

Estructuras (struct)

Conceptos

Tomás Peiretti

Estructuras

Una estructura es un conjunto de datos agrupados bajo un mismo nombre. Estos datos se conocen como **miembros** y pueden tener diferentes tipos.

```
struct nombre {  
    tipo nombre_miembro1;  
    tipo nombre_miembro2;  
    ...  
};
```

```
1 struct Alumno {  
2     int edad;  
3     char carrera;  
4     int notas[45];  
5     int tl_notas;  
6 };  
7  
8 int main() {  
9     Alumno a1, a2;  
10 }
```

Estructuras: acceso a los miembros

Los miembros de la estructura pueden ser accedidos de manera directa haciendo uso del '.' como se muestra en el siguiente ejemplo:

```
1 struct Alumno {
2     int edad;
3     char carrera;
4     int notas[45];
5     int tl_notas = 0;
6 } a1;
7
8 int main() {
9     a1.edad = 19;
10    a1.carrera = 'S';
11    a1.notas[0] = 10;
12    a1.notas[1] = 9;
13    a1.tl_notas = 2;
14
15    cout << "edad: " << a1.edad << endl; // imprime 19
16    cout << "carrera: " << a1.carrera << endl; // imprime S
17    cout << "nota en AEED: " << a1.notas[0] << endl; // imprime 10
18 }
```

Estructuras: anidaciones

Además, es posible utilizar estructuras dentro de otras estructuras:

```
1 struct Punto {
2     int x;
3     int y;
4 };
5
6 struct Recta {
7     Punto punto;
8     double pendiente;
9 };
10
11 int main() {
12     Punto p1 = {0, 10};
13     Recta r1;
14     r1.punto = p1;
15     r1.pendiente = 2.5;
16
17     cout << "La recta tiene pendiente = " << r1.pendiente << endl;
18     cout << "y pasa por (" << r1.punto.x << "," << r1.punto.y << ")" << endl;
19 }
```

Estructuras: pasaje por parámetros

Todas las estructuras por defecto **pasan por copia**.

```
1 struct Punto {
2     int x;
3     int y;
4 };
5 struct Recta {
6     Punto punto;
7     double pendiente;
8 };
9
10 void limpiarPunto1(Recta & r) {
11     r.punto.x = 0;
12     r.punto.y = 0;
13 }
14 void limpiarPunto2(Recta r) {
15     r.punto.x=-1;
16     r.punto.y=-1;
17 }
18
19 int main() {
20     Recta r1 = {{5, 10}, 2.5};
21     limpiarPunto2(r1);
22     // como no pasa por referencia, imprime (5,10)
23     cout << "(" << r1.punto.x << "," << r1.punto.y << ")" << endl;
24     limpiarPunto1(r1);
25     // como pasa por referencia, imprime (0,0)
26     cout << "(" << r1.punto.x << "," << r1.punto.y << ")" << endl;
27 }
```

Ejercicio

Supongamos el siguiente problema:

Se deben leer los datos de diferentes animales e imprimir el nombre de los animales **ordenados segun su altura**.

Primero se ingresa un entero N, que indica la cantidad de animales. Luego siguen N lineas, cada una indicando el nombre, la altura (en cm) y el peso (en kg) de un animal

Ejercicio: solución

```
1 struct Animal {
2     string nombre;
3     double peso;
4     double altura;
5 };
6
7 void ordenarPorAltura(Animal a[], int tl);
8
9 int main() {
10     int n;
11     Animal animales[10000];
12
13     cin >> n;
14     for (int i=0; i<n; i++)
15         cin >> animales[i].nombre >> animales[i].peso >> animales[i].altura;
16
17     ordenarPorAltura(animales, n);
18     for (int i=0; i<n; i++)
19         cout << animales[i].nombre << endl;
20 }
21
22 void ordenarPorAltura(Animal a[], int tl) {
23     for (int i=0; i<tl; i++) {
24         for(int j=i+1; j<tl; j++) {
25             if (a[i].altura > a[j].altura)
26                 swap(a[i], a[j]);
27         }
28     }
29 }
```

Ejercicios

- Beecrowd 1258
- Ejercicios de la guía práctica de estructuras