







Conceptos de Algoritmos Datos y Programas

CADP – TEMAS



Tipos de Datos definidos por el usuario

- Tipos de Datos subrango
- EJERCICIO PREGUNTAS FINALES

CADP – TIPOS DE DATOS- CLASIFICACION



SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE

COMPUESTO

Integer real char boolean

DEFINIDO POR EL LENGUAJE

DEFINIDO POR EL PROGRAMADOR

Son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

Permiten definir nuevos tipos de datos a partir de los tipos simples. string





SUBRANGO

Es un tipo ordinal que consiste de una sucesión de valores de un tipo ordinal (predefinido o definido por el usuario) tomado como base.

- Es simple.
- Es ordinal
- Existe en la mayoría de los lenguajes

Cómo se declara?

Cómo se

CADP – TIPOS DE DATOS- CLASIFICACION



SIMPLE: aquellos que toman un único valor, en un momento determinado, de todos los permitidos para ese tipo.

TIPO DE DATO

COMPUESTO: pueden tomar varios valores a la vez que guardan alguna relación lógica entre ellos, bajo un único nombre.

SIMPLE

COMPUESTO

string

Integer real char boolean

DEFINIDO POR EL LENGUAJE

DEFINIDO POR EL PROGRAMADOR

subrango

Son provistos por el lenguaje y tanto la representación como sus operaciones y valores son reservadas al mismo.

Permiten definir nuevos tipos de datos a partir de los tipos simples.





```
program uno;
Const
Type
    tipo1 = valor1..valor2;
var
   x,y: tipo1;
Begin
  •••
End.
```





Qué operaciones puedo hacer con cada variable?

```
program uno;
Const
Type
    notas = 0..10;
    letras = 'a'..'1';
var
   not1,not2: notas;
   let:letras;
Begin
End.
```

Se puede hacer un subrango de reales?





SUBRANGO – OPERACIONES

Operaciones Permitidas

- Asignación
- Comparación
- Todas las operaciones permitidas para el tipo base

Operaciones NO permitidas

 Depende del tipo base.

CADP – EJERCITACION



Programa 1: Se pide realizar que lea caracteres (suponga sólo minúsculas) hasta leer el carácter 'z'. Al finalizar informar cuantos caracteres leídos estaban entre 'a' y 'h'; cuántos entre 'i' y 'n' y cuántos entre 'ñ' y 'y'.

- Estructura de control principal del programa?
- Necesito definir algún tipo de datos?
- Cómo cuento según el valor?

CADP - EJERCITACION



Qué tiene de malo esta solución?

Qué puede pasar?

```
Program uno;
var
 primer, segunda, tercer :integer;
 letra:char;
Begin
  primer:= 0; segunda:=0; tercer:=0;
  read (letra);
  while (letra <> 'z') do
     begin
       if ((letra = 'a') or (letra = 'b') or ...) then
          primer:= primer + 1
       else
           if ((letra = 'i') or (letra = 'j') or ...) then
            segunda:= segunda + 1
           else
             tercer:= tercer + 1;
       read (letra);
     end;
   write (primera, segunda, tercera);
end.
```

CADP – EJERCITACION



```
Program uno;
Type
                                                   Se puede
  letras = 'a'...'z';
var
                                                    mejorar?
 primer, segunda, tercer: integer;
 letra:letras;
Begin
  primer:= 0; segunda:=0; tercer:=0;
  read (letra);
 while (letra <> 'z') do
     begin
       if ((letra = 'a') or (letra = 'b') or ...) then
          primer:= primer + 1
       else
           if ((letra = 'i') or (letra = 'j') or ...) then
            segunda:= segunda + 1
           else
             tercer:= tercer + 1;
      read (letra);
     end;
   write (primer, segunda, tercer);
end.
```

CADP - EJERCITACION Program uno;



```
Type
  letras = 'a'..'z';
var
 primer, segundo, tercer: integer;
 letra:letras;
Begin
  primer:= 0; segunda:=0; tercer:=0;
  read (letra);
  while (letra <> 'z') do
     begin
       case letra of
         'a'..'h': primer:= primer + 1;
         'i'...'n': segunda:= segunda + 1;
         'ñ'...'y': tercer:= tercer + 1;
        end;
      read (letra);
     end;
   write (primer, segunda, tercer);
end.
```