

Enseñanza de C++ al Estudiante Java

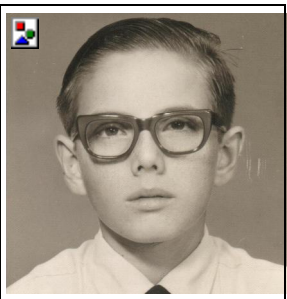
adolfo.dimare@ecci.ucr.ac.cr

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

Agenda

- Especificación y Reutilización
- C++ Para Programación 2
- Convenciones de Programación
- Transición de Java A C++
- Bibliotecas de Programas
- Prueba de Programas
- El Curso Programación 2
- Java es una Mejor Elección



UCR



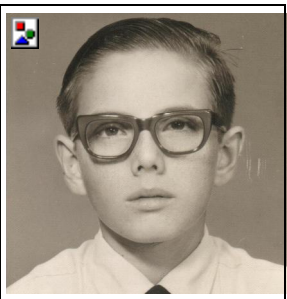
Enseñanza de C++ al Estudiante Java

adolfo.dimare@ecci.ucr.ac.cr

Especificación y Reutilización de Módulos

Módulo

Sección de un programa bien construida, con un fin específico, y que puede ser reutilizado.



UCR



Java → C++

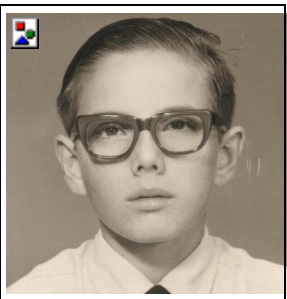
adolfo.dimare@ecci.ucr.ac.cr

Especificación y Reutilización de Módulos

Reutilización

Reutilizar significa no inventar de nuevo la rueda, aprovechando lo que otros hicieron para construir nuevas soluciones (@1992).

“Reutilización de algoritmos” (@Stepanov)



UCR



Java → C++

adolfo.dimare@ecci.ucr.ac.cr

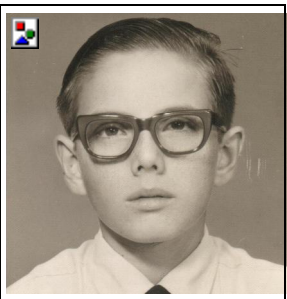
Especificación y Reutilización de Módulos

Especificación

Puede verse como un contrato en el que están definidos todos los servicios que la implementación del módulo es capaz de dar.

Toda especificación debe tener estas tres cualidades:

- Debe ser completa (debe decirse todo lo que hay que decir)
- Debe ser correcta (debe omitirse lo que no hay que decir)
- No puede ser ambigua



UCR



Java → C++

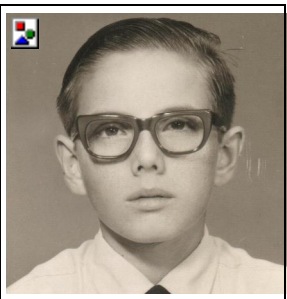
adolfo.dimare@ecci.ucr.ac.cr

C++ para Programación 2

Fuerte soporte

- Verificación de tipos
- Parametrización
- Herencia
- Polimorfismo

→ Reutilización



UCR



Java → C++

adolfo.dimare@ecci.ucr.ac.cr

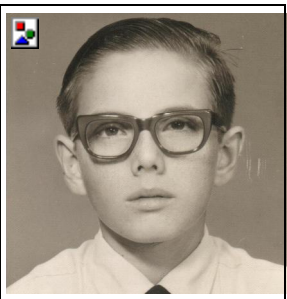
C++ para Programación 2

Cobertura C++

“Incorpora la mayor parte de los avances en programación,
sin sacrificar eficiencia”

Stroustrup, Bjarne: [Why C++ is not just an Object-Oriented Programming Language](#), OOPSLA 1995.

C++ → [PHP Visual Basic C# Java Python Ruby]



UCR

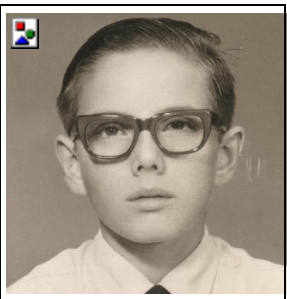


Java → C++

adolfo.dimare@ecci.ucr.ac.cr

Convenciones de Programación

- Especificación correcta, completa y no ambigua.
- Correcta indentación del código fuente.
- Correcto espaciado del código fuente.
- Código fuente escrito de manera legible y clara.
- Uso de identificadores significativos.



UCR



Java → C++

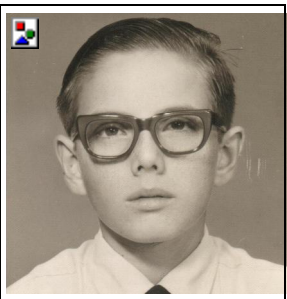
adolfo.dimare@ecci.ucr.ac.cr

Bibliotecas de Programas

Herramientas C++

- Doxygen <http://Doxygen.org>
- Code::Blocks <http://CodeBlocks.org>
- Boost <http://Boost.org>
- FLTK <http://FLTK.org>
- SQLite <http://SQLite.org>

¿Cómo se hace la transición de Java a C++?



UCR


```

#ifndef Acumulador_h
#define Acumulador_h ///< Evita la inclusión múltiple
class Acumulador{private:long m_total;long m_cantidad;long m_menor;long
m_mayor;public:Acumulador():m_total(0),m_cantidad(0),m_menor(0)
,m_mayor(0){/*borre();*/}~Acumulador(){}Acumulador(const
Acumulador&o){*this=o;}const Acumulador& operator=(const Acumulador&o){
m_total=o.m_total;m_cantidad=o.m_cantidad;m_menor
=o.m_menor;m_mayor=o.m_mayor;return*this;}void borre()
{m_total = m_cantidad = m_menor = m_mayor = 0;}long total() const{
return((m_cantidad <= 2) ? 0 : m_total-m_menor-m_mayor);}long
cantidad() const{return m_cantidad;}void acumule(long
val){m_total += val;m_cantidad++;if(m_cantidad > 1){
m_menor =( val < m_menor ? val : m_menor);m_mayor =( val > m_mayor
? val : m_mayor);}else{m_menor = m_mayor = val;}}void acumule( unsigned
n, const long * val){for(unsigned i=0; i<n; ++i){acumule(val[i]);
}}friend bool operator==(const Acumulador&A,const Acumulador&B){return(
A.m_total==B.m_total&&A.m_cantidad==B.m_cantidad&&A.m_menor==B.m_menor
&&A.m_mayor==B.m_mayor);}friend bool operator!=(const
Acumulador&A,const
Acumulador&B){return!(A==B);}friend bool check_ok( const Acumulador&
A);friend class test_Acumulador;}; /* bool */
#endif

```



Java → C++

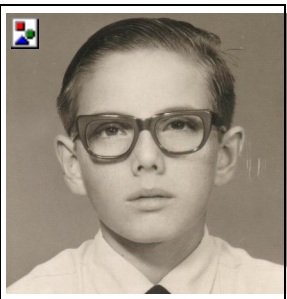
adolfo.dimare@ecci.ucr.ac.cr

Prueba de Programas

```
Acumulador::Acumulador() [inline]
```

Constructor de vector.

```
{ { // test::constructor()
    Acumulador A;
    assertTrue( 0 == A.total() );
    assertTrue( 0 == A.cantidad() );
    long vec[] = { 1, 2, 3, 4, 5 };
    A.acumule( DIM(vec) , vec );
    assertTrue( 2+3+4 == A.total() );
    A.borre();
    assertTrue( 0 == A.total() );
} }
```



UCR



Java → C++

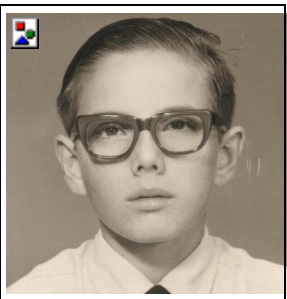
adolfo.dimare@ecci.ucr.ac.cr

El Curso Programación 2

- 11 → SDF/Algorithms and Design
- 10 → SDF/Development Methods
- 3 → SE/Software Verification Validation
- 24 Horas totales

Además:

- PD/Parallel Decomposition
- PD/Parallelism Fundamentals
- PBD/Mobile Platforms



UCR

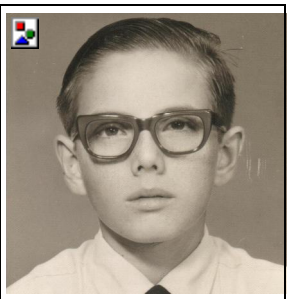


Java → C++

adolfo.dimare@ecci.ucr.ac.cr

Java es una Mejor Elección

- Plantillas o programación genérica incorporada en el lenguaje.
- Programación “pegada al hierro”.
- Apoyo para la programación orientada a objetos en la forma de parametrización y polimorfismo.
- Disponibilidad adecuada de compiladores y ambientes de compilación.
- Apoyo para separar la abstracción de cada módulo de su implementación efectiva.
- Disponibilidad de herramientas para la generación de documentación.



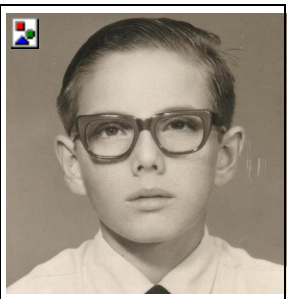
UCR



Java → C++
adolfo.dimare@ecci.ucr.ac.cr

Popularidad

1	C	19.9%
2	Java	17.2%
3	Objective-C	9.5%
4	C++	9.3%
5	C#	6.5%



UCR

Enseñanza de C++ al Estudiante Java

adolfo.dimare@ecci.ucr.ac.cr

Universidad de Costa Rica

Escuela de Ciencias de la Computación e Informática

Análisis Final

- No concentrarse en la sintaxis de C++
- Producir programas de calidad

- Especificación de módulos
- Reutilización de módulos
- Uso de ejemplos de prueba unitaria
- Complementar la documentación

[<http://www.di-mare.com/adolfo/p/cppjava.htm>]

