Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №1**по дисциплине «Структуры и организация данных»  
на тему «Линейные структуры данных»  
  
вариант *номер\_варианта*

Выполнил:  
Студент *Фамилия И.О.*  
Группа *Номер\_группы*  
  
Преподаватель:  
*Фамилия И.О.*

Санкт-Петербург  
2022 г.

*Оставляем только абзац, соответствующий выбранному уровню сложности.*

Уровень сложности – **низкий**. Данные хранятся в бинарном файле записей, а для обработки считываются в односвязный линейный список (если файл не существует, то создается пустой список). При выходе из программы обработанные данные сохраняются в том же файле. Имя файла с данными задается константой. Для организации интерфейса должно использоваться меню.Обязательные операции для списка: добавление элемента в начало списка, удаление первого элемента списка, просмотр списка, поиск в соответствии с индивидуальным вариантом. Вывод списка на экран можно выполнять в любом (главное, читабельном) виде.

Уровень сложности – **средний**. Данные хранятся в бинарном файле записей, а для обработки считываются в односвязный линейный список (если файл не существует, то создается пустой список). При выходе из программы обработанные данные сохраняются в том же файле. Имя файла с данными должно передаваться программе при ее запуске (через параметры функции main()). Если параметры пользователем при запуске программы не заданы, имя файла задается константой. Элементами данных должны являться записи сложной структуры (дополнительные поля придумать самостоятельно)**.** Обязательные операции для списка: добавление элемента в конец списка, просмотр списка, удаление последнего элемента списка, поиск в соответствии с индивидуальным вариантом. Вывод данных осуществлять в табличном виде с графлением визуально подходящими символами.

Уровень сложности – **повышенный**. данные хранятся в бинарном файле записей, а для обработки считываются в двусвязный линейный список (если файл не существует, то создается пустой список). При выходе из программы обработанные данные сохраняются в том же файле. Имя файла с данными должно передаваться программе при ее запуске (через параметры функции main()). Если параметры пользователем при запуске программы не заданы, имя файла вводится с клавиатуры. Элементами данных должны являться записи с вариантами (вариативные поля придумать самостоятельно)**.** Обязательные операции: добавление элемента в упорядоченный список с сохранением упорядоченности (ключевое поле выбрать самостоятельно), просмотр списка в прямом и обратном направлении, удаление произвольного элемента списка, поиск в соответствии с индивидуальным вариантом. Вывод данных осуществлять постранично в табличном виде с графлением визуально подходящими символами, предусмотреть листание в прямом и обратном направлении.

*Текст вариативной части задания*

Созданные типы данных:

*назначение типа (для чего он создан), объявление типа, назначение полей*

Используемые функции:

*назначение функции, заголовок функции, входные и выходные данные*.

Текст программы:

*Сюда добавляем текст программы* ***с комментариями****. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Результаты работы программы:

*Сюда вставляем скриншоты, позволяющие увидеть работу каждой функции. Рисунки обрезаем, оставляя только информативную часть. Каждый рисунок должен сопровождать пояснительный текст.*

1. Структура данных – *дек с ограниченным выходом (копируем из вариативной части задания).*

*Оставляем только абзацы, соответствующие выбранному уровню сложности.*

Уровень сложности – **низкий**. написать программу тестирования структуры данных, указанной в вариативной части задания. Файл с векторной реализацией соответствующей структуры данных взять из папки с примерами. Программа тестирования должна включать в себя следующие инструкции:

* добавление элементов в структуру с фиксацией результата (добавление произошло или нет). Если структура данных – дек, добавление элементов должно осуществляться с обоих открытых концов;
* извлечение элемента из структуры с фиксацией результата (извлекли или нет). Если структура данных – дек, извлечение элементов должно осуществляться с обоих открытых концов;
* проверка пустоты структуры;
* проверка заполненности структуры.

Уровень сложности – **средний**. Реализовать необходимую структуру данных с помощью одной структуры хранения (векторной или связной). Реализацию оформить в виде класса. Использовать созданный тип данных для решения поставленной задачи.

Уровень сложности – **повышенный**. Реализовать необходимую для решения задачи структуру данных с помощью двух структур хранения: векторной и связной,– реализацию оформить в виде иерархии классов. Абстрактный базовый класс должен задавать интерфейс, а производные – реализацию структуры данных. В программе, решающей поставленную задачу, выбор структуры хранения предоставить пользователю.

*Текст вариативной части задания (только для среднего и повышенного уровня сложности).*

Схематичное изображение структуры данных:

*рисунок*

Схематичное изображение структуры хранения, использованной в программе:

*рисунки структуры хранения в трех состояниях: пуста, заполнена частично, заполнена*

Текст программы:

*Сюда добавляем текст программы* ***с комментариями****. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Результаты работы программы:

*Скриншоты с пояснительным текстом*