

Introducción a Java

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza

C++

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     cout << "Hello, World !!" << endl;
8
9     return 0;
10 }
```

```
1 > g++ -o hello hello.cc ↵
2 > hello ↵
```

Java

```
1 public class hello
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello, World !!");
6     }
7 }
```

```
1 > javac hello.java ↵
2 > java hello ↵
```

C++

```
1 #include <string>
2 #include <iostream>
3
4 using namespace std;
5
6 int main(int argc, char* argv[])
7 {
8     string name;
9     int age;
10
11     cout << "What's your name ? " << flush;
12     cin >> name;
13     cout << "What's your age ? " << flush;
14     cin >> age;
15     cout << "Hello, " << name << " !" << endl;
16     cout << "You're " << age << " years old" << endl;
17
18     return 0;
19 }
```

Java

```
1 import java.util.Scanner;
2
3 public class tst
4 {
5     public static void main(String[] args)
6     {
7         String name = new String();
8         int age;
9
10        Scanner scanner = new Scanner(System.in);
11
12        System.out.print("What's your name ? ");
13        System.out.flush();
14        name = scanner.next();
15        System.out.print("What's your age ? ");
16        System.out.flush();
17        age = scanner.nextInt();
18        System.out.println("Hello, "+name+" !");
19        System.out.println("You're "+age+" years old");
20    }
21 }
```

Declaración mínima de una **clase**.

C++

```
1 class T
2 {
3 };
4
5 int main()
6 {
7     int age;
8     T t;
9
10    return 0;
11 }
```

Java

```
1 class T
2 {
3 }
4
5 public class tst
6 {
7     public static void main(String[] args)
8     {
9         int age;
10        T t = new T();
11    }
12 }
```



O bien...

C++

```
1 class T
2 {
3 };
4
5 int main()
6 {
7     int age;
8     T* t = new T();
9
10    delete t;
11
12    return 0;
13 }
```

Java

```
1 class T
2 {
3 }
4
5 public class tst
6 {
7     public static void main(String[] args)
8     {
9         int age;
10        T t = new T();
11    }
12 }
```



Atributos

C++

```
1 #include <iostream>
2
3 class T
4 {
5 public:
6     int age;
7 };
8
9 int main(int argc, char** argv)
10 {
11     T t;
12     t.age = 10;
13     std::cout << t.age << std::endl;
14     return 0;
15 }
```

Java

```
1 class T
2 {
3     public int age;
4 }
5
6 public class tst
7 {
8     public static void main(String[] args)
9     {
10         T t = new T();
11         t.age = 10;
12         System.out.println(t.age);
13     }
14 }
```



O bien...

C++

```
1 #include <iostream>
2
3 class T
4 {
5 public:
6     int age;
7 };
8
9 int main(int argc, char** argv)
10 {
11     T* t = new T();
12     t->age = 10;
13     std::cout << t->age << std::endl;
14     delete t;
15     return 0;
16 }
```

Java

```
1 class T
2 {
3     public int age;
4 }
5
6 public class tst
7 {
8     public static void main(String[] args)
9     {
10         T t = new T();
11         t.age = 10;
12         System.out.println(t.age);
13     }
14 }
```



Métodos

C++

```

1  class T
2  {
3  private:
4  int age;
5  public:
6  void set_age(int age)
7  { this->age = age; }
8  int get_age() const
9  { return age; }
10 };
11
12 int main(int argc, char** argv)
13 {
14     T t;
15     t.set_age(10);
16     std::cout << t.get_age() << std::endl;
17     return 0;
18 }

```

Java

```

1  class T
2  {
3  private int age;
4  public void set_age(int age)
5  { this.age = age; }
6  public int get_age()
7  { return age; }
8  }
9
10 public class tst
11 {
12     public static void main(String[] args)
13     {
14         T t = new T();
15         t.set_age(10);
16         System.out.println(t.get_age());
17     }
18 }

```


O bien...

C++

```

1  class T
2  {
3  private:
4      int age;
5  public:
6      void set_age(int age)
7          { this->age = age; }
8      int get_age() const
9          { return age; }
10 };
11
12 int main(int argc, char** argv)
13 {
14     T* t = new T();
15     t->set_age(10);
16     std::cout << t->get_age() << std::endl;
17     delete t;
18     return 0;
19 }

```

Java

```

1  class T
2  {
3      private int age;
4      public void set_age(int age)
5          { this.age = age; }
6      public int get_age()
7          { return age; }
8  }
9
10 public class tst
11 {
12     public static void main(String[] args)
13     {
14         T t = new T();
15         t.set_age(10);
16         System.out.println(t.get_age());
17     }
18 }

```

Constructores

C++

```
1 class T
2 {
3 private:
4     int age;
5 public:
6     T(int _age) : age(_age) { }
7     int get_age() const { return age; }
8 };
9
10 int main(int argc, char** argv)
11 {
12     T t(10);
13     std::cout << t.get_age() << std::endl;
14
15     return 0;
16 }
```

Java

```
1 class T
2 {
3     private int age;
4     public T(int age) { this.age = age; }
5     public int get_age() { return this.age; }
6 }
7
8 public class tst
9 {
10     public static void main(String[] args)
11     {
12         T t = new T(10);
13         System.out.println(t.get_age());
14     }
15 }
```

O bien...

C++

```

1  class T
2  {
3  private:
4      int age;
5  public:
6      T(int _age) : age(_age) { }
7      int get_age() const { return age; }
8  };
9
10 int main(int argc, char** argv)
11 {
12     T* t = new T(10);
13     std::cout << t->get_age() << std::endl;
14
15     delete t;
16
17     return 0;
18 }

```

Java

```

1  class T
2  {
3      private int age;
4      public T(int age) { this.age = age; }
5      public int get_age() { return this.age; }
6  }
7
8  public class tst
9  {
10     public static void main(String[] args)
11     {
12         T t = new T(10);
13         System.out.println(t.get_age());
14     }
15 }

```



¿Funciona esto?

C++

```

1  class T { ... }
2
3  class U {
4  private:
5      T* t;
6  public:
7      U(int age) : t(new T(age)) { }
8      int get_age() const { return t->age; }
9  };
10
11 int main(int argc, char** argv)
12 {
13     U* u = new U(10);
14     std::cout << u->get_age() << std::endl;
15     delete u;
16     return 0;
17 }

```

Java

```

1  class T { ... }
2
3  class U {
4      private T t;
5      public U(int age) { t = new T(age); }
6      public int get_age() { return t.age; }
7  }
8
9  public class tst
10 {
11     public static void main(String[] args)
12     {
13         U u = new U(10);
14         System.out.println(u.get_age());
15     }
16 }

```



Destructores

C++

```

1  class T { ... };
2
3  class U {
4  private:
5      T* t;
6  public:
7      U(int age) : t(new T(age)) { }
8      ~U() { delete t; }
9
10     int get_age() const { return t->age; }
11 };
12
13 int main(int argc, char** argv)
14 {
15     U* u = new U(10);
16     std::cout << u->get_age() << std::endl;
17     delete u;
18     return 0;
19 }

```

Java

```

1  class T { ... }
2
3  class U {
4      private T t;
5      public U(int age) { t = new T(age); }
6      public int get_age() { return t.age; }
7  }
8
9  public class tst
10 {
11     public static void main(String[] args)
12     {
13         U u = new U(10);
14         System.out.println(u.get_age());
15     }
16 }

```

Destruyores

¿Funciona esto?

C++

```

1  class T { ... }
2
3  class U {
4  private:
5      T* t;
6  public:
7      U(T* _t) : t(_t) { }
8      ~U() { delete t; }
9
10     int get_age() const { return t->age; }
11 };
12
13 int main(int argc, char** argv)
14 {
15     T* t = new T(10);
16     U* u = new U(t);
17     std::cout << u->get_age() << std::endl;
18     delete t;
19     delete u;
20     return 0;

```

Java

```

1  class T { ... }
2
3  class U {
4      private T t;
5      public U(T t)
6      { this.t = t; }
7      public int get_age() { return t.age; }
8  }
9
10 public class tst
11 {
12     public static void main(String[] args)
13     {
14         T t = new T(10);
15         U u = new U(t);
16         System.out.println(u.get_age());
17     }
18 }

```

Gestión automática de memoria dinámica,
mediante *recolección de basura* (*garbage collection*)

C++

NO

Java

SÍ



Gestión automática de memoria dinámica,

C++

NO

Java

SÍ

¿ Seguro ?



Punteros inteligentes (*smart pointers*)

C++11

```

1  #include <memory>
2
3  class T { ... }
4
5  class U {
6  private:
7      shared_ptr<T> t;
8  public:
9      U(shared_ptr<T> _t) : t(_t) { }
10     int get_age() const { return t->age; }
11 };
12
13 int main(int argc, char** argv)
14 {
15     shared_ptr<T> t = make_shared<T>(10);
16     unique_ptr<U> u = make_unique<U>(t);
17
18     cout << u->get_age() << endl;
19
20     return 0;
21 }

```

Java

```

1  class T { ... }
2
3  class U {
4      private T t;
5      public U(T t) { this.t = t; }
6      public int get_age() { return t.age; }
7  }
8
9  public class tst
10 {
11     public static void main(String[] args)
12     {
13         T t = new T(10);
14         U u = new U(t);
15         System.out.println(u.get_age());
16     }
17 }

```

Introducción a Java

Tecnología de Programación



Adolfo Muñoz – Juan Magallón
Grado en Ingeniería Informática



Universidad
Zaragoza



Escuela de
Ingeniería y Arquitectura
Universidad Zaragoza



Departamento de
Informática e Ingeniería
de Sistemas
Universidad Zaragoza