**Самостоятельная работа студента**

1. **Алгоритмы сортировки и поиска**

**Разработать программу, моделирующую базу данных с возможностью хранения, добавления, поиска, сортировки элементов структуры с последующим выводом на экран. Сортировки должны быть проведены по алгоритмам быстрой сортировки, Сортировки Шелла, сортировка выбором и сортировка вставки**

База данных должна иметь меню:

* Ввод новой записи
* Сортировка ( с выбором метода по ключу)
* Поиск
* Просмотр
* Выход

**Варианты заданий:**

1. Разработать телефонный справочник, содержащий данные об абоненте(фамилию, имя, отчество, адрес, № телефона ).

Сортировка: по номеру телефона, по Ф.О.И. владельца, по адресу владельца

Поиск: по номеру телефона, по Ф.О.И. владельца, по адресу владельца

Просмотр: все абоненты и те, которые проживают на заданной улице

1. Разработать справочник абитуриентов, содержащий следующие данные: фамилия, имя, отчество, год рождения, набранный бал.

Сортировка: по Ф.И.О., по году рождения, по проходному балу.

Поиск: по Ф.И.О.

Просмотр: всех абитуриентов, только родившихся в заданном году, только тех, проходной бал которых лежит в заданных границах.

1. Разработать классный журнал, предназначенный для учета успеваемости школьников. Ученик характеризуется фамилией, именем, отчеством, и оценками, полученными на всех занятиях по всем предметам. Учитель должен иметь возможность поставить оценку.

Сортировка: по Ф.И.О.

Поиск: по Ф.И.О.

Просмотр: оценки всех учеников по заданному предмету, всех отличников.

1. Разработать справочник товаров. Товар имеет следующие характеристики: наименование, год выпуска, цену, срок хранения, производитель.

Сортировка: по наименованию, по цене

Поиск: по наименованию, по производителю

Просмотр: всех товаров, тех, цена которых не больше заданной, всех просроченных товаров.

1. Разработать справочник сотрудников предприятия. Сотрудник характеризуется: фамилией, именем, отчеством, должностью, окладом, датой рождения.

Сортировка: по Ф.И.О, по дате рождения, по окладу.

Поиск: по Ф.И.О.

Просмотр: всех сотрудников, тех, чья должность совпадает с заданной.

1. Разработать справочник регистрации автомобилей. Автомобиль обладает следующими характеристиками: номер, Ф.И.О. владельца, марка, год выпуска.

Сортировка: по номеру, по Ф.И.О. владельца, по году выпуска

Поиск: по номеру, по Ф.И.О. владельца

Просмотр: всех автомобилей, и тех год выпуска которых равен заданному.

1. **Структуры данных**
2. **Разработать программу производящую синтаксический контроль правильности расстановки скобок в арифметическом выражении**.

Требуется разработать алгоритм и составить программу синтаксического анализа соответствия открывающих и закрывающих круглых скобок в арифметическом выражении алгоритмического языка. Программа должна напечатать таблицу соответствия скобок, причем в таблице должно быть указано, для каких скобок отсутствуют парные им. Для идентификации скобок могут быть использованы их порядковые номера в выражении. Например, для арифметического выражения языка

1 2 3 4 5 6

(a+b1)/2+6.5)\*(4.8+sin(x) должна быть напечатана таблица вида

СКОБКИ

|  |  |
| --- | --- |
| открывающая | закрывающая |
| 1 | 2 |
| - | 3 |
| 5 | 6 |
| 4 | - |

Прочерки в таблице обозначают отсутствие соответствующей скобки. При отсутствии скобок в выражении программа должна выдать соответствующее сообщени

1. **Обслуживание процессором ЭВМ очереди заданий (очередь)**

Для вычислительной системы (ВС) с одним процессором и однопрограммным последовательным режимом выполнения поступающих заданий требуется разработать программную систему для имитации процесса обслуживания заданий в ВС. При построении модели функционирования вычислительной системы должны учитываться следующие основные моменты обслуживания заданий:

генерация нового задания;

постановка задания в очередь для ожидания момента освобождения процессора;

выборка задания из очереди при освобождении процессора после обслуживания очередного задания.

При планировании очередности обслуживания заданий возможность задания приоритетов может не учитываться. Моменты появления новых заданий и моменты освобождения процессора могут рассматриваться как случайные события.

По результатам проводимых вычислительных экспериментов система имитации должна выводить информацию об условиях проведения эксперимента (интенсивность потока задания, размер очереди заданий, производительность процессора, число тактов имитации) и полученные в результате имитации показатели функционирования вычислительной системы, в т.ч.

- количество поступивших в ВС заданий;

- количество отказов в обслуживании заданий из-за переполнения очереди;

- среднее количество тактов ожидания обслуживания;

- количество тактов простоя процессора из-за отсутствия в очереди заданий для обслуживания и др. Показатели функционирования вычислительной системы, получаемые при помощи систем имитации, могут использоваться для оценки эффективности применения ВС; по результатам анализа показателей могут быть приняты рекомендации о целесообразной модернизации характеристик ВС (например, при длительных простоях процессора и при отсутствии отказов от обслуживания заданий желательно повышение интенсивности потока обслуживаемых заданий,...).

1. П**осторожить "BST - дерево" включает следующие операции:** 
   * опрос размера дерева,
   * очистка дерева,
   * проверка дерева на пустоту,
   * поиск элемента с заданным ключом,
   * включение нового элемента с заданным ключом,
   * удаление элемента с заданным ключом,
   * итератор для доступа к элементам дерева с операциями:
     + установка на корень дерева,
     + проверка конца дерева,
     + доступ к данным текущего элемента дерева,
     + переход к следующему по значению ключа элементу дерева,
     + переход к предыдущему по значению ключа элементу дерева,
     + обход дерева
2. **Построить AVL- дерево одного из типов данных с ключом поиска**.

Предусмотреть выполнение действий:

установка на корень дерева,

* проверка конца дерева,
* доступ к данным текущего элемента дерева,
* переход к следующему по значению ключа элементу дерева,
* переход к предыдущему по значению ключа элементу дерева,
* обход дерева рекурсивным способом-

при обходе подсчитать :

* общее количество вершин дерева; ( 3 балла)
* количество листьев дерева