Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина:  Основы алгоритмизации и программирования (ОАиП)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

ПРОГРАММНОЕ СРЕДСТО УЧЕТА РАБОТ САДОВОДА

БГУИР 6-05-0612-01-110

        Студент:  гр. 351001

        Орлович Д.С.

        Руководитель:

        асс. Фадеева Е.Е

Минск 2024

**Содержание**

[**СОДЕРЖАНИЕ** 4](#_Toc167236934)

[**ВВЕДЕНИЕ** 5](#_Toc167236935)

[**1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ (ПС)** 6](#_Toc167236936)

[**1.1 Назначение разработки** 6](#_Toc167236937)

[**1.2 Состав выполняемых функций** 6](#_Toc167236938)

[**1.3 Входные и выходные данные** 6](#_Toc167236939)

[**2** **АНАЛИЗ ТРЕБОВАНИЙ К ПС И** **РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ  ТРЕБОВАНИЙ** 7](#_Toc167236940)

[**2.1 Описание функциональности системы** 7](#_Toc167236941)

[**2.2 Спецификация функциональных требований** 8](#_Toc167236942)

[**3   ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА** 9](#_Toc167236943)

[**3.1 Проектирование основных динамических структур** 9](#_Toc167236944)

[**3.1.1 Структура данных о дачном участке** 9](#_Toc167236945)

[**3.1.2 Структура данных о посадках** 10](#_Toc167236946)

[**3.1.3 Структура данных о журнале работ** 10](#_Toc167236947)

[**3.2 Проектирование алгоритмов работы динамическими структурами** 10](#_Toc167236948)

[**3.2.1 Алгоритм отображения дачного участка** 10](#_Toc167236949)

[**3.3 Проектирование алгоритмов взаимодействия различных компонентов программы** 11](#_Toc167236950)

[**3.3.1 Алгоритм взаимодействия компонентов программы с данными о грядках и культурах** 11](#_Toc167236951)

[Журнал работ и книга посадок имеют возможность добавления или отметки какой-то культуры или грядки. Соответственно требуется проверка на наличие в программе кода грядки, нужного для добавления. Для перед каждым случаем добавления грядки происходит проверка на наличие самого кода грядки. 11](#_Toc167236952)

[Перед каждым случаем, когда пользователь хочет отметить какую-либо культуру, программному средству понадобится проверить запись культур на наличие такого кода культуры. 11](#_Toc167236953)

**ВВЕДЕНИЕ**

Желание, повысить уровень или объёмы производства возможно удвалетворить, при создании специального программного средства. Садоводы занимаются выращиванием овощей, фруктов, цветов и других растений на своих участках. Кроме того, стоит отметить, что выращивание растений имеет широкое общественное значение. Оно способствует экологической устойчивости, предоставляет людям свежие и качественные пищевые продукты, а также способствует физическому и эмоциональному благополучию отдельных садоводов.

Это самое программное средство по учёту работ садовода предназначено облегчить и ускорить работу с садовыми участками. Причиной создания программного связана с ростом размеров и сложности устройства садовых участков и соответствующая им сложность обслуживания этих участков.  Управление и учет работ, связанных с садоводством, может быть сложной задачей, особенно при увеличении размеров участка или числа выращиваемых культур.

Под дачным участком или садовым участком подразумевается, некоторая ограниченная территория, состоящая из грядок с растениями, которую в последствии возможно сделать другой грядкой с соответственно другим именем и другими растениями.

Цель данной курсовой работы состоит в изучении и создании программного средства для учета работ садовода. Предоставление программным средством необходимых инструментов служит для повышения эффективности и удобства управления собственным садовым участком и благодаря этому возможно достичь более качественных результатов в выращивании и уходом за растениями. Будут проанализированы различные способы взаимодействия с пользователем. Будут описаны необходимые алгоритмы и предложены их схемы, кроме того, будет проанализирована программа и корректность ее работы с реальными данными.

На основе требований к программному средству будет разработана архитектура программного средства и определены его функциональные возможности. В задачах так же стоит разработать принципы работы с данными, необходимыми для работы программы. И способы их хранения и доступа к ним.

**1 АНАЛИЗ ПРОТОТИПОВ, ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОЕТИРУЕМОМУ ПРОГРАММНОМУ СРЕДСТВУ (ПС)**

**1.1 Назначение разработки**

Назначение разработки: создать интуитивно-понятный принцип работы, интерфейс и реализовать, доступное для массового пользователя, программное средство для работы с дачным хозяйством. И свести к минимуму временные ресурса пользователя.

Предоставить пользователю возможность добавление растения на определённую грядку. И добавить описание к соответствующей грядке.

Пользователю будет дана возможность так же вести журнал работ и производить поиск какой-либо работы.

**1.2 Состав выполняемых функций**

1. Чтение данных из файлов, хранящие различные данные пользователя;
2. Организация взаимодействия с пользователем;
3. Определение записи, к которой необходимо обратиться для изменения данных в ней;
4. Редактирование данных о дачном хозяйстве в процессе работы;
5. Возможность изменять масштаб дачного участка.
6. Сохранение изменённых данных.
7. Наглядная демонстрация пользователю количества растений на какой либо грядке.
8. Возможность просмотра места, где растёт растение.

**1.3 Входные и выходные данные**

На входе: вводимые данные пользователем, файлы с типом о дачном хозяйстве;

На выходе: файлы с типом об обновлённом дачном хозяйстве с информацией об одном пользователе;

**2** **АНАЛИЗ ТРЕБОВАНИЙ К ПС И** **РАЗРАБОТКА ФУНКЦИОНАЛЬНЫХ  ТРЕБОВАНИЙ**

**2.1 Описание функциональности системы**

Работа с различными частями разрабатываемого программного средства должна быть предоставлена пользователю в комфортной форме.

При запуске программы сразу же начинается чтение из различных файлов данных о дачном участке пользователя. Дачный участок представлен в виде двумерного массива, каждый элемент которого хранит информацию о грядке и растении, которое растёт на грядке. Различные грядки помечаются разными цветами. Группа элементов массива дачного участка, имеющих общее имя(цвет), считается грядкой. Файл с грядками одна из характеристик каждой грядки это список ячеек массив, относящихся к этой грядке. Файл с списком посадок, любая из которых имеет дату посадки, статус, год посадки и строго определённый код. В программе предполагается взаимодействие с только теми культурами, которые пользователь решил создать сам или оставить из предложенных при первом запуске.

Пользователь должен иметь в доступе функции для удаления какой-либо культуры или грядки и функции для добавления грядки или культуры. Для добавления нового элемента(культуры или грядки) необходимо соблюсти определённые правила. Функции внесения новых данных предполагает создание новой записи. Все ети функции должны быть доступны пользователю напрямую.

Если функция требует более широкого объяснение, чем просто название, то по наводке мыши должно появляться дополнительное описание функции.

Завершение программы возможно при нажатии кнопки специально отведённой под эту задачу. Сохранение или несохранение данных определяется желание пользователя. Возможность сохранения доступна при нажатии определённой комбинации клавиш и кнопки на окне программы.

**2.2 Спецификация функциональных требований**

Основные функции программного средства должны быть спроектированы и реализованы в соответствии со следующими требованиями:

**2.2.1 Функция заполнения грядками садового участка:**

* 1. Выполняется при окончании чтения файла с грядками.
  2. Пользователь может иметь возможность как-либо косвенно изменять содержание отдельного элемента массива.
  3. Обязательно чтобы каждый элемент массива принадлежал к грядке.
  4. Отображение дачного участка происходит в соответствие с текущим масштабом, определённым пользователем.

**2.2.2 Функция чтения файла с информацией о грядках:**

1. Поиск нужного файла осуществляется по уникальному имени и имеет расширение \*.cul.
2. Данная функция начинает выполняться при запуске программного обеспечения и только после прочтения файла с культурами.
3. Содержит информацию о всех грядках.
4. Каждая грядка должна иметь абсолютно уникальную комбинацию элементов массива садового участка.
5. Имя грядки должно быть уникально.
6. Цвет грядки должен быть уникальным.

**2.2.3 Функция просмотра сада:**

1. Следует иметь возможность демонстрации расположения культур на всём садовом участке.
2. Возможность выбора грядки доступна двумя способами.
3. По выбору грядки должен произойти подсчёт и вывод информации об культурах на этой грядке.
4. Это функция не может изменить данные об садовом участке а только использовать эти данные

**2.2.4 Функция выделения элементов сада:**

1. Происходит двумя методами (одиночное и множественное).
2. Множественное выделение должно происходить с верхнего левого угля в противоположный.
3. Множественное выделение происходить при удержании левой кнопки мыши.
4. При одиночном выборе окно редактирование уже содержит информацию об выделенном элементе.

**2.2.5 Функция удаления записи грядки:**

1. Удаление грядки доступно только при выборе грядки в окне изменения грядки.
2. Удаление разрешено при условии, что она не используется в данный момент.
3. После удаление грядки её имя и цвет становятся доступными для использования в создании новой грядки.

**2.2.6 Функция удаления записи грядки:**

1. Удаление культуры доступно только при выборе культуры в окне изменения грядки.
2. Удаление разрешено при условии, что она не используется в данный момент.
3. После удаление культуры её название становятся доступными для использования в создании новой грядки.

**2.2.7 Функция сохранения изменений:**

1. Данная функция начинает своё выполнение после нажатия пользователем определённой кнопки
2. Сохраняет изменения во все файлы.
3. После сохранения пользователю выводится сообщение об успешном сохранении.

**3   ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА**

**3.1 Проектирование основных динамических структур**

**3.1.1 Структура данных о дачном участке**

Дачный участок - это просто двумерный массив, каждый элемент которого просто запись, включающая в себя запись культуры и номер грядки, к которой этот элемент относиться.

Элемент дачного участка содержит поля:

1. Код грядки
2. Запись о культуре

Код культуры устанавливается в файле, со всеми возможными кодами культуры. Каждому коду ставится в соответствие своё имя культуры.

Так же для удобства работы с грядками, к которым принадлежат различные элементы массива, было решено сделать отдельную запись под грядки, в которой хранится все имена грядок и их номера.

**3.1.2 Структура данных о цвете грядки**

 Для обеспечения функциональности сопоставления номера грядки с уникальным цветом в программном средстве, необходимо создать способ определения цвета грядки по её номеру.

Предлагается использовать структуру данных, по функционалу похожую на хэш-таблицу, с ключом и значением:

Номер грядки(ключ) – целочисленное значение, которое будет однозначно идентифицировать каждую грядку в саду.

Цвет грядки(значение) – уникальный цвет самой грядки

**3.1.3 Структура данных о списке культур**

Список культур - это динамическая структура данных, содержащая записи о культуре и указатель на следующую запись.

Запись культуры представляет из себя просто несколько полей:

1. Срок произрастания.
2. Уникальное имя культуры.
3. Уникальный код культуры.

**3.1.4 Структура данных о списке грядок**

Список грядок - это динамическая структура данных, содержащая записи о грядке и указатель на следующую запись.

Запись грядки представляет из себя просто несколько полей:

1. Уникальный код грядки.
2. Уникальное имя грядки.

**3.2 Проектирование алгоритмов работы динамическими структурами**

**3.2.1 Алгоритм отображения дачного участка**

Изначально для определения области, в которой будет происходить отрисовка дачного участка, определяется размер каждого элемента дачного участка и их количества в каждом направлении.

Для более удобного вывода дачного участка отрисовка происходит в центре области для отрисовки, которая меняет при изменении масштаба.

При изменении масштаба происходит новый расчёт центра отрисовки и размера элементов садового участка.

При выборе пользователем элементов цвет выделения зависит от цвета выделяемого элемента, который определяется координатой нажатия мыши.

**4 СОЗДАНИЕ ПРОГРАММНОГО СРЕДСТВА**

**4.1 Проектирование модулей программного средства**

Программное средство разрабатывается на основе спроектированных алгоритмов (см. раздел 3) и спецификации функциональных требований (см. раздел 2). Программное средство имеет модули:

1. uMain – модуль основной формы программы;
2. uMain\_logical – вспомогательный модуль основной формы выполняющий работу с файлами.
3. uGarden – модуль формы об изменениях грядки;
4. uGarden\_logical – вспомогательный модуль изменения грядки, выполняющий проверки и изменения данных о садовом участке;
5. uAddNewCulture – модуль формы добавления культуры;
6. uAddNewGarden – модуль формы добавления грядки;
7. uShowAllGardens – модуль формы просмотра садового участка;
8. uDictionary – модуль с вспомогательной структурой данных;
9. uGlobalData – модуль с функциями и переменными участвующими в большей части программы;
10. uSP – модуль формы о справке программного средства;

**4.2 Разработка модулей программного средства**

Таблица 4.1 – Описание подпрограмм.

|  |  |  |
| --- | --- | --- |
| Имя и тип под- программы | Назначение подпрограммы | Параметры, назначение и возвращаемые значения |
| Procedure FormCreate | Чтение всех данных из файлов, инициализация основный структур данных | (Sender: TObject);  Sender – объект работы процедуры (форма) |
| Procedure PaintBox1MouseDown | Определение начального элемента выделения грядок | (Sender: TObject; Button: TMouseButton;  Shift: TShiftState; x, y: integer)  Sender – объект работы процедуры (область для рисования)  Shift – состояние клавиши Shift на момент возникновения события  X,Y – положение курсора |
| Procedure PaintBox1Click | Открытие формы изменения грядок | (Sender: TObject);  Sender – объект работы процедуры (область для рисования) |
| Procedure PaintBox1MouseLeave | Стирание последнего выделения на садовом участке | (Sender: TObject);  Sender – объект работы процедуры (область для рисования) |
| Procedure PaintBox1MouseMove | Отрисовка выделенных грядок и определение текущего(крайнего элемента выделения) | (Sender: TObject;  Shift: TShiftState; x, y: integer)  Sender – объект работы процедуры (область для рисования)  Shift – состояние клавиши Shift на момент возникновения события  X,Y – положение курсора |
| Procedure PaintBox1Paint | Отрисовка всего дачного участка | (Sender: TObject);  Sender – объект работы процедуры (область для рисования) |
| Procedure ActionMinusExecute | Уменьшение масштаба дачного хохзяйства | (Sender: TObject);  Sender – объект работы процедуры (список событий формы) |
| Procedure ActionPlusExecute | Увеличение масштаба дачного хозяйства | (Sender: TObject);  Sender – объект работы процедуры (список событий формы |
| Procedure ActionShowAllExecute | Отображение формы показа дачного хозяйства | (Sender: TObject);  Sender – объект работы процедуры (список событий формы |
| Procedure ActionSaveExecute | Сохранение всех данных в файлы | (Sender: TObject);  Sender – объект работы процедуры (список событий формы |
| Procedure FormShow | При отображении основной формы запускается передача в форму изменения грядок определённые структуры | (Sender: TObject);  Sender – объект работы процедуры (форма) |
|  |  |  |
| Procedure createLists | Инициализирует необходимые для работы формы списки | (Culturelist: PtCulture; Gardenlist: PtGarden);  Culturelist – динамический список культур  Gardenlist – динамиеский список грядок |
| Procedure ActionAddCultureExecute | Отображение формы добавления культуры. Добавление происходит в зависимости от итога закрытия формы добавления | (Sender: TObject);  Sender – объект работы процедуры (список событий формы) |
| Procedure ActionAddGardenExecute | Отображение формы добавления грядки. Добавление происходит в зависимости от итога закрытия формы добавления | (Sender: TObject);  Sender – объект работы процедуры (список событий формы) |
| Procedure ActionDeleteCultureExecute | Удаление записи о выбранной культуре | (Sender: TObject);  Sender – объект работы процедуры (список событий формы) |
| Procedure ActionDeleteGardenExecute | Удаление записи о выбранной грядке | (Sender: TObject);  Sender – объект работы процедуры (список событий формы) |
| Procedure ButtonSaveClick | Применение изменений грядки к массиву дачного участка | (Sender: TObject);  Sender – объект работы процедуры (кнопка) |
| function FormShowForChange | Редактирование только одной грядки | (CurrPoint: TPoint);  CurrPoint – точка хранящая индексы элемента дачного участка для редактирования |
| function FormShowForChangeCtrl | Редактирование только нескольких грядок | (CurrPoint, StartPoint: TPoint);  CurrPoint – начальная точка(верхняя левая) хранящая индексы элемента дачного участка для редактирования  StartPoint – конечная точка(нижняя правая) хранящая индексы элемента дачного участка для редактирования |
| Procedure ListViewCultureClick | Определение выбранной записи культур | (Sender: TObject);  Sender – объект работы процедуры (список) |
| Procedure ListViewGardenClick | Определение выбранной записи грядок | (Sender: TObject);  Sender – объект работы процедуры (список) |
| Procedure CreateListViewGarden | Создание строк грядок в компоненте формы, отображающей все грядки | (list: PtGarden);  list – динамиеский список грядок |
| Procedure CreateListViewCulture | Создание строк культур в компоненте формы, отображающей все культуры | (list: PtCulture);  list – динамиеский список культур |
|  |  |  |
| function ShowForAdd | Инициализирует необходимые для работы формы списки и передача элемента для возможного добавления | (var culture: cultureListInfo; culturelist: ptculture);  Culture – добавляемая культура  culturelist– динамиеский список культур |
| function IsValidInputData | Проверка на корректность введённых данных культуры | (const strName, const strTime: string)  strName - имя добавляемой грядки  strTime – время созревания культуры, представленной в строке |
| Procedure ButtonSaveClick | Сохранение данных культуры (элемент будет добавлен) | (Sender: TObject);  Sender – объект работы процедуры (кнопка) |
| procedure ButtonCancelClick | Не сохраняет данные о культуре (элемент не будет добавлен) | (Sender: TObject);  Sender – объект работы процедуры (кнопка) |
|  |  |  |
| procedure IsValidInputData | Инициализирует необходимые для работы формы списки и передача элемента для возможного добавления | (const strName: string; const color: TColor);  strName – имя добавляемой грядки  color – цвет добавляемой грядки |
| function ShowForAdd | Проверка на корректность введённых данных грядки | (var garden: TGARDEN; gardenlist: ptgarden)  garden – добавляемая грядка  gardenlist – динамиеский список грядок |
| procedure ButtonSaveClick | Сохранение данных грядки (элемент будет добавлен) | (Sender: TObject);  Sender – объект работы процедуры (кнопка) |
| procedure ButtonCancelClick | Не сохраняет данные о грядке (элемент не будет добавлен) | (Sender: TObject);  Sender – объект работы процедуры (кнопка) |
|  |  |  |
| Function FormShowAll | Инициализирует необходимые для работы формы списки | (culturelist: ptCulture; gardenlist: PtGarden);  Culturelist – динамический список культур  Gardenlist – динамиеский список грядок |
| procedure FormShow | Начальное отображение на компоненте всех культур и грядок | (Sender: TObject);  Sender – объект работы процедуры (форма) |
| Procedure FormCreate | Создание нужного формата компонентов форм | (Sender: TObject);  Sender – объект работы процедуры (форма) |
| Procedure PrintSelectedGarden | Выводит выбранную из списка компонента формы грядку | (const idGarden: integer);  idGarden – индекс выдранной грядки |
| Procedure PaintBox1MouseDown | Выводит выбранную по нажатию грядку | (Sender: TObject; Button: TMouseButton; Shift: TShiftState; x, y: integer);  Sender – объект работы процедуры (область для рисования)  Shift – состояние клавиши Shift на момент возникновения события  X,Y – положение курсора |
| Procedure PrintSelectedCulture | Отображает культуры на дачном участке | (const nameCulture: string);  nameCulture – имя выбранной строки |
| Procedure CreateListCultureAll | Выводит на компонент формы все культуры на дачном участке | (CultutrList: ptCulture);  Culturelist – динамический список культур |
| Procedure CreateListViewGarden | Выводит на компонент формы все грядки на дачном участке | (gardenlist: PtGarden);  Gardenlist – динамиеский список грядок |
| Procedure CreateListViewCulture | Выводит на компонент формы все культуры на дачном участке принадлежащие какой-либо выбранной грядке | (const idGarden: array of integer);  idGarden – массив все грядок для отображения |
| Procedure printAllGarden | Отрисовывает весь садовый участок | - |
|  |  |  |
| procedure SaveFileGarden; | Запись в файл всех грядок в садовом участке | (list: PtGarden);  list – динамический список грядок |
| procedure ReadFileGarden; | Чтение из файла всех грядок в садовом участке | (list: PtGarden);  list – динамический список грядок |
| procedure ReadFileCulture; | Чтение из файла всех культур в садовом участке | (list: PtCulture);  list – динамический список культур |
| procedure SaveFileCulture; | Запись в файл всех культур в садовом участке | (list: PtCulture);  list – динамический список культур |
| procedure ReadFileColor; | Чтение файла с индексами грядок и их цвета | - |
| procedure SaveFileColor; | Запись в файл всех индексов грядок и из цветов | - |
| procedure SaveGardenMas; | Запись в файл массива дачного участка | - |
| procedure ReadGardenMas; | Чтение из файла массива дачного участка | - |
|  |  |  |
| function CheckCultureUsing | Проверка на использование в садовом участке определённой культуры | (const culture: string);  Culture – имя культуры |
| function CheckGardenUsing | Проверка на использование в садовом участке определённой грядки | (const gardencod: integer);  Gardencod – индекс грядки |
| procedure DeleteCulture | Удаление из списка культур определённой записи | (const name: string; list: Ptculture);  Name – имя культуры для удалния  list – динамический список культур |
| procedure DeleteGarden | Удаление из списка грядок определённой записи | (const name: string; list: ptgarden);  Name – имягрядки для удалния  list – динамический список грядок |
| procedure AddCulture | Добавление в список культур записи | (const newCulture: cultureListInfo; list: Ptculture);  newCulture – имя культуры для добавления  list – динамический список культур |
| procedure AddGarden | Добавление в список грядок записи | (const newgarden: TGarden; list: ptgarden);  newgarden – имя культуры для добавления  list – динамический список грядок |
| function IsValidGarden | Проверка на валидность введённого имени грядки | (list: ptgarden; const str: string; var cod: integer);  list – динамический список грядок  str – имя грядки  cod – индекс грядки в списке грядок |
| function IsValidCulture | Проверка на валидность введённого названия культуры | (list: Ptculture; const str: string; var cod: integer);  list – динамический список культур  str – имя культуры  cod – индекс культуры в списке культур |
|  |  |  |
| function IdentifyColor | Определение цвета грядки по её коду | (const cod: integer)  cod – индекс шрядки в списке грядок |
| function ConvertStringToDate | Конвертация из строки в дата-тип | (const dateStr: string; var date: TMyDate);  dateStr – строка хранящая дату  date – дата созданная из строки |
| function ConvertDateToString | Конвертация из дата-типа в строку | (const dateStr: TMyDate)  dateStr – дата которая будет конвертированна в строку |
| function GetIdGarden | Возврат индекса грядки по её имени | (const Name: string; list: PtGarden);  Name – имя грядки  list – динамический список грядок |

**5** **ТЕСТИРОВАНИЕ, ПРОВЕРКА РАБОТОСПОСОБНОСТИ И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ**

**5.1 Тестирование программного средства**

Таблица 5.1 – Тестирование программного средства

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специфика тестирования | Номер теста | Вводимые данные | Ожидаемый результат | Полученный результат |
| Запуск программы | 1 | - | Отображение садового участка, все кнопки из меню должны иметь иконку | Тест пройден |
| Отображения садового участка | 2 | Нажатие кнопок увеличения/уменьшая масштаба | Увеличение / уменьшение клеток каждого элемента садового участка | Тест пройден |
| Добавление записи культуры | 3 | Нажатие кнопки добавления культуры, ввод данных и добавление | Открытие формы добавления культуры, при условии корректности введённых данных, изменение структуры данных культур | Тест пройден |
| Добавление записи грядки | 4 | Нажатие кнопки добавления грядки, ввод данных и добавление | Открытие формы добавления грядки, при условии корректности введённых данных, изменение структуры данных грядки | Тест пройден |
| Удаление записи культуры | 5 | Нажатие кнопки удаления культуры | Открытие формы удаления культуры, изменение структуры данных культур | Тест пройден |
| Удаление записи грядки | 6 | Нажатие кнопки удаления грядки | Открытие формы удаления грядки, изменение структуры данных грядки | Тест пройден |
| Выделение элемента сада | 7 | Движение курсора по области рисования | Выделение только одного элемента сада | Тест пройден |
| Выделение элементов сада | 8 | Движение курсора по области рисования с зажатой ЛКМ | Выделение в области прямоугольника сразу нескольких элементов сада | Тест пройден |
| Подготовка данных для редактирования грядок | 9 | Выделение элементов сада | Отображение формы со всеми списками культур и грядок | Тест пройден |
| Редактирование элементов сада (грядки) | 10 | Изменение цвета грядки | Цвет меняется для всех выделенных элементов сада | Тест пройден |
| Редактирование элементов сада (культур) | 11 | Изменение культуры | Культура изменяется для всех выделенных элементов сада | Тест пройден |
| Просмотр сада | 12 | Нажатие пункта открытия просмотра сада | Открытие формы с уже отрисованным дачным участком | Тест пройден |
| Отображение грядки | 13 | Нажатие определённой грядки в списке или в области отрисовки | Отображение только выделенной грядки | Тест пройден |
| Отображение культуры | 14 | Нажатие определённой культуры в списке | Выделение определённым образом только элементов с нужной культурой | Тест пройден |
| Счёт количества культур | 15 | Выбор определённой грядки | Отображение всех культур на всех элементах грядки и подсчёт их количества |  |
| Выход из программы | 16 | Нажатие кнопки “крестик” | Завершение программы, закрытие формы | Тест пройден |
| Сохранение изменений | 17 | Нажатие определённых клавиш или кнопки сохранения | Сохранение всех данных | Тест пройден |
| Загрузка данных после их сохранения | 18 | Запуск программы после сохранения | Отображения сада в соответствии с последним сохранением | Тест пройден |

**5.2 Анализ полученных результатов**

Ошибок при тестировании не найдено. Программное средство работает корректно. Все формы отображают верные данные, в нужном порядке. Все события, создаваемые пользователем работают правильно. Все результаты взаимодействия пользователя происходят соответствуя ожиданию.

Структура данных культур и грядок по ходу выполнения программы хранят корректные данные. Все проверки на адекватность введённых данных выполняют свои функции.

Все формы программы появляются при ожидаемой необходимости и отображают компоненты в нужном месте и отображают верные данные. Время отклика программы не превышает критических значений.

**6 РУКОВОДСТВО ПО УСТАНОВКЕ И ЭКСПЛУАТАЦИИ**

1. Установите exe-файл программы в одной папке с типизированными файлами данных расширения \*.cul.

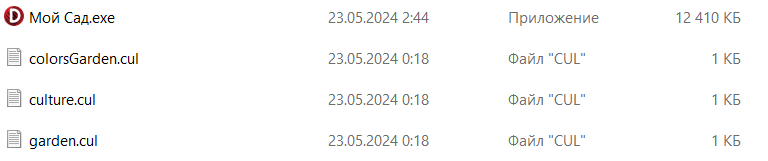


Рис. 6.1 – Пример расположения некоторых необходимых для работы файлов.

1. Основная форма программы

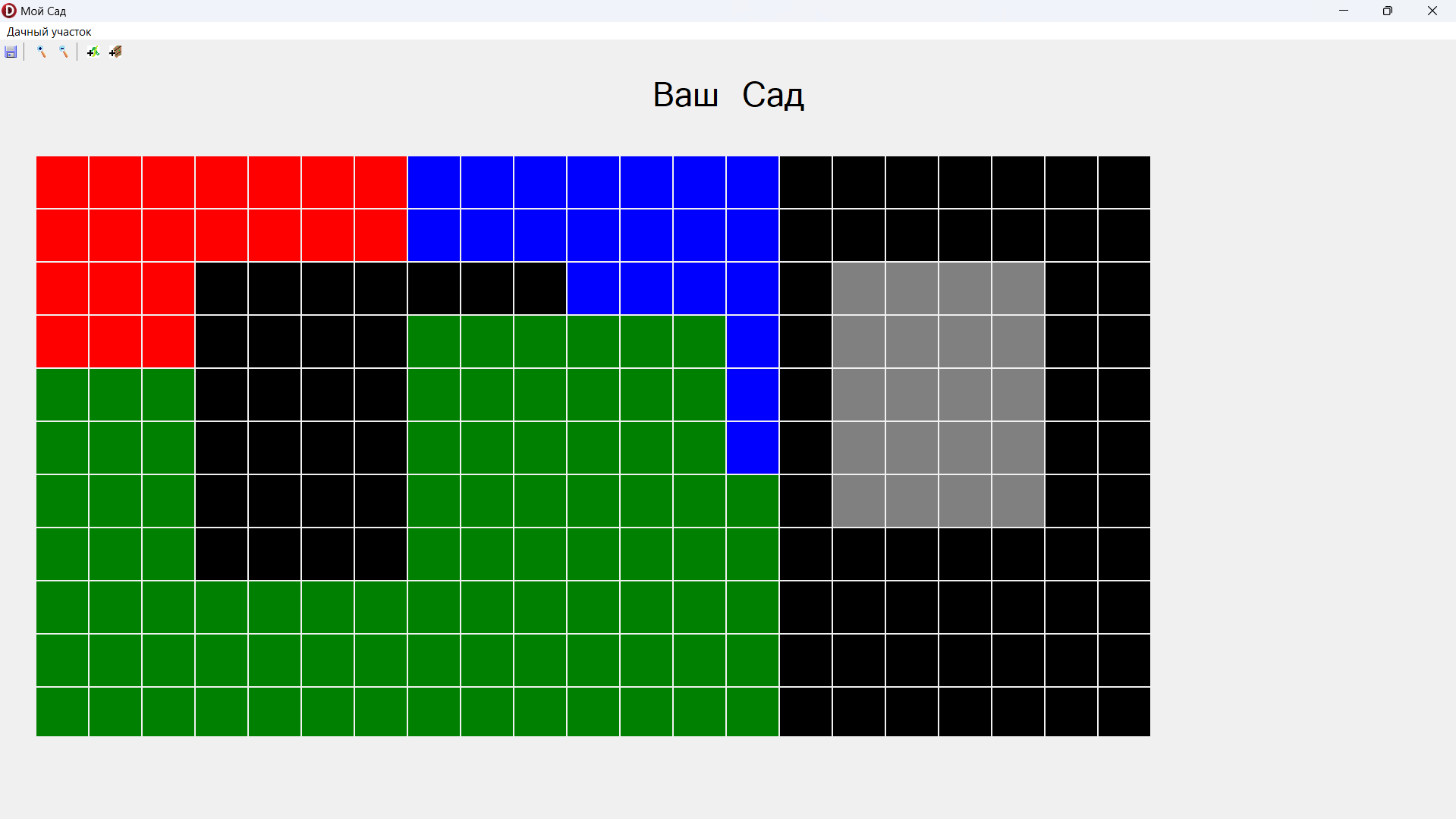


Рис. 6.2 – Основное меню программы.

1. Выделение нескольких элементов сада

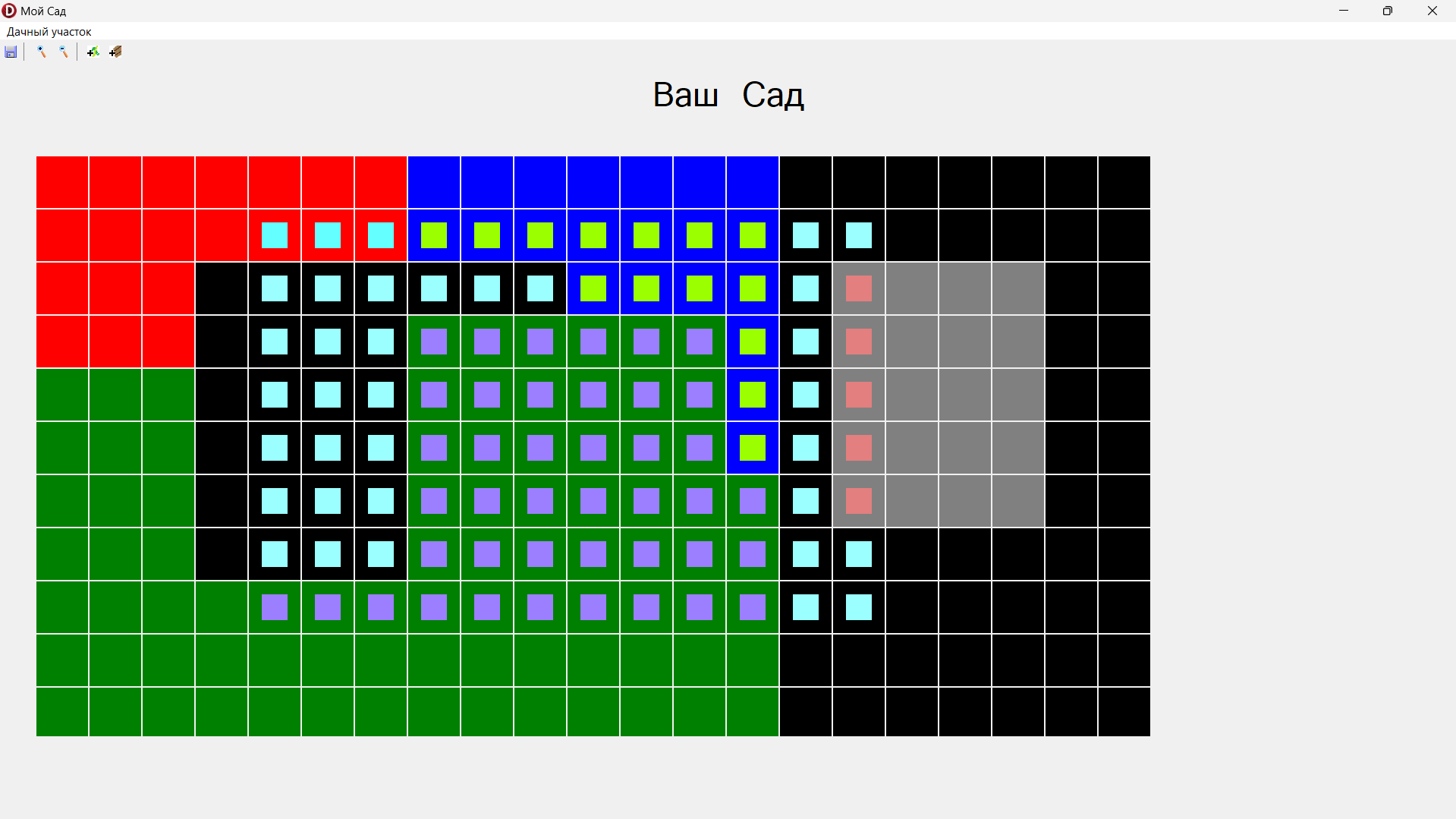


Рис. 6.3 – Выделение грядок для редактирования.

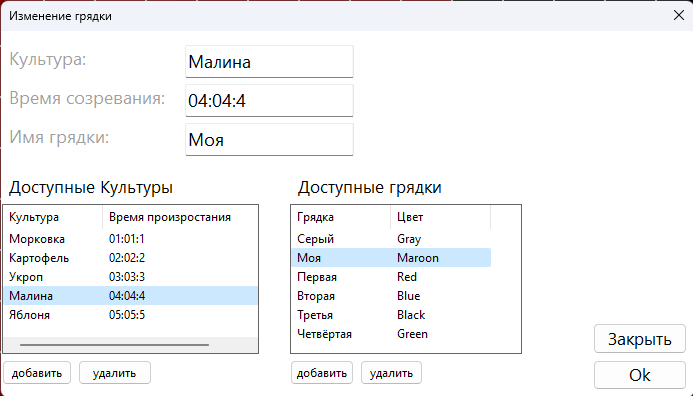
1. Редактирование грядок при выборе определённых элементов сада 

Рис. 6.4 – Окно редактирования грядок.

1. При нажатии кнопки “Ок” выбранные изменения отобразятся на форме

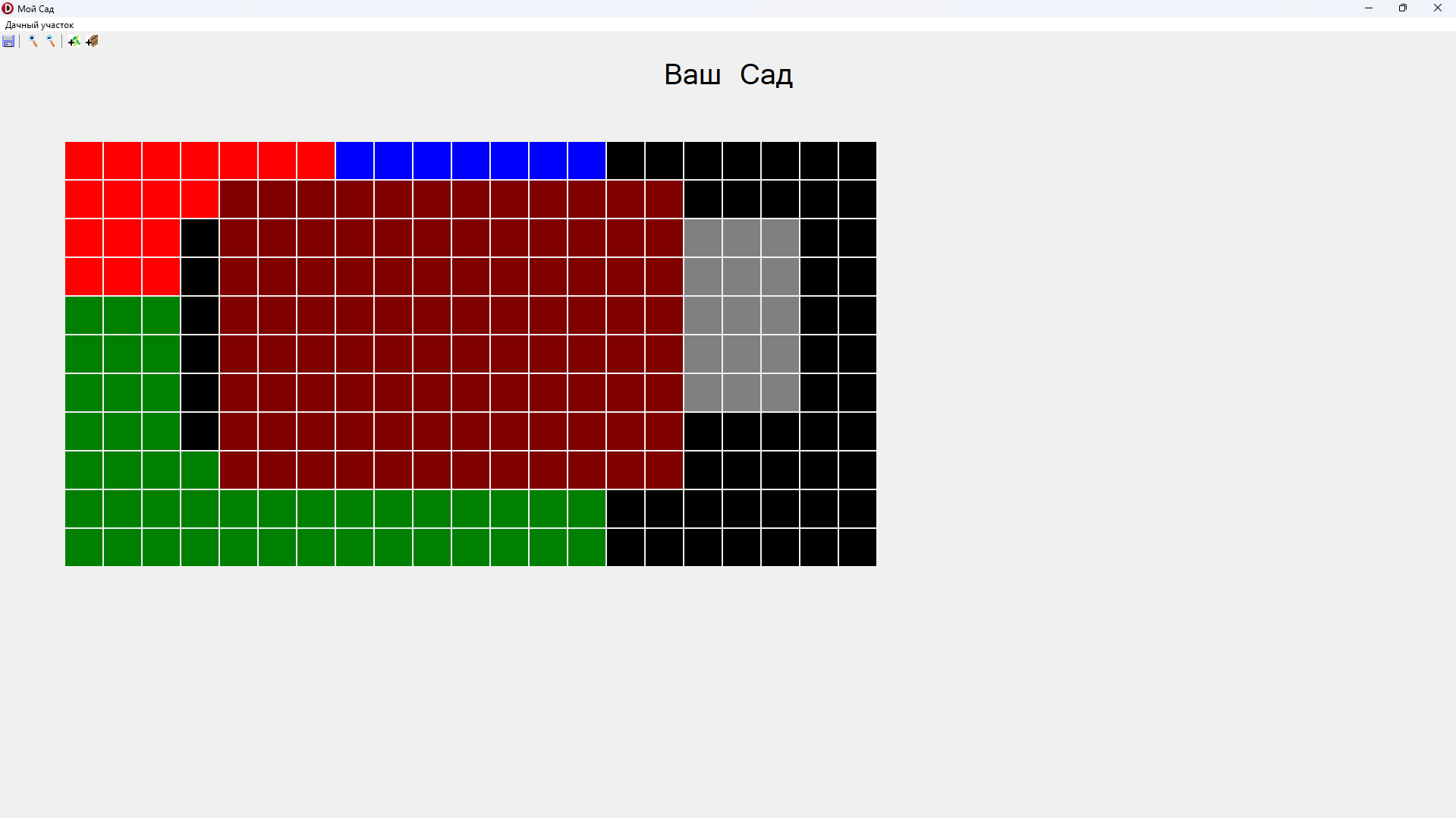


Рис. 6.5 – Основное меню программы.

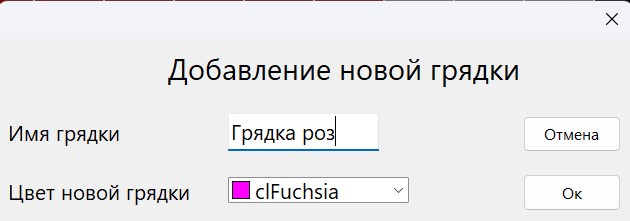
1. При нажатии кнопки добавления грядки откроется окно добавления новой грядки

Рис. 6.6 – Форма добавления новой грядки

1. При попытке ввести недопустимые значения будет получено уведомление

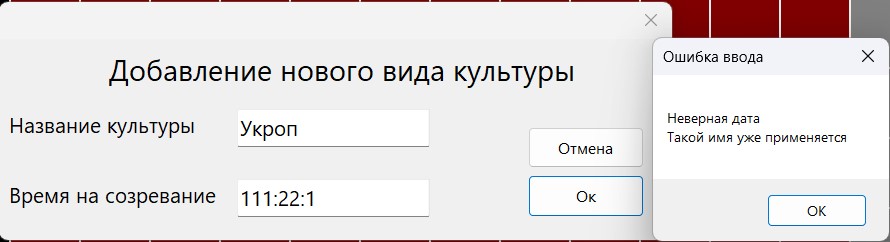


Рис. 6.7 – Информирование о некорректности введённых данных

1. При нажатии кнопки сохранения или комбинации клавиш Ctrl + S произойдёт сохранение всех данных

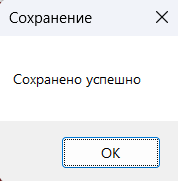


Рис. 6.8 – Информирование о сохранении данных

1. При нажатии раздела меню “Дачный участок”

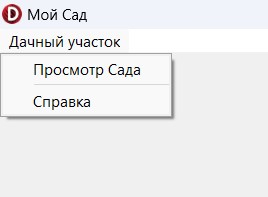


Рис. 6.9 – Разделы меню “Дачный участок”

1. В программе при выборе подменю “Справа” отобразиться небольшая справка

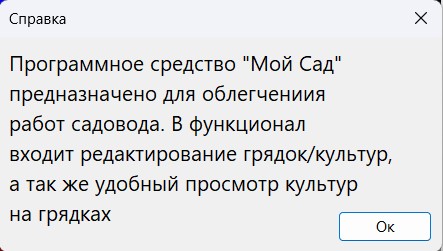


Рис. 6.10 – Справка о программном средстве.

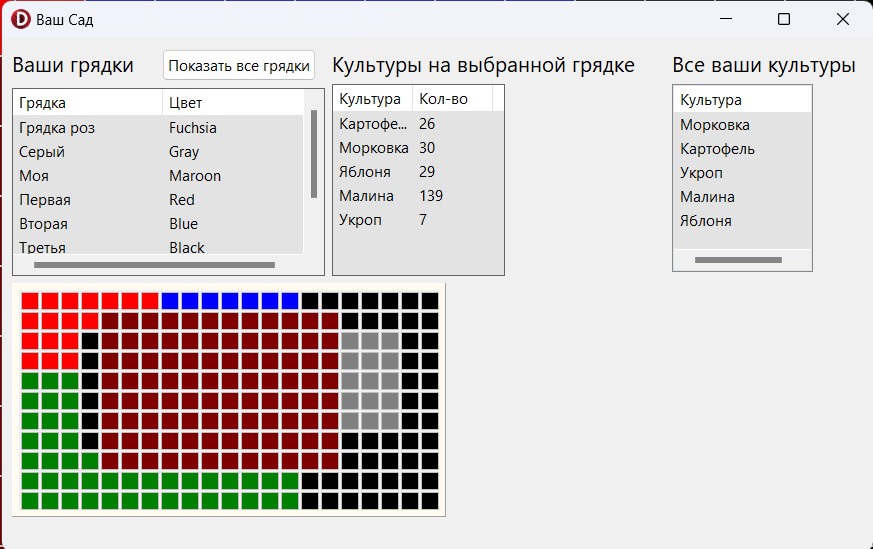
1. Функционал программы включает в себя также просмотр садового участка 

Рис. 6.11 – Просмотр дачного хозяйства.

1. Просмотр конкретной грядки

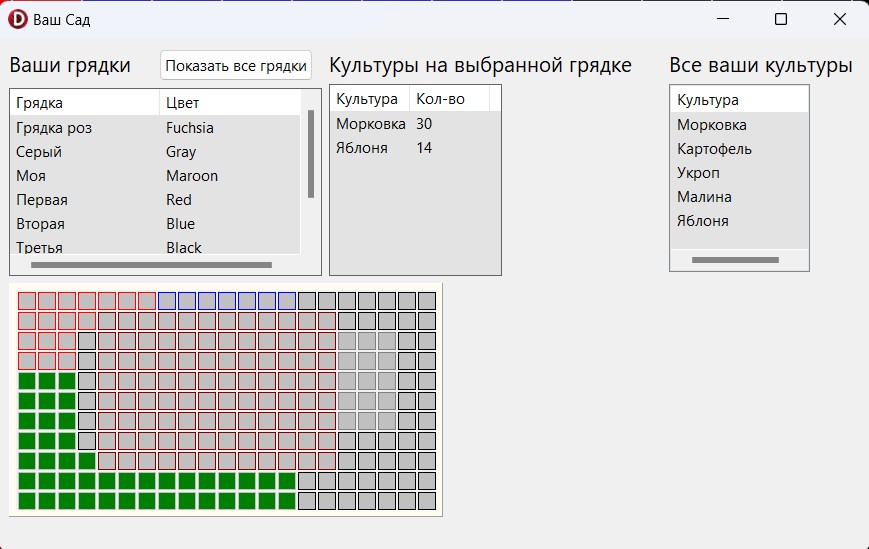


Рис. 6.12 – Отображение выбранной грядки.

1. Просмотр конкретной культуры

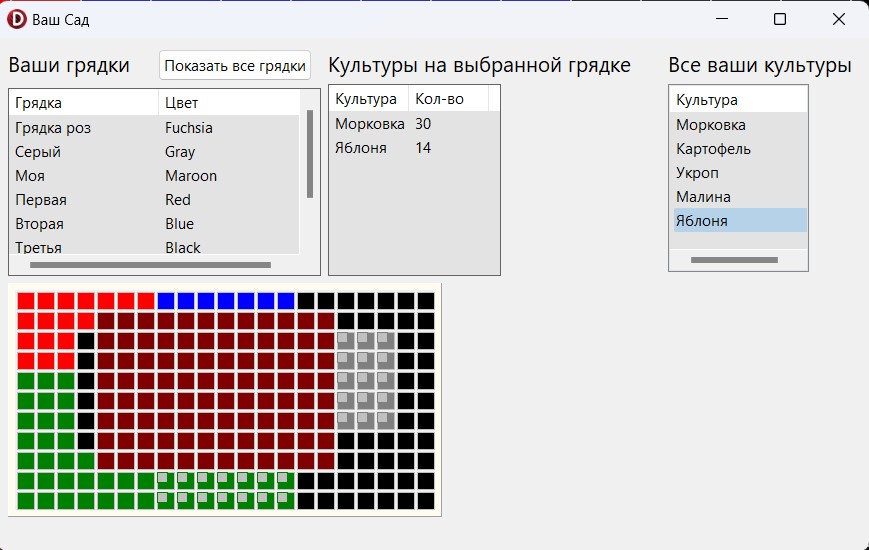


Рис. 6.9 – Отображение выбранной культуры.

1. При необходимости выйти из программы нужно нажать крестик

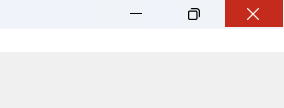


Рис. 6.10 – Закрытие программы.

**ЗАКЛЮЧЕНИЕ**

В рамках данной курсовой работы было разработано программное средство для учёта работ садовода. В ходе выполнения проекта были решены следующие основные задачи:

1. Проведён анализ предметной области и выявлены ключевые требования к функционалу программного средства учёта работ садовода.
2. Разработана структура данных для хранения информации о садовых культурах и грядках
3. Разработанное программное средство позволяет удобно и эффективно взаимодействовать с садовым участком.
4. Проведено тестирование разработанного программного средства, что позволило убедиться в корректности его работы.

Разработанное программное средство обладает высокой практической ценностью для садоводов-любителей и профессионалов. Его использование позволяет повысить производитель-ность труда, оптимизировать расходы и улучшить качество ведения садового хозяйства. Кроме того, данное решение может быть в дальнейшем расширено дополнительными модулями, такими как интеграция с метеоданными, прогнозирование урожайности, управление теплицами и т.п.

**ПРИЛОЖЕНИЕ А**

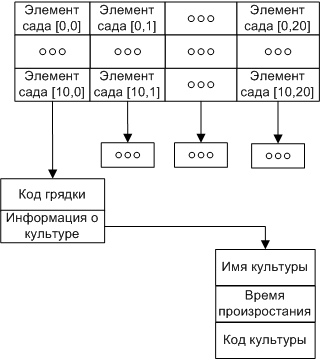


Рис. А1 – Представление данных в массиве сада

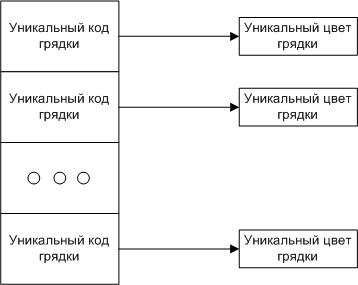


Рис. А2 – Представление данных структуре данных для хранения цвета каждой грядки

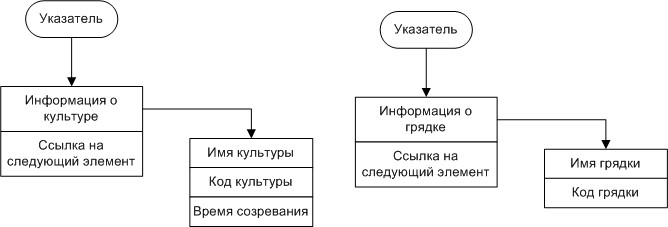


Рис. А3 – Представление данных в структурах данных всех грядок и культур

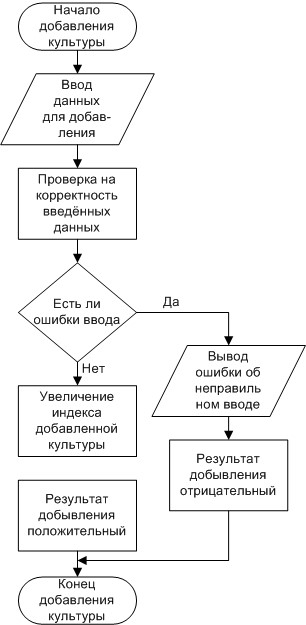


Рис. А4 – Алгоритм добавления новой культуры



Рис. А5 – Алгоритм отрисовка садового участка



Рис. А6 – Алгоритм подпрограммы по отрисовки квадрата(элемента сада и др)

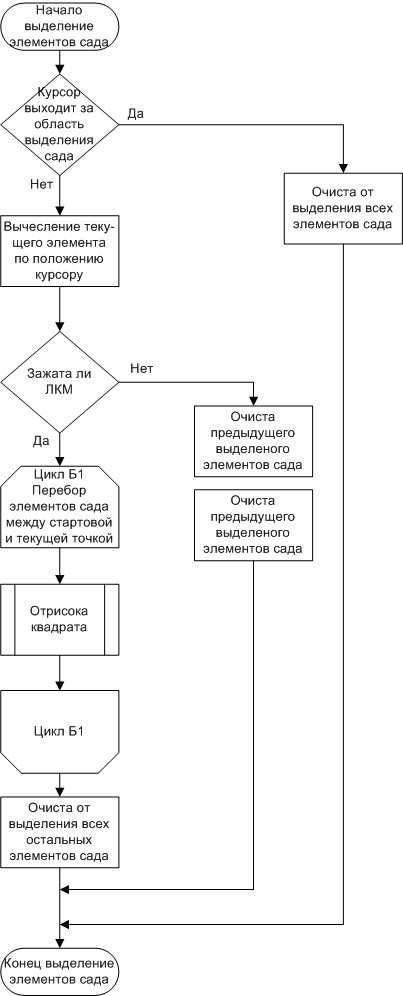


Рис. А7 – Алгоритм выделения элементов сада для изменения

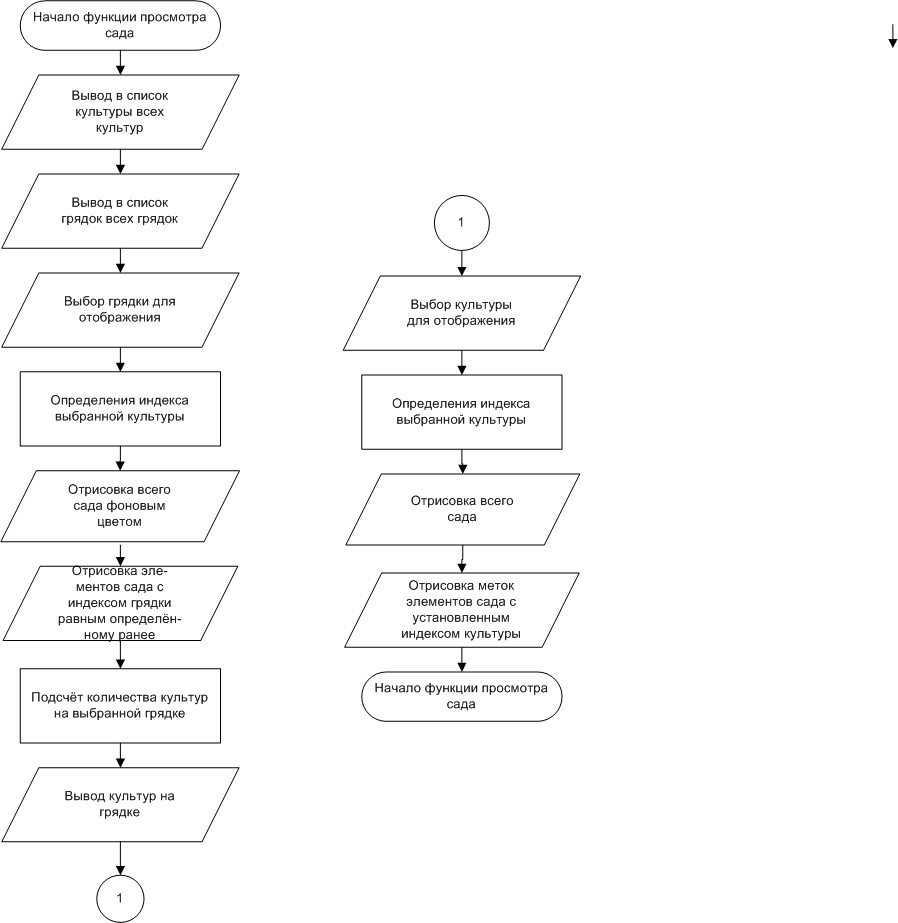


Рис. А8 – Алгоритм просмотра всего садового участка

**ПРИЛОЖЕНИЕ В**