Programación orientada a objetos

Nombre: Oscar Alejandro Penilla Skakievich

Investigación 1

Fecha: 14/2/22

Grupo:4C1

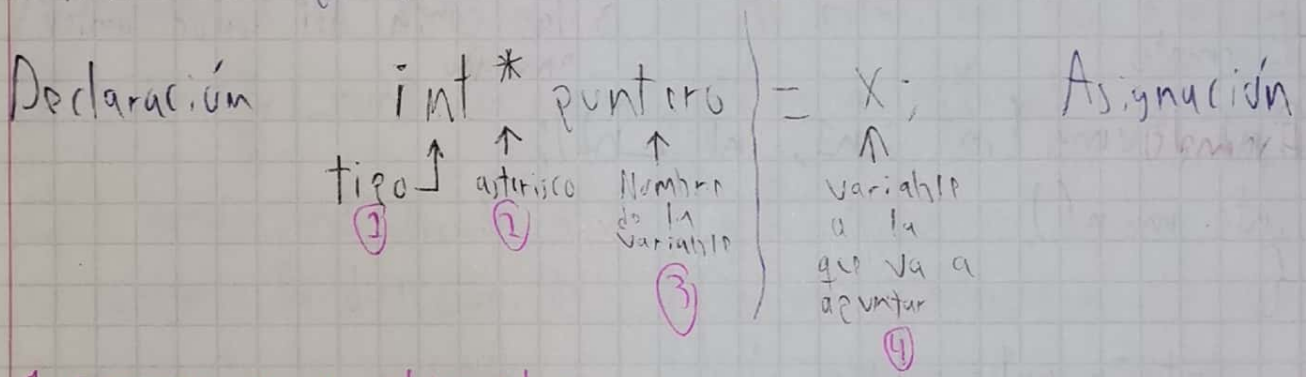
Oscar Armilla Skakirvich
20300tol

Investigación

Que es un puntero?

Un puntero es una variable que almacena la dirección de memoria, al almacenar una dirección de memoria le podemos asignar un valor

Sintaxis para el uso de punteros



1 Siempre se especifica el tipo y debe de ser congruente con lo que va a apuntar

2 Este se usa para que se sepa que se habla y declara un puntero

3 El nombre con el que se va a identificar ese espacio en la memoria

4 Puede ser cualquier variable del mismo tipo

En este ejemplo "X" no podria ser un char, debe de ser un entero

Paso de parametro por valor

Descripcion: En esta al hacer una función se va a hacer una copia de las variables que se pasan a esta, si se modifican en la funcion no les afecta en el main o de donde las envian

Sintaxis

Al llamar la funcion

① ②

función(valor)

1 nombre de la funcion

2 valor que vamos a enviar

Al estar en la función

① ② ① ③

int funcion(int valor)

1 el tipo tanto de la función como de la variable

2 el nombre de la funcion

3 la copia del valor antes enviado

-Ejemplo

```
int suma(int i_n1, int i_n2);
int main()
{
    int i_num1 = 1;
    int i_num2 = 4;
cout<<suma(i_num1, i_num2);
}
int suma(int i_n1, int i_n2)// 1, 4
{
    i_n1++; // 2  pero i_num1 sigue siendo 1
    i_n2 = i_n2-2; // 2  pero i_num2 sigue siendo 4
    return i_n1 + i_n2; // 4
}
```

Las primeras dos variables nunca cambiaron, solo se les hizo una copia

Paso de parametros por referencia

Descripción: La principal diferencia en comparacion al paso de parametros por valor es que en este si afecta el valor de las variables que enviamos, ademas que en esta se usan punteros

Sintaxis

Al llamar a la función

```
funcion(&i_num1, &i_num2);
```

(2)

Al estar en la función

```
intFuncion(int* i_n1, int* i_n2)
```

(1)

1 El ampersand ayuda a que se obtenga la dirección de memoria de la variable

2 El asterisco ayuda a la funcion a acceder a ese lugar de la memoria dond está la variable

→ Nota: Los demás elementos no señalados son los mismos que en el paso de parametros por valor

Ejemplo en la siguiente pagina.

Ejemplo

```
int suma(int* i_n1, int* i_n2);
int main()
{
  int i_num1=1;
  int i_num2=4;
  cout << suma(&i_num1, &i_num2);//aparece un 4
}//pero tanto i_num1 como i_num2, ahora valen 2
int suma(int* i_n1, int* i_n2)
{
    i_n1++;//2, como en el pasado paso lo mismo
    i_n2 = i_n2-2;//2
    return i_n1+i_n2;//returnamos 2+2
}
```

Quizas en este codigo no se pueda apreciar mucho pero si cambiaramos esta funcion por un proceso (void) seguiria dandonos el 4, ya que si bien no devolveria nada, las variables valdrian 2.