**Практическая работа 6**

**Типы данных**

**Ответьте на вопросы:**

**Что такое запись? Из чего она состоит?**

1. Запись (record) – это составной тип данных, который объединяет несколько полей различных типов в одну структуру.  
   Пример:

var

person: TPerson;

begin

person.Name := 'Иван';

person.Age := 17;

end.

**Как объявляются и создаются записи в Паскале? Как обращаться к полям записи?**

1. Запись объявляется через type, а доступ к полям осуществляется через .  
   Пример:

type

TBook = record

Title: string;

Year: Integer;

end;

var

book: TBook;

begin

book.Title := 'Война и мир';

book.Year := 1869;

end.

1. **Зачем используются записи в виде двумерной таблицы?**

Они позволяют удобно хранить сложные структуры данных в виде таблицы, где каждая строка представлена записью.

Пример:

type

TStudent = record

Name: string;

Grade: Integer;

end;

var

Students: array[1..10] of TStudent;

1. **Как использовать with при работе с записями?**

with упрощает доступ к полям записи.

Пример:

var

person: TPerson;

begin

with person do

begin

Name := 'Анна';

Age := 25;

end;

end.

1. **Как происходит считывание/запись записей в файл?**

Используется file of record.

Пример:

type

TStudent = record

Name: string[20];

Grade: Integer;

end;

var

f: file of TStudent;

s: TStudent;

begin

Assign(f, 'students.dat');

Rewrite(f);

s.Name := 'Иван';

s.Grade := 5;

Write(f, s);

Close(f);

end.

1. **Что такое множество в Паскале? Как оно задается?**

Множество (set) – это структура, содержащая элементы одного типа в диапазоне 0..255.

Пример:

var

Letters: set of Char;

begin

Letters := ['A', 'B', 'C'];

end.

1. **Какие действия можно выполнять с множествами?**

Операции: объединение +, разность -, пересечение \*.

Пример:

var

A, B, C: set of 1..10;

begin

A := [1, 2, 3, 4];

B := [3, 4, 5, 6];

C := A \* B; // C = [3,4]

end.

1. **Для чего используется операция in при работе с множествами?**

Определяет принадлежность элемента множеству.

Пример:

if 'A' in Letters then

WriteLn('Буква A есть в множестве');

1. **Опишите динамические данные, в чем их особенность?**

Они выделяются в куче во время выполнения программы.

1. **Что такое указатель? Как он объявляется?**

Указатель – это переменная, хранящая адрес данных.

Пример:

type

PInt = ^Integer;

var

p: PInt;

x: Integer;

begin

New(p);

p^ := 10;

Dispose(p);

end.

1. **Что такое список? Какие виды списков бывают?**

Список – динамическая структура данных. Виды: односвязный, двусвязный, циклический.

1. **Что такое односвязный список?**

Структура, где каждый элемент указывает на следующий.

Пример:

type

PNode = ^TNode;

TNode = record

Data: Integer;

Next: PNode;

end;

1. **Алгоритм работы с алфавитно-частотным словарем**

Читаем текст.

Подсчитываем количество вхождений слов.

Записываем в структуру данных.

Сортируем по частоте.

1. **Что такое двусвязные списки?**

Каждый элемент содержит ссылки на предыдущий и следующий.

Пример:

type

PNode = ^TNode;

TNode = record

Data: Integer;

Prev, Next: PNode;

end;

1. **-16.Изменение алгоритма работы с двусвязными списками и сортировка**

Добавление указателя Prev, сортировка по убыванию частоты.

1. **Что такое стек? Приведите пример.**

Стек – структура LIFO (Last In, First Out).

type

TStack = array[1..10] of Integer;

1. **Принцип стека**

Последним добавленный элемент извлекается первым.

1. **Основные операции со стеком**

* Push – добавление
* Pop – удаление

1. Что такое очередь?

Очередь – структура FIFO (First In, First Out).

1. **Что такое стек?**

(повтор)

1. **Принцип очереди**

Первый добавленный элемент извлекается первым.

1. **Основные операции с очередями**

* Enqueue – добавление
* Dequeue – удаление

1. **Что такое дек?**

Двусторонняя очередь, вставка/удаление возможны с обоих концов.

type

TDeque = array[1..10] of Integer;

1. **Подпрограммы для дека**

* Добавление в начало
* Добавление в конец
* Удаление из начала
* Удаление из конца

1. **Что такое дерево?**

Иерархическая структура с узлами и связями.

1. **Что такое двоичное дерево?**

Каждый узел имеет не более двух потомков.

1. **Алгоритм поиска в дереве**

Рекурсивный обход с проверкой значения.

Пример:

function Find(Node: PNode; Value: Integer): PNode;

begin

if (Node = nil) or (Node^.Data = Value) then

Result := Node

else if Value < Node^.Data then

Result := Find(Node^.Left, Value)

else

Result := Find(Node^.Right, Value);

end;

1. Улыбнитесь😊