д.) и сложные типы (объекты, структуры). Это делает их универсальными и подходящими для множества задач.

Когда массив достигает своего текущего максимального размера и необходимо добавить новый элемент, происходит перераспределение памяти. Обычно создается новый массив большего размера, элементы копируются из старого массива в новый, и старый массив освобождается. Этот процесс может быть дорогим с точки зрения производительности, но он обычно происходит редко благодаря использованию стратегий увеличения размера (например, удвоение текущего размера).

Над динамическими массивами можно выполнять различные операции:

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		6

- Создание массива: инициализация нового динамического массива с заданным начальным размером;
- Добавление элементов: добавление нового элемента в конец массива или на указанную позицию;
- Удаление элементов: удаление элемента из конца массива или с указанной позиции;
- Доступ к элементам: чтение и изменение элементов массива по индексу;
- Поиск элементов: поиск элемента по значению и возврат его индекса;
- Сортировка: упорядочивание элементов массива с использованием различных алгоритмов сортировки;
- Итерация: проход по всем элементам массива с использованием циклов или итераторов;
- Изменение размера: вручную или автоматически изменение размера массива при необходимости.

Изучение принципа работы динамических массивов может понадобиться всем начинающим программистам. Программный продукт "Динамические массивы" предназначен для обучения студентов и начинающих программистов основам работы с динамическими массивами. Основная цель продукта — предоставить интерактивную, визуально-наглядную и практическую платформу для изучения и закрепления знаний о динамических массивах, их свойствах, методах и применениях.

Прямых аналогов у данного программного продукта немного. Существуют различные сайты которые предлагают курсы по программированию и структурам данных, в том числе и динамическим массивам. Однако они могут не всегда предоставлять интерактивные визуализации по динамическим массивам. Некоторые учебники и онлайн-руководства предлагают интерактивные примеры кода и задачи, но могут быть ограничены в функциональности и не поддерживать визуализацию изменений в реальном времени. Приложения и программы предлагают интерактивные среды для изучения структур данных, включая динамические массивы, но могут не предоставлять полный набор функций для комплексного обучения и практики.

Программный продукт будет использоваться регулярно студентами и преподавателями, работающими над курсами по программированию и структурам данных. Он может быть использован как в рамках учебных занятий, так и для самостоятельного изучения и практики.

Задачи которые будут автоматизированы в этом программном продукте:

- Возможность создания массивов различных типов данных, таких как целые числа, числа с плавающей запятой и символы;

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		7

- Автоматизация процесса добавления новых элементов с динамическим изменением размера массива;
- Упрощение процесса удаления элементов и соответствующая корректировка размера массива;
- Чтение и изменение элементов массива по индексу;
- Реализация эффективных алгоритмов поиска элементов в массиве по значению или индексу;
- Предоставление различных методов сортировки массивов;
- Автоматическое увеличение или уменьшение размера массива при добавлении или удалении элементов;
- Средства для визуализации изменений в массиве в реальном времени, чтобы учащиеся могли видеть, как изменяются данные и структура массива.

Новая задача будет упрощать решение поставленной задачи тем что предоставит интерактивный интерфейс, который позволяет учащимся непосредственно взаимодействовать с динамическими массивами, наблюдать за изменениями в реальном времени и визуализировать структуры массива и процессов изменения его размера, что облегчит понимание концепций. Цель данного курсового проекта – разработать программный продукт, который будет сочетать в себе элементы обучения структуры динамических массивов.

К входной информации можно отнести вводимые пользователем значения, например размер динамического массива который он указал, элемент который необходимо найти, удалить или добавить.

К выходной информации можно отнести динамический массив который будет создан по данным пользователя, информация полученная в результате работы с массивом.

Постоянной информацией в проекте будут текстовая информация, справочная информация, описание и название функций и методов.

1.2 Инструменты разработки

Для разработки проекта я использовал среду Delphi 11, так как это очень удобная и доступная среда разработки на данный момент. На Delphi 11 возможно разрабатывать проекты различной сложности самому либо в команде.

Delphi – это интегрированная среда разработки (IDE) для создания программного обеспечения на языке программирования Object Pascal.

Delphi известен своей высокой производительностью и быстротой разработки. Визуальный редактор форм позволяет быстро создавать пользовательский интерфейс, а богатая библиотека компонентов предоставляет широкий спектр готовых решений для реализации различных функциональных возможностей.

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		8

Delphi 11 обеспечивает кроссплатформенную разработку для различных операционных систем, таких как Windows, macOS, iOS, Android и Linux. Это позволяет создавать приложения, работающие на различных устройствах, что может быть особенно важно для проектов, ориентированных на широкую аудиторию. Object Pascal, используемый в Delphi, сочетает в себе простоту и выразительность. Он обладает синтаксисом, который легко читается и поддерживается, что упрощает процесс разработки и обеспечивает понятность кода даже для новых членов команды. Delphi поставляется с обширным набором инструментов и библиотек, включая компоненты для работы с базами данных, сетевым взаимодействием, графикой и многим другим. С помощью визуального редактора форм можно создавать пользовательский интерфейс приложений путем перетаскивания и настройки компонентов.

Delphi позволяет создавать различные виды программ такие как: оконные приложения, консольные приложения и приложения для работы с базами данных. Delphi предоставляет мощные средства для работы с базами данных. Среда позволяет использовать компоненты для доступа к различным СУБД, создавать сложные запросы и управлять данными в приложениях.

В Delphi сочетаются различные основные и дополнительные инструменты для упрощения разработки приложений. С помощью конструктора форм не нужно самому просчитывать где разместить элемент программы, это можно сделать с помощью визуального интерфейса. За счёт этого можно сэкономить много времени.

Adobe photoshop 2023 – это мощный графический редактор, который широко используется для редактирования и обработки изображений. Включает в себя широкий спектр инструментов и функций, позволяющих профессионалам и любителям создавать и редактировать изображения любой сложности. Необходим для оформления интерфейса и для работы с текстом, включая различные шрифты, стили и эффекты. Photoshop предоставляет множество инструментов для редактирования цветов, яркости, контрастности и других характеристик изображений.

Microsoft Word 2021 – это универсальное приложение для обработки текстовых документов. Оно предоставляет широкий спектр возможностей для создания, редактирования и форматирования текста. С помощью Microsoft Word легко создавать разнообразные документы, такие как письма, резюме, отчеты, статьи, презентации и многое другое. Программа обладает удобным и интуитивно понятным интерфейсом, что делает её доступной для пользователей всех уровней навыков. Благодаря возможностям автоматизации, шаблонам и интеграции с другими приложениями Microsoft Office, такими как Excel и PowerPoint, Word является незаменимым инструментом для работы с текстовой информацией в офисной среде, образовании и повседневной жизни. Microsoft Word необходим для написания документации.

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		9

Microsoft Powerpoint 2021 – это программное обеспечение для создания и демонстрации презентаций, разработанное компанией Microsoft. PowerPoint позволяет пользователям создавать профессионально выглядящие слайды с текстом, изображениями, графикой, аудио- и видеофайлами. С помощью широкого набора инструментов форматирования и макетов легко оформить слайды в соответствии с их потребностями и предпочтениями. Программа также предоставляет функции анимации и переходов, что позволяет создавать динамичные и запоминающиеся презентации. PowerPoint интегрируется с другими приложениями Microsoft Office, что обеспечивает совместимость и возможность обмена данными между различными типами документов.

Smart Install Maker – это программа для создания установочных пакетов для программного обеспечения под операционные системы Windows. Она обладает простым и интуитивно понятным интерфейсом, который позволяет пользователям легко создавать профессиональные установщики для своих программ.

С помощью Smart Install Maker можно создавать нестандартные установочные пакеты, включая различные опции установки, параметры, создание ярлыков, добавление лицензионных соглашений и многое другое. Программа поддерживает автоматическое создание систем восстановления и резервного копирования, что обеспечивает дополнительную гибкость и надежность процесса установки.

Smart Install Maker также предоставляет возможность создания многоязычных установочных пакетов, что позволяет программам быть доступными для широкой аудитории. Кроме того, программа поддерживает сжатие файлов и защиту паролем, обеспечивая безопасность и компактность установочных пакетов.

Dr.Explain – это инструмент для создания документации и руководств пользователя. Он автоматизирует процесс написания документации для ПО и веб-сайтов.

Разработка ведется на ноутбуке Asus, который имеет следующие системные характеристики:

- Процессор Amd Ryzen 5 4600H;
- Видеокарта Nvidia Geforce RTX 3050;
- SSD-накопитель на 500 гб;
- Объем ОЗУ 16 гб;
- Операционная система Windows 11 Pro.

Однако программа не очень требовательна к системным ресурсам и способна работать на любых современных компьютерах.

1.3 Требования к приложению

На этапе исследования предметной области был установлен целый ряд

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		10

требований, которые предъявляются к разрабатываемой программе:

- Ограничения связанные с возможностями среды программирования Delphi при разработке программного продукта;
- Необходима возможность вызова справки;
- Возможность установки программы из установочного пакета;

Требования к интерфейсу:

Программа должна иметь приятную стилистику и цветовую гамму для глаз. Интерфейс должен быть понятным и легким в использовании даже для пользователей без предварительного опыта работы с программами по программированию. Не должно быть большого количества кнопок на экране и они не должны быть слишком маленькие. Должны присутствовать различные элементы взаимодействия с программой такие как поля ввода и элементы управления. Пользователь не должен чувствовать дискомфорт от расположения элементов с любого экрана на котором запущена программа.

Программа должна иметь приятное окно заставки.

Необходимо предоставить подробные справочные материалы, описание функций и подсказок, помогающих пользователям понять принципы работы с динамическими массивами и функциональные возможности программы.

Все объекты должны помещаться на окно программы полностью. Не следует перегружать окна большим количеством текстовой информации. Должны соблюдаться правильные пропорции объектов на экране. Все элементы должны иметь общую стилистику.

Требования к надежности:

Программный продукт должен быть стабильным и надежным в работе, минимизируя возможные сбои, ошибки и аварийные ситуации.

Необходимо предусмотреть осуществление проверки вводимых пользователем данных на корректность и при необходимости предоставление соответствующих сообщений об ошибках.

Необходимо реализовать механизмы обеспечения отказоустойчивости программы, позволяющие восстанавливаться после сбоев и восстановить работоспособность.

Время восстановления после отказа вызванное программным сбоем электропитания не должны превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на работоспособность программы. Необходимо фиксировать все этапы разработки программного продукта. Это необходимо для восстановления процесса разработки на любом этапе.

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		11

Программа после сбоя должна запускаться повторно. Должна быть доступна в любой момент времени.

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		12

2 Проектирование задачи

2.1 Организация данных

Проектирование задачи – это очень важный и ответственный этап в разработке любого приложения.

Он важен потому, что методы управления формами пользователем основываются на высокой степени специализации каждого из компонентов.

Необходимым условием при разработке данного приложения является описание организации данных, т.е. логическая и физическая структура данных в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первым видом являются данные, которые будут введены разработчиком на этапе реализации задачи. Сюда можно отнести изображения (иконки), описания методов работы с динамическими массивами, исходные коды методов работы с динамическими массивами.

Вторым видом данных, используемых в программе, является вводимая пользователем информация. Входной информацией в разрабатываемой программе будут являться данные, которые введет пользователь. Это может быть количество данных (количество элементов), сами данные (элементы), которые необходимо добавить, удалить, изменить или найти. Тип данных может выбрать сам пользователь. Это может быть integer (пользователь вводит только целые числа), real (пользователь вводит вещественные числа) или char (пользователь вводит символы). Но необязательно заполнять элементы вручную, в программе предусмотрена генерация элементов самой программой. Подходящей структурой для хранения данных являются динамический одномерный и двумерный массивы.

Третьим видом данных является результат программы – это могут быть данные полученные в результате проведения различных операций над массивами, отсортированные пользовательские данные. Его также относят к отдельному виду, так как ни пользователь, ни разработчик его не вводят, а программа сама получает его в результате выполнения определенных действий.

Таким образом, организация данных является важной задачей при разработке данной и любой программы.

2.2 Процессы

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая курсовая программа. Главной ее задачей будет являться: демонстрация работы динамических массивов, алгоритмов их обработки и сортировки. Весь данный процесс будет заключаться в следующем.

Пользовательские данные хранятся в матрице имеющей разное количество строк и столбцов. Над матрицей будут выполняться различные действия.

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		13

В разделе «Создание массива» пользователь создаёт массив, который может состоять из разных типов данных. Пользователь сам может вводить значения в массив либо выбрать заполнение программой автоматически. Для визуализации массива будет использоваться компонент StringGrid. С помощью процедур будет выбираться количество строк и столбцов этого компонента. Так же с помощью процедуры будет выполняться заполнение компонента элементами.

В разделе «Сортировка массива» пользователь выбирает каким способом будет отсортирован массив. С помощью процедур будет осуществляться алгоритм сортировки массива.

В разделе «Поиск в массиве» пользователь выбирает какой элемент он хочет найти в массиве. С помощью процедур будет выполняться поиск элементов по параметрам выбранным пользователем.

В разделе «Чтение и изменение элементов массива» пользователь выбирает какие операции хочет провести над элементами в массиве. С помощью различных процедур будет возможность удалять, изменять и добавлять элементы в массив.

2.3 Описание внешнего пользовательского интерфейса

Важным аспектом выполнения курсового проекта является налаживание диалога между пользователем и программой. Это во многом зависит от того, как программист спроектирует программу, какие компоненты будут включены и какие процессы будут автоматизированы. Во-первых, особое внимание следует уделить интерфейсу. Разработчик должен оформить программу таким образом, чтобы пользователь сразу понял, что от него требуется.

Для обеспечения эффективной работы пользователя нужно создать целостное приложение, охватывающее конкретную предметную область, где все компоненты будут сгруппированы по их функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс. Приложение должно позволять пользователю решать задачи с минимальными усилиями по сравнению с работой с разрозненными объектами.

Ниже представлена система меню и организация навигации между окнами программы (Рисунок 1):

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		14



Рисунок 1 – Навигация между окнами программы

Прототип – это наглядная модель пользовательского интерфейса. В сущности это «черновик», созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа. Ниже можно наблюдать прототипы:

- 1) Логотипа (Рисунок 2);
- 2) Заставки (Рисунок 3);
- 3) Главного окна (Рисунок 4);
- 4) Окна, содержащего основные разделы программы (Рисунок 5);
- 5) Окна, демонстрирующего визуализацию работы с динамическими массивами (Рисунок 6);
- 6) Окна, содержащего теорию по выбранному разделу визуализации (Рисунок 7);
- 7) Окна, содержащего код программы (Рисунок 8).



Рисунок 2 – Прототип логотипа



Рисунок 3 – Прототип заставки

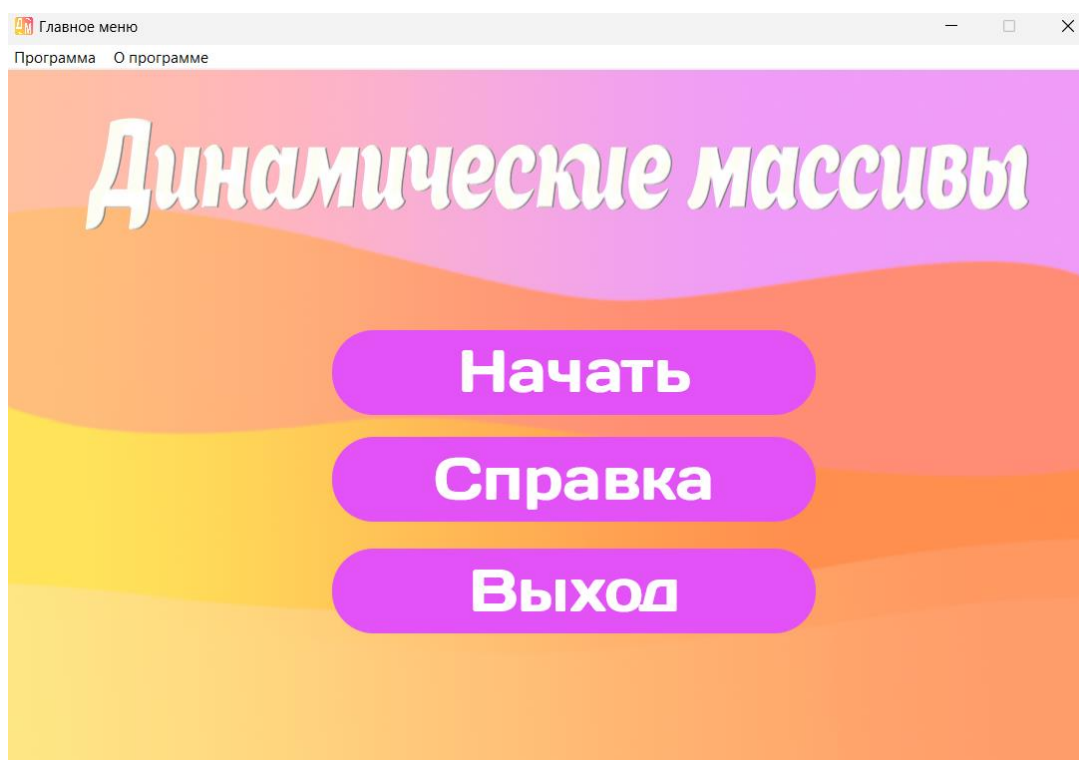


Рисунок 4 – Прототип главного окна

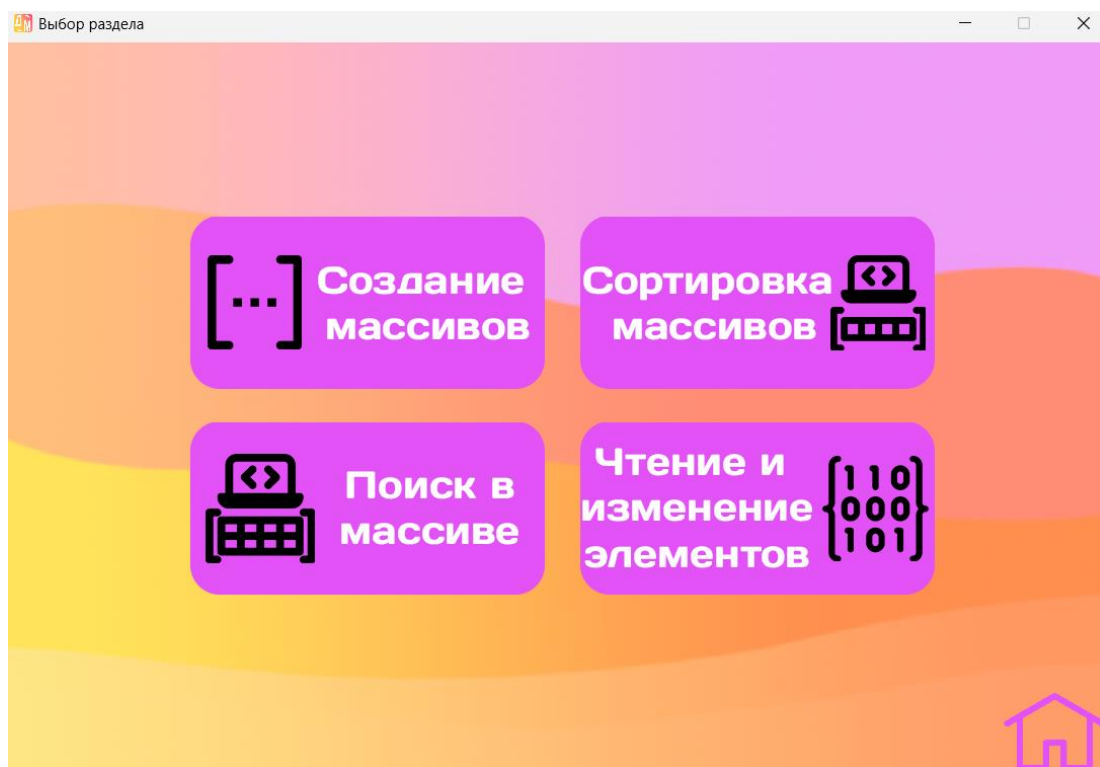


Рисунок 5 – Прототип окна, содержащего основные разделы программы

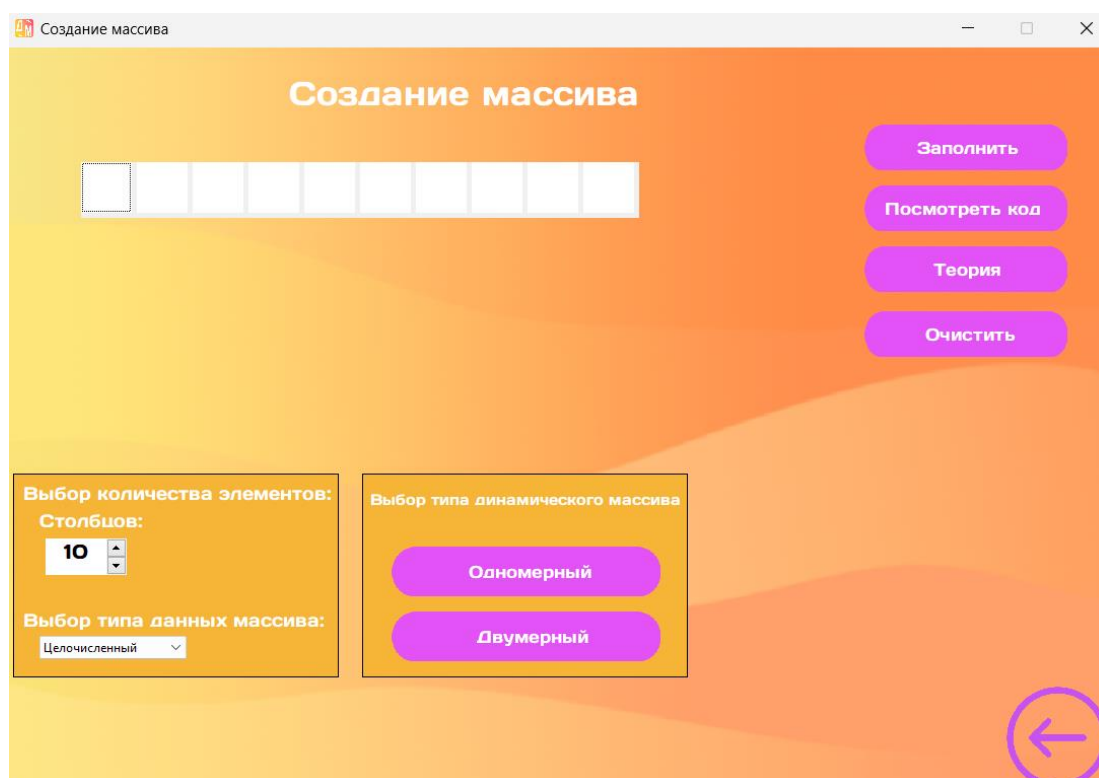


Рисунок 6 – Прототип окна, демонстрирующего визуализацию работы с динамическими массивами

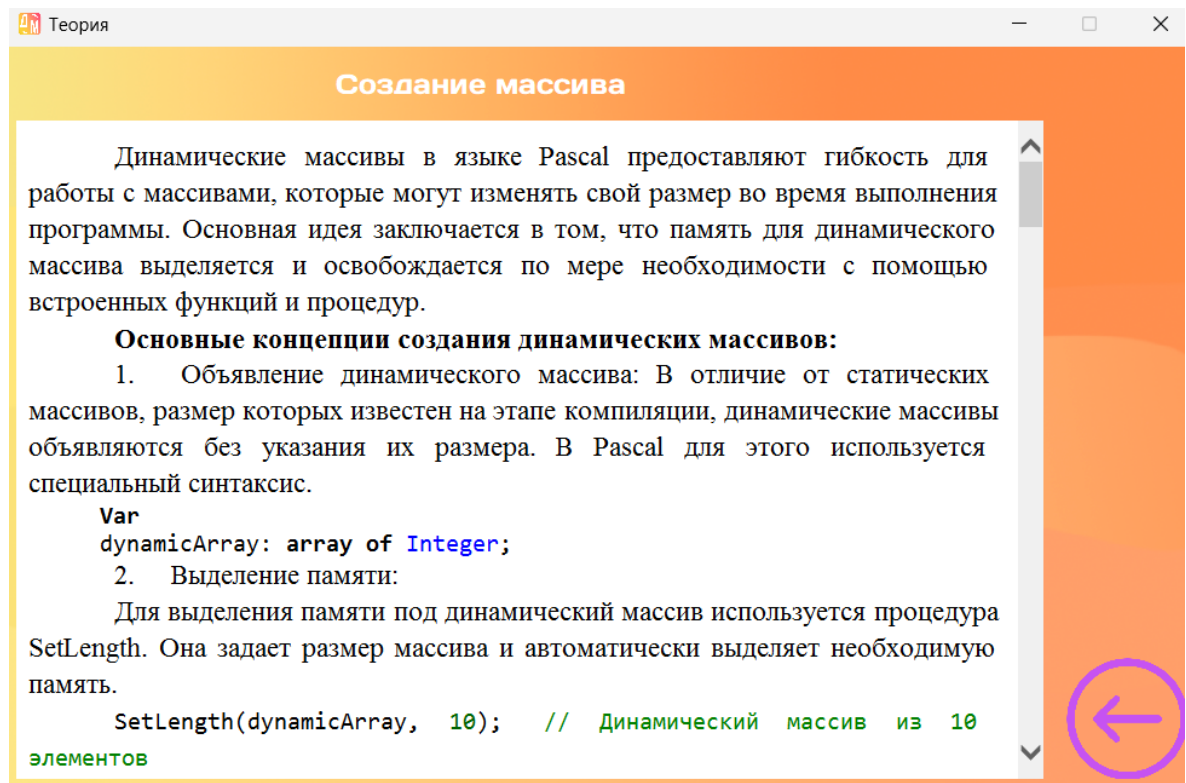


Рисунок 7 – Прототип окна, содержащего теорию по выбранному разделу визуализации

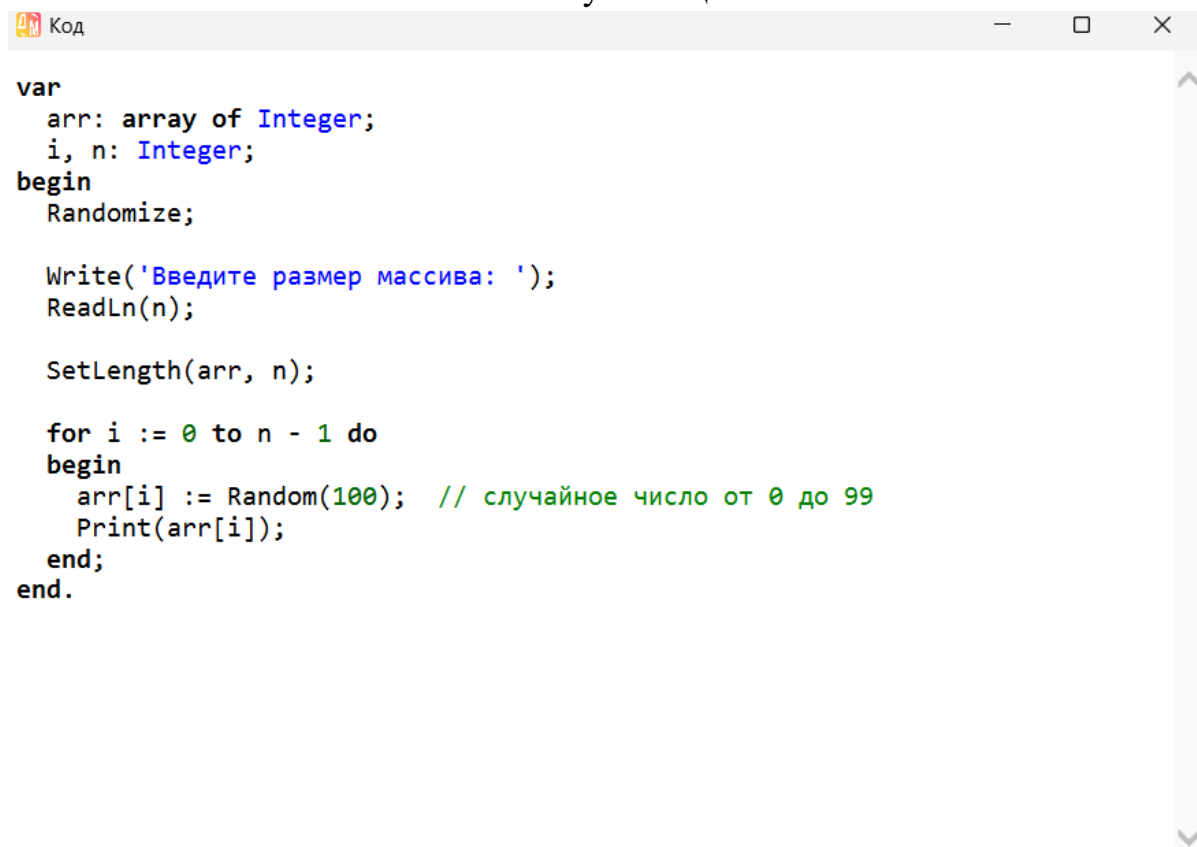


Рисунок 8 – Прототип окна, содержащего код программы

3 Реализация

3.1 Реализация проекта

3.1.1 Структура программы

Данный курсовой проект состоит из 6 модулей. Далее будет описано назначение каждого из них.

Модуль MainFormUnit является модулем, за которым закреплена главная форма с которой можно перейти на любую другую форму проекта, просмотреть справку о программе и справку о динамических массивах.

Модуль SplashFormUnit является модулем, который отвечает за заставку программы.

Модуль SectionsFormUnit отвечает за форму выбора раздела программы.

Модуль ArrayFormUnit является модулем, отвечающим за форму визуализации работы с динамическими массивами.

Модуль TheoryFormUnit является модулем, отвечающим за форму вывода теоретического материала.

Модуль CodeFormUnit является модулем, отвечающим за форму вывода кода программы.

3.1.2 Структура и описание процедур и функций пользователя

Описание разработанных процедур и функций приводится в таблице 1.

Таблица 1 – Процедуры и функции

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
procedure StartBtnClick(Sender: TObject);	MainFormUnit	StartBtn	Переход на следующую форму
procedure ExitBtnClick(Sender: TObject);	MainFormUnit	ExitBtn	Закрытие программы
procedure N3Click(Sender: TObject);	MainFormUnit	N3	Переход в «Создание массива»
procedure N4Click(Sender: TObject);	MainFormUnit	N4	Переход в «Создание массива»
procedure N5Click(Sender: TObject);	MainFormUnit	N5	Переход в «Сортировка массива»
procedure N6Click(Sender: TObject);	MainFormUnit	N6	Переход в «Поиск в массиве»

Продолжение таблицы 1

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
procedure FormCreate(Sender: TObject);	MainFormUnit	MainForm	Установка настроек при входе в программу
procedure FormDestroy(Sender: TObject);	MainFormUnit	MainForm	Удаление настроек при выходе из программы
procedure ManualBtnClick(Sender: TObject);	MainFormUnit	ManualBtn	Открытие справки по динамическим массивам
procedure N2Click(Sender: TObject);	MainFormUnit	N2	Открытие справки
procedure CreateArr_button- Click(Sender: TObject);	SectionsFormUnit	CreateArr_button	Переход в раздел «Создание массива»
procedure BackToHome_Button- Click(Sender: TObject);	SectionsFormUnit	BackToHome_Button	Кнопка возврата на главную форму
procedure SortArr_ButtonClick(Sender: TObject);	SectionsFormUnit	SortArr_button	Переход в окно «Создание массива»
procedure FindArr_ButtonClick(Sender: TObject);	SectionsFormUnit	FindArr_Button	Переход в раздел «Сортировка массива»
procedure EditArr_ButtonClick(Sender: TObject);	SectionsFormUnit	EditArr_Button	Переход в раздел «Поиск в массиве»
procedure FormClose(Sender: TObject; var Action: TCloseAction);	SectionsFormUnit	SectionsForm	Открытие главной формы при закрытии

Продолжение таблицы 1

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
procedure CreateParams(var Params: TCreateParams) ; override;	SectionsFormUnit	-	Не допускает закрытие ярлыка
procedure Timer1Timer(Sender: TObject);	SplashFormUnit	Timer1	Остановка таймера по завершению
procedure FormCreate(Sender: TObject);	SplashFormUnit	SplashForm	Запуск анимации
procedure FormShow(Sender: TObject);	TheoryFormUnit	TheoryForm	Прогрузка материала в web-browser
procedure SetTheoryID(NewTheoryID: Integer);	TheoryFormUnit	-	Установка ключа для загрузки материала
procedure Exit_buttonClick(Sender: TObject);	TheoryFormUnit	Exit_button	Выход на предыдущую форму
procedure Exit_ButtonClick(Sender: TObject);	ArrayFormUnit	Exit_Button	Выход на форму выбора раздела
procedure Gen_ButtonClick(Sender: TObject);	ArrayFormUnit	Gen_Button	Заполнение StringGrid
procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);	ArrayFormUnit	UpDown1	Изменение количества столбцовStringGrid
procedure one_dimensionalClick(Sender: TObject);	ArrayFormUnit	one_dimensional	Установка одномерного StringGrid
procedure two_dimensionalClick(Sender: TObject);	ArrayFormUnit	two_dimensional	Установка двумерного StringGrid
procedure UpDown2Click(Sender: TObject; Button: TUDBtnType);	ArrayFormUnit	UpDown2	Установление количества строк StringGrid

Продолжение таблицы 1

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
procedure StringGrid-SetEditText(Sender: TObject; ACol, ARow: Integer; const Value: string);	ArrayFormUnit	StringGrid	Проверка введенного типа данных в таблицу
procedure FormShow(Sender: TObject);	ArrayFormUnit	ArrayForm	Установка параметров при запуске формы
procedure SetMenuID(NewMenuID: Integer);	ArrayFormUnit	-	Установка параметров с предыдущей формы
procedure Sort_button-Click(Sender: TObject);	ArrayFormUnit	-	Запуск сортировки массива
procedure Sort();	ArrayFormUnit	-	Процедура сортировки
procedure SortDescending();	ArrayFormUnit	-	Процедура сортировки
procedure SelectionSort_Increase();	ArrayFormUnit	-	Процедура сортировки
procedure SelectionSort_Decreasing();	ArrayFormUnit	-	Процедура сортировки
procedure Bubblesort_Increase();	ArrayFormUnit	-	Процедура сортировки
procedure DisplayGrid();	ArrayFormUnit	-	Обновление содержимого StringGrid
procedure Bubblesort_Decreasing();	ArrayFormUnit	-	Процедура сортировки
procedure StringGridDrawCell(Sender: TObject; ACol, ARow: Integer; Rect: TRect; State: TGridDrawState);	ArrayFormUnit	StringGrid	Закрашивание ячеек StringGrid
procedure FormHide(Sender: TObject);	ArrayFormUnit	ArrayForm	Установка параметров при выходе из формы

Продолжение таблицы 1

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
procedure Find_Button-Click(Sender: TObject);	ArrayFormUnit	Find_Button	Запуск процедуры поиска
procedure Element_Iteration();	ArrayFormUnit	-	Процедура поиска
procedure Binary_search();	ArrayFormUnit	-	Процедура поиска
procedure ComboBox_Find-Change(Sender: TObject);	ArrayFormUnit	ComboBox_Find	Изменение параметров поиска при выборе в Combobox_Find
procedure Show_button-Click(Sender: TObject);	ArrayFormUnit	Show_button	Вывод элемента по индексу
procedure execute_button-Click(Sender: TObject);	ArrayFormUnit	execute_button	Изменение элементов StringGrid по индексу
procedure TheoryBtnClick(Sender: TObject);	ArrayFormUnit	TheoryBtn	Запуск окна теории
procedure CreateParams(var Params: TCreateParams) ; override;	ArrayFormUnit	-	Не допускает закрытие ярлыка
procedure ClearBtnImgClick(Sender: TObject);	ArrayFormUnit	ClearBtnImg	Очистка StringGrid
procedure CodebtnImgClick(Sender: TObject);	ArrayFormUnit	CodebtnImg	Открытие окна с кодом программы
procedure SetCodeID(NewCodeID: Integer);	CodeFormUnit	-	Установка значения для окна с кодом программы
procedure FormShow(Sender:TObject);	CodeFormUnit	CodeForm	Установка параметров при открытии окна

3.1.3 Описание использованных компонентов

Описание использованных при разработке приложения компонентов приводится в таблице 2

Таблица 2 – Использованные компоненты

Компонент	На какой форме расположено	Назначение
TMainMenu	MainForm	Используется для создания главного меню
TPanel	MainForm, SectionsForm, ArrayForm	Используется для группировки компонентов и в качестве кнопок
TImage	MainForm, SectionsForm, SplashForm, TheoryForm, ArrayForm	Используется как фон, скин на кнопки, элементы интерфейса
TTimer	SplashForm	Используется для отображения заставки
TWebBrowser	TheoryForm, CodeForm	Используется для загрузки веб-страниц
TLabel	TheoryForm, ArrayForm	Отображение надписей на форме
TComboBox	ArrayForm	Используется для выбора действий
TEdit	ArrayForm	Используется для ввода значений пользователем
TUpDown	ArrayForm	Используется для изменения количества строк и столбцов TStringGrid
TStringGrid	ArrayForm	Используется для визуализации работы с динамическими массивами
TMemo	ArrayForm	Используется для вывода текста
TShape	ArrayForm	Используется для создания фона в TPanel

3.2 Спецификация программы

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

Имя файла	Назначение
DynamicArrays.dproj	Файл проекта, связывает все файлы из которых состоит приложение.

Продолжение таблицы 3

Имя файла	Назначение
DynamicArrays.exe	Исполняемый файл проекта, используется для запуска программы на выполнение.
MainFormUnit.pas	Файл программного модуля главной формы
MainFormUnit.dfm	Главная форма
SectionsFormUnit.dfm	Форма выбора раздела
SectionsFormUnit.pas	Файл программного модуля формы выбора раздела
Setup.exe	Установочный пакет программы
SplashFormUnit.dfm	Форма заставки
SplashFormUnit.pas	Файл программного модуля формы заставки
TheoryFormUnit.dfm	Форма теории
TheoryFormUnit.pas	Файл программного модуля формы теории
ArrayFormUnit.dfm	Форма работы с массивом
ArrayFormUnit.pas	Файл программного модуля формы
CodeFormUnit.pas	Файл программного модуля формы вывода кода
CodeFormUnit.dfm	Форма вывода кода
Manual.chm	Файл справки
days.ttf	Файл установки шрифтов для работы программы
d3.gif	Анимация загрузки
down-arrow.png	Изображения кнопки
homepage.png	Изображения кнопки
кнопка пустая.png	Изображения кнопки
кнопка-1.png	Изображения кнопки
кнопка-2.png	Изображения кнопки
кнопка-3.png	Изображения кнопки
кнопка-4.png	Изображения кнопки
кнопкавыход.png	Изображения кнопки
кнопканачать.png	Изображения кнопки
кнопкасправка.png	Изображения кнопки
Логотип.png	Изображение логотипа
Логотип.ico	Иконка логотипа
надпись.png	Изображения для главной формы
фон.png	Фоновое изображение
фон2.png	Фоновое изображение

Продолжение таблицы 3

Имя файла	Назначение
Динамические массивы.htm	Теоретический материал
Поиск в массиве.htm	Теоретический материал
Создание массива.htm	Теоретический материал
Сортировка массива.htm	Теоретический материал
Чтение и изменение элементов.htm	Теоретический материал
бинарный поиск.htm	Текстовый материал
вещественный.htm	Текстовый материал
символьный.htm	Текстовый материал
целочисленный.htm	Текстовый материал
процедурой.htm	Текстовый материал
прямого выбора.htm	Текстовый материал
пузырьком.htm	Текстовый материал
удаление элемента.htm	Текстовый материал

4 Тестирование

В процессе разработки данной программы мы активно устраняли возникающие ошибки и недоработки. После завершения этапа написания кода мы провели тщательное функциональное тестирование. Это тестирование гарантировало, что все элементы программы работают корректно в автономном режиме.

Отчет о результатах тестирования представлен в таблице 4

Таблица 4 – Отчет о результатах тестирования

Идентификатор теста	Тест	Ожидаемый результат	Физический результат	Результат тестирования
1	Проверка входа пользователем в приложение	Открытие главного окна	Открытие главного окна	Выполнено
2	Проверка главного меню кнопки «Создание массива»	Открытие формы «Создание массива»	Открытие формы «Создание массива»	Выполнено
3	Проверка главного меню кнопки «Сортировка массива»	Открытие формы «Сортировка массива»	Открытие формы «Сортировка массива»	Выполнено
4	Проверка главного меню кнопки «Поиск в массиве»	Открытие формы «Поиск в массиве»	Открытие формы «Поиск в массиве»	Выполнено
5	Проверка главного меню кнопки «Изменение массива»	Открытие формы «Чтение и изменение элементов»	Открытие формы «Чтение и изменение элементов»	Выполнено
6	Проверка главного меню кнопки «О программе»	Открытие справки	Открытие справки	Выполнено
7	Проверка кнопки «Начать»	Открытие формы «Выбор раздела»	Открытие формы «Выбор раздела»	Выполнено

Продолжение таблицы 4

Идентифика- тор теста	Тест	Ожидаемый результат	Физический результат	Результат те- стирование
8	Проверка кнопки «Справка»	Открытие формы «Тео- рия»	Открытие формы «Тео- рия»	Выполнено
9	Проверка кнопки «Вы- ход»	Закрытие при- ложение	Закрытие при- ложение	Выполнено
10	Проверка кнопки «Со- здание масси- вов»	Открытие формы «Со- здание мас- сива»	Открытие формы «Со- здание мас- сива»	Выполнено
11	Проверка кнопки «Сор- тировка мас- сивов»	Открытие формы «Сор- тировка мас- сива»	Открытие формы «Сор- тировка мас- сива»	Выполнено
12	Проверка кнопки «По- иск в мас- сиве»	Открытие формы «По- иск в мас- сиве»	Открытие формы «По- иск в мас- сиве»	Выполнено
13	Проверка главного меню кнопки «Чтение и изме- нение элемен- тов»	Открытие формы «Чте- ние и измене- ние элемен- тов»	Открытие формы «Чте- ние и измене- ние элемен- тов»	Выполнено
14	Проверка кно- пок измене- ния размеров массива	Изменение количества строк и столб- цов массива	Изменение количества строк и столб- цов массива	Выполнено
15	Проверка па- нели выбора типа данных массива	Изменение типа данных массива	Изменение типа данных массива	Выполнено
16	Проверка кнопки «Од- номерный»	Массив стал одномерным	Массив стал одномерным	Выполнено
17	Проверка кнопки «Дву- мерный»	Массив стал двумерным	Массив стал двумерным	Выполнено

Продолжение таблицы 4

Идентифика- тор теста	Тест	Ожидаемый результат	Физический результат	Результат те- стирование
18	Проверка кнопки «За- полнить»	Заполнение массива эле- ментами по выбранным параметрам	Заполнение массива эле- ментами по выбранным параметрам	Выполнено
19	Проверка кнопки «По- смотреть код»	Открытие формы «Код»	Открытие формы «Код»	Выполнено
20	Проверка кнопки «Тео- рия»	Открытие формы «Тео- рия»	Открытие формы «Тео- рия»	Выполнено
21	Проверка кнопки «Очи- стить»	Очистка мас- сива	Очистка мас- сива	Выполнено
22	Проверка кнопки «Назад»	Переход на форму «Вы- бор раздела»	Переход на форму «Вы- бор раздела»	Выполнено
23	Проверка кнопки «До- мой»	Переход на главную форму	Переход на главную форму	Выполнено
24	Проверка па- нели выбора типа сорти- ровки массива	Смена типа сортировки	Смена типа сортировки	Выполнено
25	Проверка па- нели выбора сортировки массива	Смена сорти- ровки массива	Смена сорти- ровки массива	Выполнено
26	Проверка кнопки «От- сортировать»	Запуск проце- дуры сорти- ровки	Запуск проце- дуры сорти- ровки	Выполнено
27	Проверка меню выбора метода поиска	Смена метода поиска мас- сива	Смена метода поиска мас- сива	Выполнено
28	Проверка кнопки «Вы- полнить»	Запуск проце- дуры поиска в массиве	Запуск проце- дуры поиска в массиве	Выполнено

Продолжение таблицы 4

Идентификатор теста	Тест	Ожидаемый результат	Физический результат	Результат тестирования
29	Проверка кнопки «Вывести»	Вывод значения элемента по индексу	Вывод значения элемента по индексу	Выполнено
30	Проверка кнопки «Выполнить»	Выполнение выбранного действия над массивом	Выполнение выбранного действия над массивом	Выполнено
31	Проверка панели выбора действия над массивом	Изменения действия над массивом	Изменения действия над массивом	Выполнено

При разработке программного продукта было решено множество проблем, например, не осуществлялась проверка корректности вводимых значений пользователем, не очищался массив при выходе из формы, поэтому в будущем пользователь не столкнется с данными проблемами.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

5 Применение

5.1 Общие сведения о программном продукте

Цель данного проекта заключается в знакомстве начинающих программистов с такой структурой данных как динамические массивы. В программе пользователю предлагается ознакомиться с такими действиями в динамических массивах как:

- Создания массива;
- Сортировка массива;
- Поиск в массиве;
- Чтение и изменение элементов массива.

В этом программном продукте можно углубленно ознакомиться с динамическими массивами и увидеть как они реализованы на языке программирования PascalABC с помощью различных вариантов настроек и визуализаций процессов.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилась на моих друзьях, одноклассниках, работать с данной программой было комфортно.

Тестирование проводилось на разных классах ЭВМ. Одним из таких устройств был ПК со следующими системными характеристиками:

- Процессор Amd Ryzen 3 3200g;
- Графический адаптер Amd Radeon RX 5600XT;
- Объем ОЗУ 16Гб;
- Операционная система Windows 11 Pro.

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Установщик.exe. Появится окно установки приложения «Динамические массивы»

Затем достаточно следовать приведенной инструкции установки приложения.

5.3 Выполнение программы

5.3.1 Запуск программы

Данную программу можно запустить различными способами. Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием «Динамические массивы».

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\Козейко\Динамические массивы).

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		31

По подготовленным тестам будет осуществляться функциональное и полное тестирование программного продукта. Отчет о результатах тестирования будет представлен в 4 разделе пояснительной записки.

5.3.2 Инструкции по работе с программой

При запуске программы вы попадаете на форму «Главное меню» (Рисунок 8).

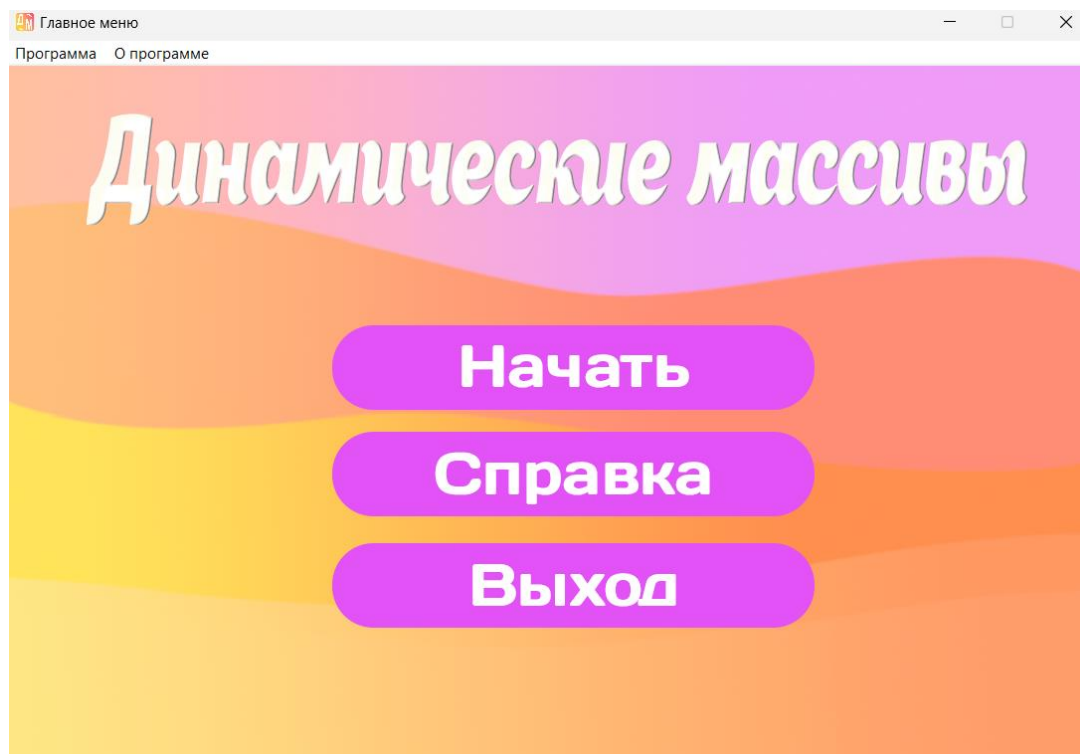


Рисунок 8 – Главное меню

В главном меню можно увидеть такие элементы как:

- Название программы;
- Кнопка «Начать» для перехода к программе;
- Кнопка «Справка» которая содержит информацию о динамических массивах;
- Кнопка "Выход" для выхода из программы;
- Основное меню, которое содержит такие кнопки как:
 - 1) Создание массива
 - 2) Сортировка массива
 - 3) Поиск в массиве
 - 4) Чтение и изменение элементов
 - 5) О программе, для вызова этой справки

Если вы выберете «Начать» вам откроется форма для выбора раздела работы с динамическими массива (Рисунок 9).

На этой форме находятся 4 кнопки: «Создание массива», «Сортировка массива», «Поиск в массиве», «Чтение и изменение массива». При нажатии одной из кнопок вам откроется соответствующая разделу форма (Рисунок 10).



Рисунок 9 – Меню выбора раздела

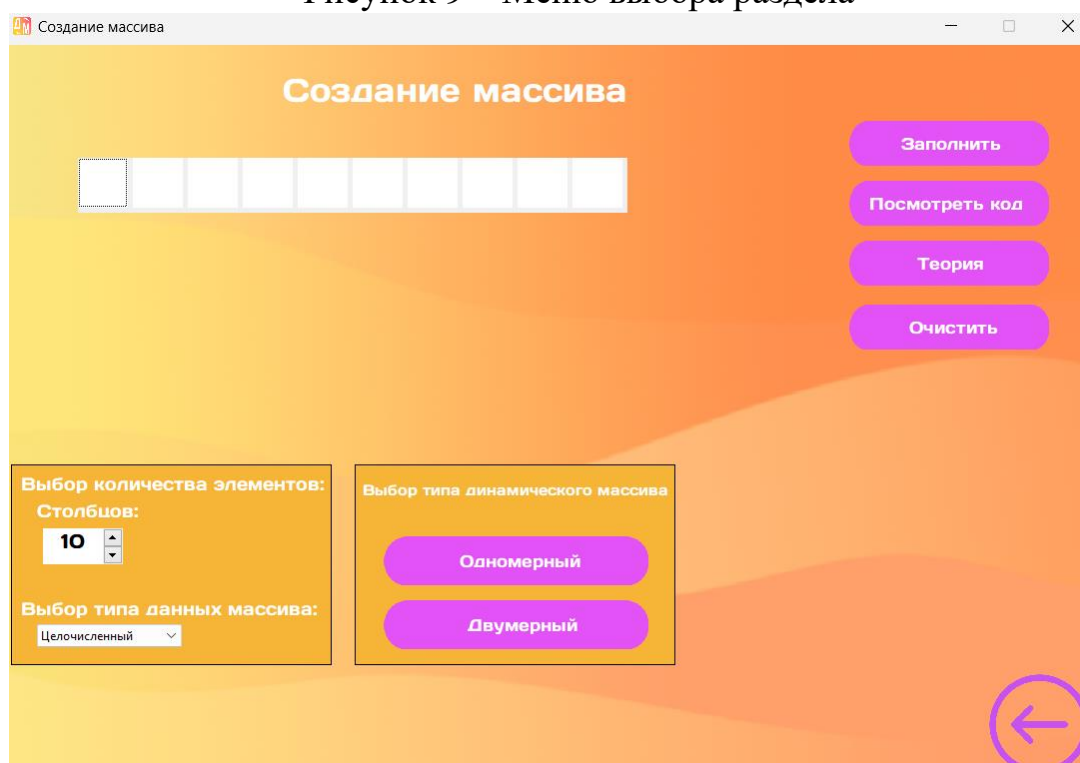
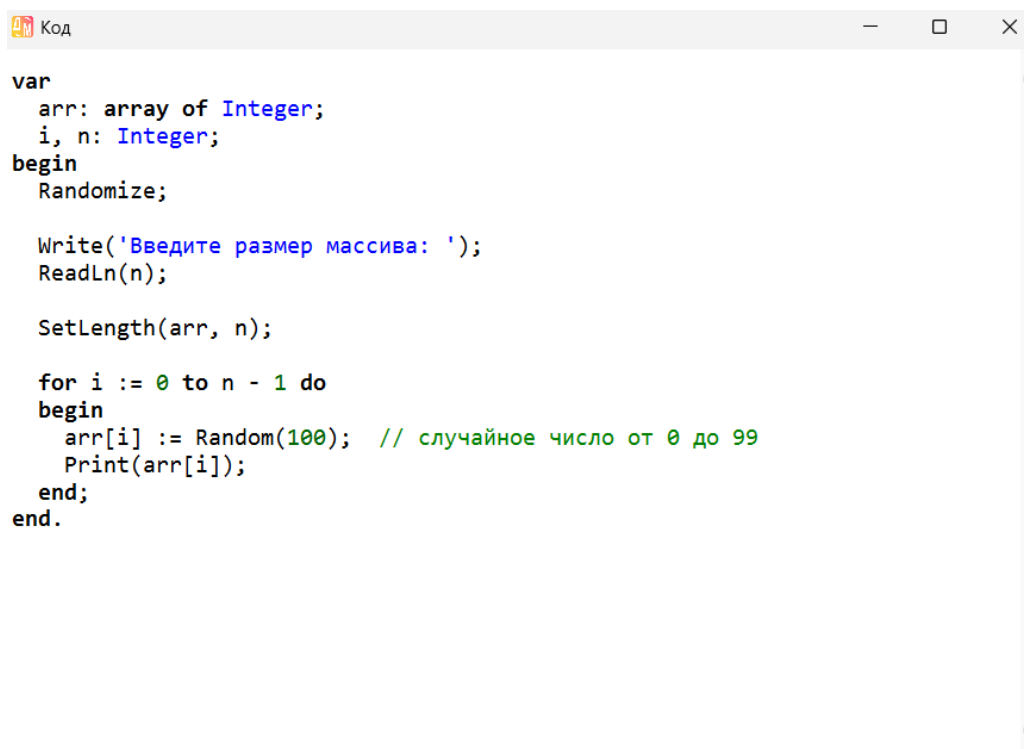


Рисунок 10 – Раздел "Создание массива"

В каждой выбранной по разделу форме в нижней части экрана буду загружаться панели в которых можно выполнять различные действия над массивами.

Во всех формах выбираемых разделов будут обязательно присутствовать такие кнопки как: «Заполнить», «Посмотреть код», «Теория» и «Очистить».

Кнопка "Заполнить" заполняет массив случайным значениям. Кнопка «Посмотреть код» открывает окно с кодом программы которую вы выбрали (Рисунок 11). Кнопка «Теория» открывает окно с теорией по выбранному разделу (Рисунок 12). Кнопка «Очистить» очищает весь массив.



```

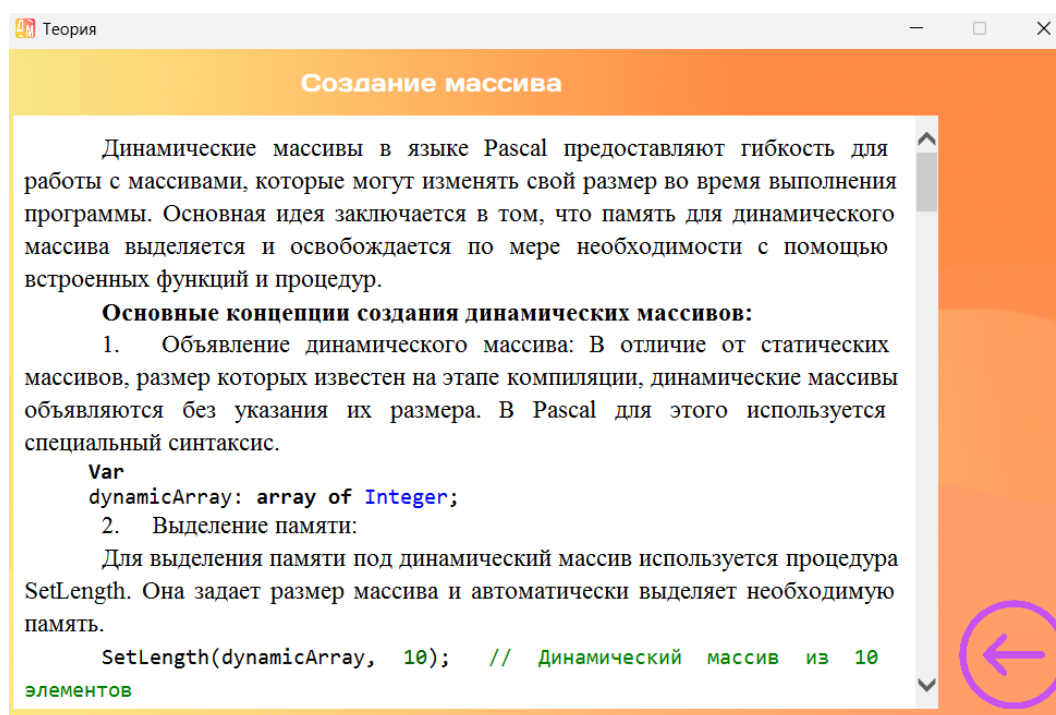
Код
var
  arr: array of Integer;
  i, n: Integer;
begin
  Randomize;

  Write('Введите размер массива: ');
  ReadLn(n);

  SetLength(arr, n);

  for i := 0 to n - 1 do
  begin
    arr[i] := Random(100); // случайное число от 0 до 99
    Print(arr[i]);
  end;
end.
  
```

Рисунок 11 – Форма с кодом



Создание массива

Динамические массивы в языке Pascal предоставляют гибкость для работы с массивами, которые могут изменять свой размер во время выполнения программы. Основная идея заключается в том, что память для динамического массива выделяется и освобождается по мере необходимости с помощью встроенных функций и процедур.

Основные концепции создания динамических массивов:

1. **Объявление динамического массива:** В отличие от статических массивов, размер которых известен на этапе компиляции, динамические массивы объявляются без указания их размера. В Pascal для этого используется специальный синтаксис.
`Var
dynamicArray: array of Integer;`
2. **Выделение памяти:**
 Для выделения памяти под динамический массив используется процедура SetLength. Она задает размер массива и автоматически выделяет необходимую память.
`SetLength(dynamicArray, 10); // Динамический массив из 10 элементов`

Рисунок 12 – Форма с теорией

5.3.3 Завершение работы с программой

Завершить работу с программой можно несколькими способами:

- Нужно нажать на крестик в углу формы;
- Также можно нажать кнопку выход на главной форме (Рисунок 13).



Рисунок 13 – Кнопка выхода

5.4 Использование системы справочной информации

Справочную систему можно запустить с помощью пункта меню «О программе» (Рисунок 14).

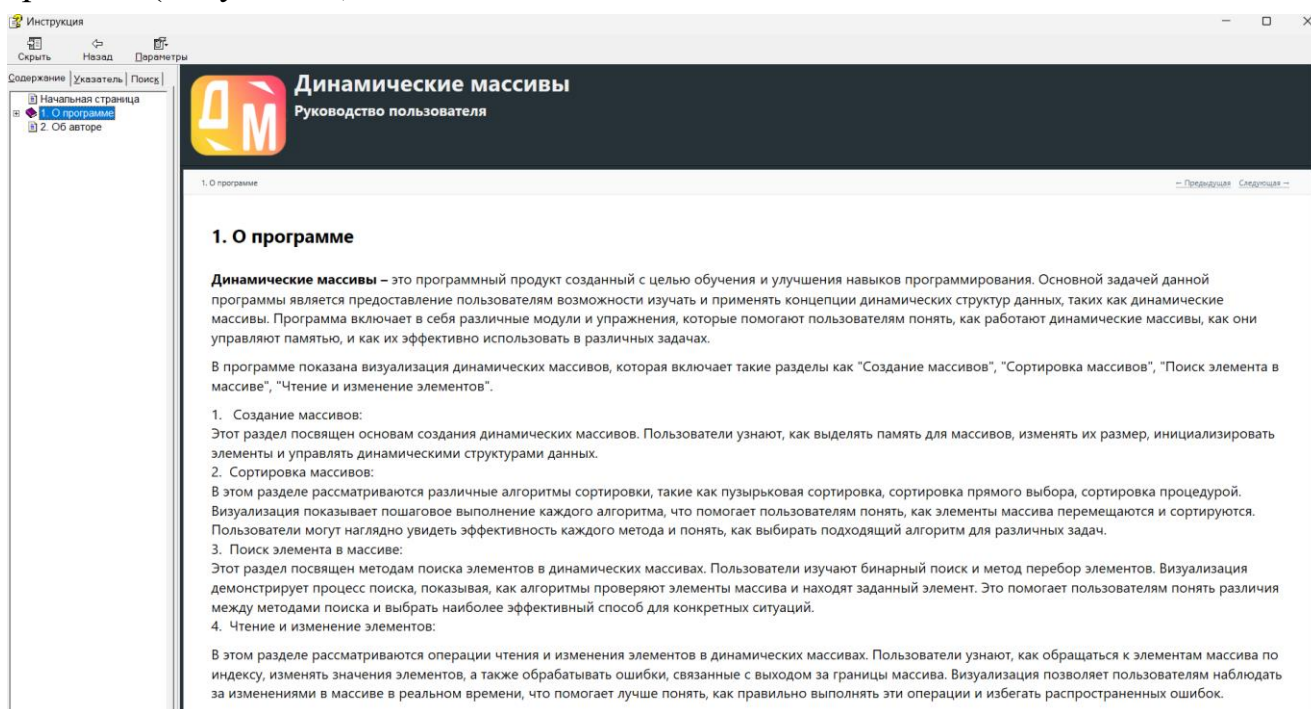


Рисунок 14 – Запуск справки

Заключение

В ходе данного курсового проекта была выполнена разработка программного продукта, предназначенного для обучения студентов и начинающих программистов основам работы с динамическими массивами. Проект предоставляет интерактивную платформу, которая визуализирует процессы создания, модификации и управления динамическими массивами, тем самым облегчая понимание этих концепций.

Используемая среда разработки Delphi обеспечила высокую производительность и кроссплатформенность, что позволило создать удобный и функциональный интерфейс для взаимодействия с пользователем. Дополнительные инструменты, такие как Adobe Photoshop и Microsoft Office, были применены для создания графических элементов и документации, что повысило общую привлекательность и информативность программы.

Основные задачи, которые автоматизирует данное приложение, включают создание массивов различных типов данных, автоматическое изменение их размера, добавление и удаление элементов, а также сортировку и поиск данных в массивах. Особое внимание было уделено интерфейсу программы, чтобы он был интуитивно понятным и удобным для пользователей, независимо от их опыта работы с программированием.

В результате проект не только упрощает процесс обучения работе с динамическими массивами, но и предоставляет полезный инструмент для преподавателей и студентов, способствующий лучшему пониманию и закреплению знаний в данной области. Проект отвечает всем установленным требованиям и демонстрирует стабильную и надежную работу, минимизируя возможные ошибки и обеспечивая отказоустойчивость.

Таким образом, данный курсовой проект представляет собой ценный вклад в образовательные ресурсы по программированию, предлагая эффективный и доступный способ изучения и практики работы с динамическими массивами.

					КП 2-40 01 01.33.4.1.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		36

Список использованных источников

1. Культин, Н. Основы программирования в Delphi 7. - Санкт-Петербург: Питер, 2009. - 640 с.
2. Лишнер Р. Delphi. Справочник. - Пер. с англ. - СПб: Символ-Плюс, 2001. - 640 с.
3. Уроки Delphi начинающим с нуля [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://delphi-manual.ru/>. – Дата доступа: 20.06.2024
4. 100 компонентов Delphi [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://beluch.ru/progr/100comp.htm>. – Дата доступа: 18.06.2024
5. DelphiSchool, уроки программирования, примеры, обучение [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://delphischool.3dn.ru/>. – Дата доступа: 17.06.2024
6. ChatGPT [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://chatgpt.com/>. – Дата доступа: 21.06.2024

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		37

Приложение А
Листинг программы

					КП 2-40 01 01.33.41.10.24 ПЗ	Лист
Изм.	Лист	№докум.	Подпись	Дата		38