

Курсовая работа

на тему

«Динамические структуры данных. Название вашего варианта»

Разработать программу в соответствии с вариантом.

Использовать односвязные и/или двусвязные линейные списки (забыть про массивы!)

Все элементы обработки данных должны быть оформлены в виде процедур.

Лабораторную выполнять в PascalABC (PascalABC.NET) или в Lazarus.

Данные считываются из файла (файл создать в этой же программе) и выводятся на экран или в файл.

Текст курсовой работы должен содержать:

1. введение
2. постановку задачи
3. описание используемых типов данных (с обоснованием)
4. подробное описание всех процедур и функций
5. полный текст программы.
6. скриншоты результатов работы программы
7. заключение

При необходимости студент имеет право добавить дополнительные поля с данными.

Придумать такие задачи к своему варианту (и согласовать с преподавателем), чтобы были задействованы все (или почти все) процедуры и (или) функции, перечисленные ниже, а также имелись сложные запросы:

- создание пустого списка в соответствии с вариантом;
- проверка списка на пустоту;
- заполнение списка;
- печать элементов списка в порядке их расположения;

- определение длины списка;
- вставка нового элемента;
- удаление элемента в соответствии с условием;
- разбиение списка на два списка по какому-либо условию;
- соединение двух линейных списков в один список;
- переворот списка;
- удаление всех элементов списка;
- поиск максимального элемента списка;
- поиск минимального элемента;
- поиск элемента с заданным значением x
- изменение значения элемента списка с заданным значением.
- Сортировка по какому-либо параметру.

Задания по вариантам

Вариант 1. Авиакасса

Разработать программу, которая содержит текущую информацию о заявках на авиабилеты.

Каждая заявка содержит:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.

Вариант 2. Характеристики ноутбуков

Разработать программу, которая содержит сведения о ноутбуках.

Структура каждой записи:

- производитель;
- диагональ;
- процессор;

- тактовая частота;
- объем ОП в Гб;
- объем HDD или SSD в Гб;
- вес;
- цена.

Вариант 3. Сотрудники предприятия

Разработать программу, которая содержит динамическую информацию о сотрудниках предприятия.

Сведения о каждом сотруднике содержат:

- фамилию и инициалы сотрудника;
- название занимаемой должности;
- год поступления на работу.

Вариант 4. Персональные данные

Разработать программу, которая содержит динамическую информацию о личных данных.

Сведения о каждом человеке содержат:

- фамилию
- имя;
- номер телефона;
- адрес
- дата рождения (день, месяц, год).

Вариант 5. Характеристики ПК

Разработать программу, которая содержит сведения о персональных компьютерах.

Структура каждой записи:

- марка процессора
- тактовая частота
- объем памяти в Гб

- объем SSD в Гб
- цена

Вариант 6. База данных спортсменов

Разработать программу, которая содержит сведения о спортсменах.

Структура каждой записи:

- фамилия спортсмена;
- пол;
- возраст,
- город;
- вид спорта,
- результат (занятое место).

Вариант 7. Студенческая группа

Разработать программу, которая содержит сведения студентах группы

Структура записи:

- фамилия
- имя
- отчество студента;
- размер стипендии;
- место жительства (общежитие, квартира);
- семейное положение.

Вариант 8. Магазин

Разработать программу, которая содержит динамическую информацию о товарах.

Сведения о каждом товаре содержат:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях,
- количество.

Вариант 9. Железнодорожный вокзал

Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается:

- номер поезда;
- станция назначения;
- время отправления;

Вариант 10. Каталог видеофильмов

Разработать программу, которая содержит сведения о фильмах.

Структура каждой записи:

- название фильма
- жанр фильма
- год выхода в прокат
- актеры
- режиссер
- продолжительность.

Вариант 11. База учеников

Разработать программу, которая содержит динамическую информацию об учениках.

Сведения о каждом ученике содержат:

- личный номер ученика;
- фамилию и инициалы ученика;
- год рождения;
- адрес.

Вариант 12. Книжный магазин

Разработать программу, которая содержит сведения о книгах в книжном магазине.

Структура каждой записи:

- ISBN

- автор
- название
- издательство
- кол-во страниц
- цена
- наличие

Вариант 13. Файловая система

В файловой системе каталог файлов организован как линейный список.

Для каждого файла в каталоге содержатся следующие сведения:

- имя файла;
- дата создания (день, месяц, год)
- время создания (часы, минуты, сек);
- количество обращений к файлу.

Вариант 14. Бюро обмена квартир

Картотека в бюро обмена квартир организована как линейный список.

Сведения о каждой квартире содержат:

- количество комнат;
- этаж;
- площадь;
- ремонт;
- адрес.

Вариант 15. Библиотека

Разработать программу, которая содержит текущую информацию о книгах в библиотеке.

Сведения о книгах содержат:

- номер УДК;
- фамилию и инициалы автора;

- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

Вариант 16. Словарь

Предметный указатель организован как линейный список.

Каждая компонента указателя содержит

- слово
- часть речи
- перевод
- номера страниц, на которых это слово встречается (количество номеров страниц, относящихся к одному слову, от одного до десяти).

Вариант 17. Касса Аэрофлота

Разработать программу, которая содержит сведения о наличии билетов и рейсах Аэрофлота.

Структура записи:

- номер рейса;
- пункт назначения;
- время вылета,
- время прибытия;
- количество свободных мест в салоне.

Вариант 18. Каталог сотовых телефонов

Разработать программу, которая содержит сведения о сотовых телефонах.

Структура каждой записи:

- артикул
- марка
- модель
- диагональ

- ОС
- объем оперативной памяти
- объем встроенной памяти
- кол-во ядер

Вариант 19. Работа цеха

Разработать программу, которая содержит сведения о количестве изделий, собранных сборщиками цеха за неделю.

Каждая запись содержит:

- фамилию сборщика;
- наименование изделия
- модель
- количество изделий, собранных им ежедневно в течении шестидневной недели;

Вариант 20. Магазин игрушек

Разработать программу, которая содержит сведения об ассортименте игрушек в магазине.

Сведения о каждой игрушке содержат:

- артикул
- вид игрушки
- название игрушки;
- цена;
- количество;
- возрастные границы (например, от 2 до 5 лет)

Вариант 21. Студенческая группа

Разработать программу, которая содержит сведения студентах группы

Структура записи:

- фамилия студента;

- размер стипендии;
- место жительства;
- семейное положение.

Вариант 22. Подписка на периодические издания

Разработать программу, которая содержит динамическую информацию о подписке на газеты и журналы.

Сведения о подписке содержат следующие поля:

- наименование (вид) издания;
- название издания,
- стоимость подписки
- город издательства
- подписчики (если есть, их может не быть);

Вариант 23. Успеваемость студентов

Разработать программу, содержащую сведения о сдаче студентами сессии.

Сведения о каждом студенте содержат:

- индекс группы;
- фамилию студента;
- оценки по пяти зачетам (з/н);
- оценки по 4-м экзаменам;

Вариант 24. Пациенты клиники

Разработать программу, которая содержит сведения о пациентах глазной клиники.

Структура каждой записи:

- фамилия пациента;
- пол;
- возраст,
- город;
- диагноз

- ФИО врача.

Вариант 25. Каталог канцелярских товаров

Разработать программу, которая содержит сведения о канцелярских товарах.

Структура каждой записи:

- артикул
- вид товара
- наименование
- цена
- наличие на складе.

Вариант 26. Сотрудники института

Разработать программу, которая содержит динамическую информацию о сотрудниках института.

Сведения о каждом сотруднике содержат:

- фамилию и инициалы сотрудника;
- название отдела;
- должность;
- стаж;
- год рождения;
- оклад.

Вариант 27. Каталог бытовой техники

Разработать программу, которая содержит сведения о бытовой технике.

Структура каждой записи:

- артикул
- вид товара
- марка
- название модели
- цена

- страна-производитель

Вариант 28. Выбор учебных дисциплин

Разработать программу, которая содержит сведения о том, какие из 5-ти предлагаемых дисциплин по выбору желает слушать студент.

Структура записи:

- фамилия студента;
- индекс группы;
- пять дисциплин;
- средний балл успеваемости.

Выбираемая дисциплина отмечается символом – 1, иначе – 0.

Вариант 29. Продажа кондитерских изделий

Разработать программу, которая содержит сведения наличия кондитерских изделий на складе.

Структура каждой записи:

- артикул
- вид товара
- наименование
- цена за 1 кг.
- наличие на складе

Вариант 30. Записная книжка

Разработать программу, которая содержит сведения о родственниках, знакомых и коллегах.

Структура каждой записи:

- фамилия
- имя
- отчество
- номер телефона

- e-mail
- почтовый адрес
- место работы

Вариант 31. Каталог автомобилей

Разработать программу, которая содержит сведения об автомобилях.

Структура каждой записи:

- фирма
- модель
- тип кузова
- год выпуска
- тип топлива
- кол-во дверей
- цена

Вариант 32. Студенты факультета

Разработать программу, которая содержит сведения о студентах некоторого факультета.

Структура каждой записи:

- фамилия, имя, отчество;
- дата рождения (день, месяц, год);
- группа;
- средний балл;
- стипендия.

Вариант 33. Orders (Заказы)

Develop a program that contains information about orders.

The structure of each record:

- Order ID
- Sales representative

- Product
- Quantity
- Price for one

(Do not use arrays.) For every operation create procedure or function. Create numeric menu or other.

Operations:

- creating an empty list
- testing whether or not a list is empty
- filling the list with elements of the specified structure
- print the elements of list in the order of occurrence
- find length of a linked list
- inserting a new element of the specified structure as the first one of a list
- searching a given element in linked list
- change an element of the specified structure (change price for one)
- deleting the element of a list with given Order ID
- deleting an element with given sales representative
- linked list deletion.