

## Задачи на лабораторные работы.

### 1. Операции над матрицами.

Цель: написать программу, позволяющую производить операции над матрицами для частного случая целых чисел.

Задачи:

- 1.1 Создать контейнер для хранения двумерного массива чисел типа int.
- 1.2 Используя созданный контейнер реализовать следующие операции с матрицами:
  - умножение матрицы на число;
  - умножение матриц;
  - сложение матриц;
  - вычитание;
  - транспонирование.
- 1.3 Создать пользовательский интерфейс, который будет позволять редактировать матрицы задавать операции над ними и отображать результат выполнения<sup>1</sup>. Сделать помощник (wizard) который позволит сохранять результаты (полностью/частично) в выбранный файл, выбирать подпись, ставить дату.

<sup>1</sup> Для случая  $[\lambda * M_{4 \times 4}^{<x>} \langle x \rangle \lambda * M_{4 \times 4}^{<x>} \langle x \rangle \lambda * M_{4 \times 4}^{<x>}]$ , где  $\langle x \rangle$  - выбираемая операция  $\langle + \rangle$ ,  $\langle - \rangle$  или  $\langle * \rangle$ , а  $\langle t \rangle$  - опция, позволяющая задать операцию транспонирования,.

### 2. (\*) Электронная таблица.

Цель: написать программу, реализующую функционал электронной таблицы для частного случая 2-х типов ячеек (текстовые и цифровые данные).

Задачи:

- 2.1 Разработать контейнер для ячейки электронной таблицы, поддерживающий 2 типа ячеек, предназначенных для хранения текстовой и цифровой информации.
- 2.2 Разработать двумерный контейнер для хранения контейнеров ячеек таблицы.
- 2.3 Используя созданные контейнеры реализовать следующие операции:
  - сложение/вычитание;
  - конкатенация;
  - перенос ячейки;
  - копирование ячейки;
  - конвертация ячеек разных типов.
- 2.4 Создать пользовательский интерфейс, который будет позволять для таблицы размером 10x10 добавлять/удалять ячейки таблицы, редактировать ячейки таблицы и осуществлять операции из п. 2.3. Для конвертации ячеек сделать помощник (wizard).

### 3. Расширение разрядности.

Цель: написать программу, реализующую контейнер с расширяемой разрядностью.

Задачи:

- 3.1 Разработать контейнер для хранения разрядного сегмента на базе типа `uint8_t`.
- 3.2 Реализовать на базе контейнера разряда контейнер с расширяемой разрядностью, поддерживающий следующие операции:
  - сложение;
  - вычитание;
  - умножение.
- 3.3 Создать пользовательский интерфейс который будет позволять задавать значения и тип операций для 5-сегментных разрядных контейнеров. Создать помощник (wizard) для настройки параметров выполнения операций.

### 4. Символьные вычисления.

Цель: написать программу, реализующие базовые математические операции для чисел представленных в виде строк.

Задачи:

- 4.1 Разработать контейнер для хранения числа на базе типа `char`.
- 4.2 Реализовать на базе контейнера для хранения числа следующие операции:
  - вычитание;
  - сложение;
  - умножение;
  - деление.
- 4.3 Создать пользовательский интерфейс, позволяющий вводить числа и задавать операции между ними, а также отображающий результат операций. Создать помощник (wizard) для настройки параметров выполнения операций.

### 5. Редактирование текста.

Цель: написать программу, реализующую работу с изменениями на примере редактора текстов.

Задачи:

- 5.1 Разработать контейнер(ы) для хранения перечня(-ей) изменений.
- 5.2 С использованием разработанных контейнеров реализовать следующие операции:
  - отмена произведённых операций;
  - возвращение отменённых операций.
- 5.3 Создать пользовательский интерфейс, реализующий отображение вводимого текста, а также контейнеров отмены/возвращения. Создать помощник (wizard) для настройки параметров отмены/возвращения.

## 6. Форматирование текста.

Цель: написать программу, реализующую работу с форматированием текста на примере редактора текстов.

Задачи:

6.1 Разработать контейнер для хранения текста.

6.2 Используя разработанный контейнер реализовать следующие операции:

- выравнивание по ширине;
- выравнивание по центру;
- выравнивание по правому краю;
- выравнивание по левому краю.

Форматирование реализовывать с учётом отступа абзаца.

6.3 Создать пользовательский интерфейс, реализующий отображение форматированного и неформатированного текста и элементов управления. Создать помощник (wizard) для настройки параметров форматирования.

## 7. Телефонно-адресная книга.

Цель: написать программу, реализующую функционал телефонно-адресной книги.

Задачи:

7.1 Разработать структуру данных для хранения информации и контейнер для хранения структур данных (контейнер реализовывать с использованием пары массивов, которые и будут определять ассоциации).

7.2 На основе контейнера реализовать следующие операции:

- добавление абонента;
- удаление абонента;
- изменение информации об абоненте;
- запрос информации об абоненте.

7.3 Создать пользовательский, который позволял бы осуществлять поиск, отображение и редактирование данных об абоненте. Создать помощники (wizard) для добавления/удаления абонента.

## 8. Канал обмена сообщений.

Цель: написать программу, реализующую канал обмена сообщений по схеме «публикация-подписка».

Задачи:

8.1 Разработать контейнер для хранения очереди сообщений.

8.2 Используя разработанный контейнер построить маршрутизатор сообщений.

8.3 Создать пользовательский интерфейс, отображающий доступные для подписки каналы, форму для отправки сообщений и входящие сообщения. Создать помощники (wizard) для оформления подписки/отписки.

## 9. Управление состоянием датчиков.

Цель: разработать программу, реализующую компактное хранение небольших переменных в памяти для частного случая 2-битных переменных.

Задачи:

9.1 Разработать контейнер для хранения переменных.

9.2 На основе разработанного контейнера построить пульт управления датчиками, реализующий следующий функционал:

- добавление нового датчика;
- удаление датчика;
- проверка состояния датчика.

9.3 Создать пользовательский интерфейс для отображения состояния датчиков для случая 10 датчиков. Создать помощники (wizard) для подключения/отключения датчиков.

## 10. (\*) Менеджер памяти. Выделение памяти (выделение областей памяти, контроль оставшегося свободного места, оставшегося пространства, дефрагментация).

Цель: разработать программу, реализующую управление «памятью»<sup>2</sup>.

Задачи:

10.1 Разработать менеджер памяти, реализующий следующий функционал:

- выделение памяти;
- освобождение памяти;
- поиск свободного участка памяти;
- контроль незанятой памяти;
- дефрагментацию памяти.

10.2 Создать пользовательский интерфейс, который бы отображал состояние памяти в текущий момент, позволял бы производить операции с памятью (п. 10.1).

<sup>2</sup> Во избежание неоправданного усложнения программы, в данной задаче «память» представим как массив элементов типа `unsigned char`.

## 11. Балансировка нагрузки.

Цель: разработать программу, реализующую распределение нагрузки методом перебора по круговому циклу.

Задачи:

11.1 Разработать контейнер для очереди заданий (учитывая приоритет задания).

11.2 Используя разработанный контейнер как ввод построить алгоритм распределения нагрузки (учитывая время выполнения задания).

11.3 Создать пользовательский интерфейс, отображающий работу алгоритма балансировки. Создать помощник (wizard) для настройки параметров работы при запуске алгоритма.

## 12. Двухнаправленный ассоциативный контейнер.

Цель: разработать программу, реализующую функционал телефонной книги.

Задачи:

12.1 Разработать структуру данных для хранения информации и контейнер для хранения структур данных.

12.2 На основе контейнера реализовать следующие операции:

- добавление абонента;
- удаление абонента;
- изменение информации об абоненте;
- поиск телефонного номера по имени абонента;
- поиск имени абонента по телефонному номеру.

12.3 Создать пользовательский интерфейс, позволяющий осуществлять поиск, отображение и редактирование данных об абоненте. Создать помощники (wizard) для добавления/удаления абонента.

## 13. Канонический степенной многочлен. Дифференцирование, сложение, вычитание.

Цель: написать программу, реализующие базовые математические операции для многочленов вида  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ .

Задачи:

13.1 Разработать контейнер для хранения многочлена.

13.2 Используя разработанный контейнер реализовать следующие операторы:

- сложение многочленов;
- вычитание многочленов;
- дифференцирование многочленов.

13.3 Создать пользовательский интерфейс, позволяющий отображать многочлены, задавать их коэффициенты<sup>3</sup> и определять операторы для случая  $n=6$ .

<sup>3</sup> Коэффициенты могут задаваться как буквами так и цифрами.

## 14. Транзитные перелёты (нахождение кратчайших путей между вершинами ориентированного графа).

Цель: написать программу, реализующую нахождение кратчайшего маршрута перелёта с учётом транзитных маршрутов (для карты маршрутов полётов использовать ориентированный граф).

14.1 Создать контейнер для реализации графа.

14.2 Используя разработанный контейнер разработать алгоритм нахождения минимального маршрута.

14.3 Создавать пользовательский интерфейс, отображающий карту маршрутов (вид карты согласуется отдельно), позволяющий задавать весовые коэффициенты и дальности путей и отображающий выбранный маршрут.

15. Кратчайший маршрут (нахождение кратчайших путей между вершинами неориентированного графа).

Цель: написать программу, реализующую нахождение кратчайшего маршрута с учётом карты перекрёстков и светофоров (для карты использовать неориентированный граф).

14.1 Создать контейнер для реализации неориентированного графа.

14.2 Используя разработанный контейнер разработать алгоритм нахождения минимального маршрута.

14.3 Создавать пользовательский интерфейс, отображающий карту маршрутов (вид карты согласуется отдельно), позволяющий задавать весовые коэффициенты и дальности путей и отображающий выбранный маршрут.

16. (\*) Сравнение файлов.

Цель: написать программу, реализующую сравнение файлов.

Задачи:

16.1 Разработать программу сравнения файлов, реализующую сравнение файлов с учётом добавленных и удалённых фрагментов.

16.2 Создать пользовательский интерфейс для отображения результатов сравнения.

17. (\*) Триангуляция многоугольника.

Цель: написать программу, реализующую алгоритм триангуляции многоугольника.

Задачи:

17.1 Разработать контейнер для реализации многоугольника.

17.2 Разработать алгоритм для триангуляции (выбрать такую совокупность хорд, что никакие две хорды не будут пересекаться, а весь многоугольник будет поделен на треугольники). Общая длина хорд должна быть минимальной.

17.3 Создать пользовательский интерфейс для задания параметров многоугольника, отображения его и его разбиения на треугольники.

18. (\*) Копирование при записи (copy on write).

Цель: написать программу, реализующую контейнер создающий копию своих данных только при обращении на запись.

Задачи:

18.1 Разработать контейнер для данных, реализующий стратегию copy on write для контейнера целиком и для части данных.

18.2 Создать пользовательский интерфейс, позволяющий изменять содержимое контейнеров и отображающий результаты работы алгоритма.

## 19. Предикативная система набора текста T9.

Цель: написать программу, реализующую предикативную систему набора текста.

Задачи:

19.1 Создать контейнер для хранения словаря.

19.2 Используя контейнер реализовать алгоритм T9.

19.3 Создать пользовательский интерфейс для реализации пользовательского ввода и отображения результатов работы алгоритма.

## 20. Определение частоты встречаемости пар слов в тексте.

Цель: написать программу, реализующую подсчёт частоты встречаемости пар слов в тексте.

Задачи:

20.1 Создать контейнер для хранения пар слов.

20.2 Используя разработанный контейнер написать алгоритм подсчёта встречаемости пар слов в тексте (при разработке алгоритма учесть факт что подсчёт частотности встречаемости слов может проводиться не только для соседних слов, но и через слово, через 2 слова и т.д.).

20.3 Создать пользовательский интерфейс для отображения результатов. Создать диалоги для открытия файла с текстом и сохранения результата.

## 21. Замена сокращений.

Цель: написать программу, реализующую поиск сокращённых слов и замену их полными.

Задачи:

21.1 Разработать контейнер для хранения сокращений.

21.2 Разработать алгоритм поиска и замены сокращений в тексте.

21.4 Создать пользовательский интерфейс позволяющий производить поиск и замену сокращений в режимах «заменить всё» и «найти и заменить». Создать диалоги для открытия файла с текстом и сохранения результата.

## 22. Нахождение ошибок в словах.

Цель: написать программу, реализующую проверку слов на соответствие словам из состава словаря, введённого в программу.

Задачи:

22.1 Создать контейнер для реализации словаря.

22.2 Реализовать алгоритм для проверки правописания слов (для определения меры схожести слов использовать дистанцию Хэмминга).

22.3 Создать пользовательский интерфейс позволяющий производить проверку правописания слов в режимах «проверить все» и «проверить и перейти к следующему». Создать диалоги для открытия файла с текстом и сохранения результата.