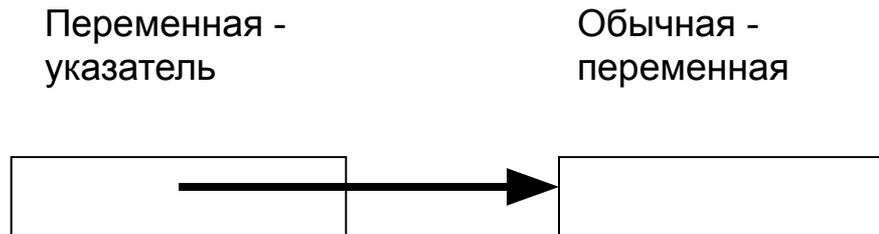


# Динамические структуры

# Переменные - указатели

- **Переменная – указатель** хранит адрес переменной.



**Имя:Тип;**

**Имя**- имя переменной-указателя; **Тип** – тип переменной, на которую может указывать переменная-указатель **Имя**

В начале работы программы переменная-указатель «ни на что не указывает».

- Указатель пустой – **NIL**.
- P2, P1: ^integer;
- P3: ^real;
- .....
- P2:=P1;
- Если P указывает на I, то в результате выполнения P^:=5;, значение I будет равно пяти.

# Динамические переменные

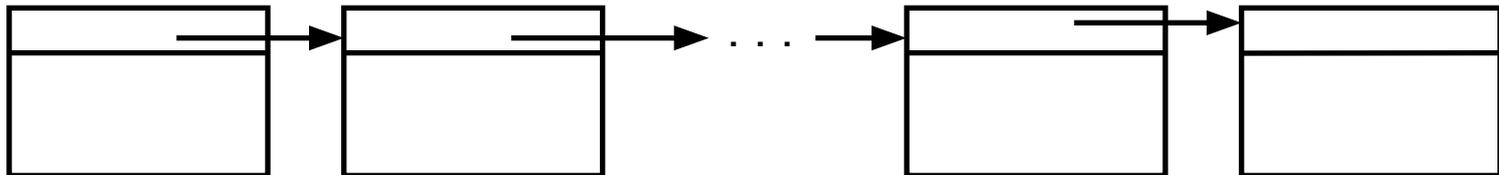
- **Динамической переменной** называется переменная, память для которой выделяется во время работы программы. Выделение памяти для динамической переменной осуществляется вызовом процедуры **NEW**. У **NEW** один параметр-указатель на переменную определенного типа, память для которой надо выделить. Пусть **P** – указатель на **real**, то в результате **NEW(p)**, будет создана переменная типа **real**, и переменная-указатель будет содержать адрес этой переменной.
- У динамической переменной нет имени, поэтому обратиться к ней можно только при помощи указателя.

# Пример программы

- //uses crt;
- **Var**
- p1,p2,p3:^integer;
- **Begin**
- **New**(p1);
- **New**(p2);
- **New**(p3);
- writeln('Введите 2 числа');
- Readln(p1^,p2^);
- P3^:=p1^+p2^;
- writeln('Сумма чисел = ',p3^);
- **End.**

# Связанные списки

- Указатели и динамические переменные позволяют создавать сложные динамические структуры данных, такие как связанные списки и деревья.



Связанные списки

- Каждый элемент списка представляет собой запись, состоящую из двух частей. 1-я – информационная. 2-я – обеспечивает связь со следующим, и возможно, с предыдущим элементом списка.
- Список, в котором обеспечивается связь только со следующим элементом, называется **односвязным**.

# Пример

Type

```
P_stud=^student;
```

```
Student=record
```

```
    Fam:string[20];
```

```
    Name:string[20];
```

```
    Group:byte;
```

```
    Address:string[60];
```

```
    Next:p_stud;{указатель на следующий элемент  
списка}
```

```
End;
```

```
Var
```

```
    Head:p_stud;
```

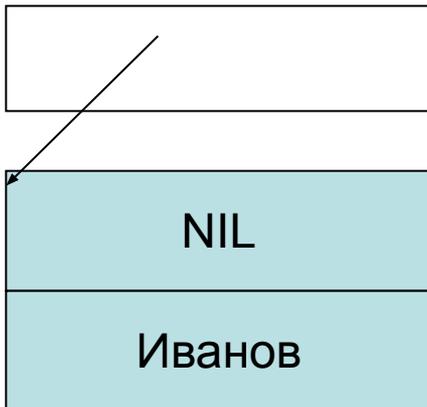
Добавлять данные можно в начало, в конец или в нужное место списка.

head



Список пустой: head ни на что не указывает

head



После добавления одного элемента head указывает на этот элемент

- **type**
- p\_student:=^student;
- student=**record**
- name:string[20];
- next:p\_student;
- **end;**
- **var**
- head:p\_student;
- curr:p\_student;
- buf:string[20];
- **begin**
- **while** buf<>'\*' **do**
- **begin**
- write('фамилия->');
- readln(buf);
- **if** buf<>'\*' **then**
- **begin**
- **new**(curr);
- curr^.name:=buf;
- curr^.next:=head;
- head:=curr;
- **end;**
- **end;**
- writeln('\*\*\*\*\*Введенный список \*\*\*\*\*');
- curr:=head;
- **while** curr<>nil **do**
- **begin**
- writeln(curr^.name);
- curr:=curr^.next ;
- **end;**
- **end.**