*Односвязные списки*

*26.01.2022*

*Односвзный список -*

*Для того чтобы создать динамический список необходимо указать на его голову Head где хранится адрес ячейки*

*С помощью функции создать добавление узла в односвязный список перед данным узлом*

1. *Создать односвязный список*

*Type*

*pNode:^Node;*

*Node: record;*

*a: integrer;*

*Next: pNode; //*

*указатель на Node*

*End;*

*Var*

*Head, tmp: pNode;*

*Begin*

*Head:=nil;*

*New(tmp); //создает новую запись, пока еще пустую.*

*Tmp^.a:=5; // присвается значение записи.*

*Tmp^.next:=Head // Поместить адрес head в tmp*

*Head:=tmp*

*2)Найти узел с заданным содержанием*

*Tmp:=head ;*

*Pp:=nil*

*While tmp<>nill and pp=nil do*

*Begin*

*If (tmp^.a=6) pp:=tmp;*

*Tmp:= tmp^.next;*

*End;*

*Head:=AddBefore (Head, pp, 200);*

*End.*

*3)Обратится к функции*

*Function AddBefore (Head, pp, : pNode; I: integer): pNode;*

*Var*

*Z, P, Tmp:pNode*

*Begin*

*New(tmp);*

*Tmp^.a:=I;*

*Tmp^.next:=pp;*

*Z:=Head;*

*While Z^.next <> pp do*

*Z:=Z^.next;*

*Z^.next:=tmp;*

*AddBefore:= Head;*

*End;*

*4)Вывести на печать*

*Tmp:= Head;*

*While tmp <> nil do*

*Begin*

*Writeln (tmp^.a);*

*Tmp:=tmp^.next;*

*End;*

1. *Удаление из списка*

*Function del Node (Head:pNode; I:integer):pNode;*

*Var*

*K, Tmp: pNode;*

*Begin*

*Tmp:=Head;*

*While tmp <> nil and tmp^.a <> I do*

*Begin*

*If (tmp^.a = I) then*

*End;*

*Tmp:= Head;*

*While tmp^.next <>k do*

*Tmp:=tmp^.next;*

*Tmp^.next:= k^.next;*

*Dispose (k);*

*End;*

*End.*

# 27.02.2022

*Type*

*Person = class*

*Private*

*fName : string;*

*fAge : integer;*

*Public*

*Constructor Create(Name : string; Age : unteger);*

*Begin*

*fName := Name;*

*fAge := Age;*

*End;*

*Procedure Print;*

*Property Name : string, read fName;*

*Property Age : integer, read fAge;*

*End;*

*Person = clas*

*Private*

*….*

*Public*

*,.,,.,*

*End;*

Коструктор

*Public*

*Conctructor Create(Name : string);*

*Begin*

*fName := Name*

*fAge := Age;*

*End;*

# Описание метода

Procedure Persone.Print;

Begin

Writeln($ ‘имя’ : (Name) возраст : (Age));

End;

# Использование

Переменные типа класс называются объектами или экземплярами класса. Они инициализируются вызовом конструктора класса - специального метода, выделяющего память под объект класса и инициализирующего его поля.:

Var p : Peron := new Person (‘Иванов’, ‘20’)

p.Print

**Моделирование на классах**

**Задача Мост**

**Условие: Через реку имеется мост с ограничением по весу, в начале моста стоит шлагбаум с весами, если вес больше ограничения, то машина съезжать в лево**

**Класс весы**

**type**

**Wight = class**

**private W:integer;**

**public**

**constructor Create;**

**begin**

**self.W := 0;**

**end;**

**procedure setw(W:integer);**

**begin**

**self.W := W;**

**end;**

**Function getw():integer;**

**begin**

**getW := W;**

**end;**

**end;**

**Класс Шлагбаум**

**type**

**barrier = class**

**private status : boolean;**

**public**

**Constructor Create;**

**Begin**

**self.status:=false;**

**End;**

**procedure open**

**Begin**

**self.status:=true;**

**End;**

**procedure close;**

**Begin**

**self.status:=false;**

**End;**

**End;**

**Класс Машина**

**type**

**car = class**

**private wight:integer;**

**public**

**constructor Create(wight:integer);**

**begin**

**self.wight := wight;**

**end;**

**function getw(): integer;**

**begin**

**writeln('автомобиль весит ', wight);**

**getW := wight;**

**end;**

**end;**

**Класс Контролёр**

**type**

**Controller = class**

**private**

**w:Wight;**

**b:Barier;**

**MaxWight:integer;**

**public**

**constructor Create(MaxWight:integer; w:Wight; b:Barier);**

**begin**

**self.w := w;**

**self.b := b;**

**self.MaxWight := MaxWight;**

**end;**

**procedure control(c:Car);**

**begin**

**w.setw(c.getW);**

**if w.getW < MaxWight then**

**begin**

**b.open();**

**b.close();**

**end**

**else**

**writeln('превышение веса');**

**end;**

**end;**

**Весь код**

**type**

**Wight = class**

**private W:integer;**

**public**

**constructor Create;**

**begin**

**self.W := 0;**

**end;**

**procedure setw(W:integer);**

**begin**

**self.W := W;**

**end;**

**Function getw():integer;**

**begin**

**getW := W;**

**end;**

**end;**

**type**

**barier = class**

**private status:boolean;**

**public**

**constructor Create;**

**begin**

**self.status := false;**

**end;**

**procedure open;**

**begin**

**writeln('Шлагбаум открыт');**

**self.status := true;**

**end;**

**procedure close;**

**begin**

**writeln('Шлагбаум закрыт');**

**self.status := false;**

**end;**

**end;**

**type**

**car = class**

**private wight:integer;**

**public**

**constructor Create(wight:integer);**

**begin**

**self.wight := wight;**

**end;**

**function getw(): integer;**

**begin**

**writeln('автомобиль весит ', wight);**

**getW := wight;**

**end;**

**end;**

**type**

**Controller = class**

**private**

**w:Wight;**

**b:Barier;**

**MaxWight:integer;**

**public**

**constructor Create(MaxWight:integer; w:Wight; b:Barier);**

**begin**

**self.w := w;**

**self.b := b;**

**self.MaxWight := MaxWight;**

**end;**

**procedure control(c:Car);**

**begin**

**w.setw(c.getW);**

**if w.getW < MaxWight then**

**begin**

**b.open();**

**b.close();**

**end**

**else**

**writeln('превышение веса');**

**end;**

**end;**

**var**

**w:Wight;**

**b:Barier;**

**c:Car;**

**control:Controller;**

**begin**

**w:=new Wight;**

**b:= new Barier;**

**control:= new Controller(5000, w, b);**

**for i:integer := 1 to 5 do**

**begin**

**c:= new Car(random(2, 6)\*1000);**

**control.control(c);**

**end;**

**end.**