

Presentación de la asignatura

Programación Orientada a Objetos
Facultad de Informática



Juan Pavón Mestras
Dep. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense Madrid

Profesor

Juan Pavón Mestras

jpavon@fdi.ucm.es

<http://grasia.fdi.ucm.es/jpavon>

- *Tutorías*: despacho 440
Miércoles y Jueves -> 11:00 a 13:00

- O a través del campus virtual.

- Página web de la asignatura:
 - En el campus virtual de la UCM (moodle):
<https://campusvirtual.ucm.es/>

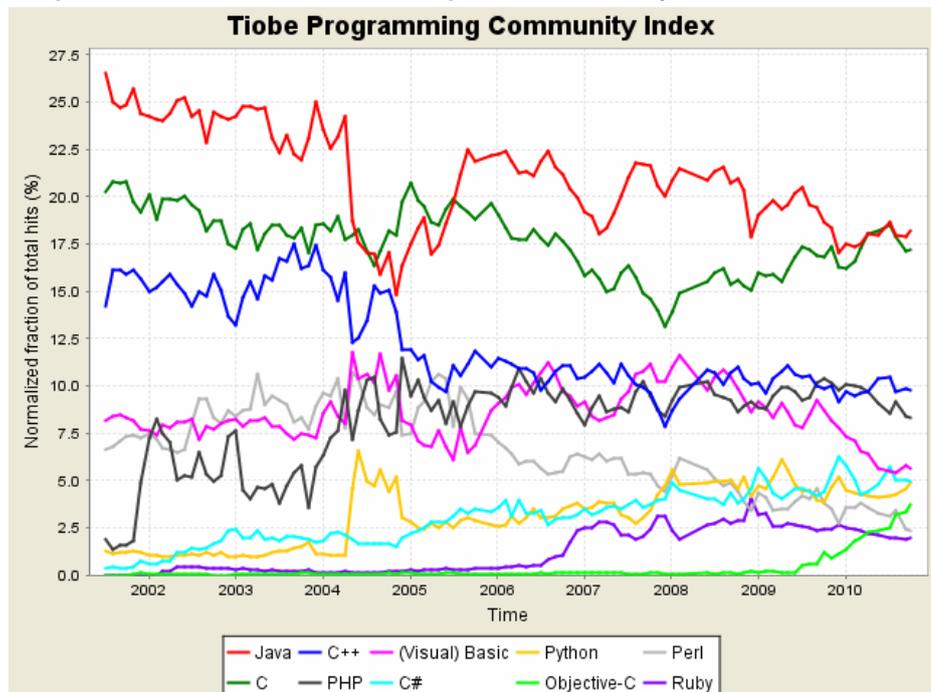
Alumnos

- Encuesta:

¿Con qué lenguaje de programación has escrito al menos un programa alguna vez?

Uso de lenguajes de programación

- <http://www.tiobe.com/tpci.htm> (sep. 2010)



Uso de lenguajes de programación

- Algunas observaciones
 - La popularidad de los lenguajes es como las olas. Tras cada ola la popularidad decrece hasta que llega la próxima.
 - Se necesitan tres olas para consolidarse
 - Los lenguajes dominantes son orientados a objetos
 - Java es indiscutiblemente el número uno
 - C++ declina lentamente
 - Suben C# y Objective-C
 - Excepciones: C, VisualBasic, PHP, PERL
 - Hay un incremento de lenguajes dinámicos
 - Inicialmente de tipo procedural (PERL) que va siendo sustituido por
 - OO: Python, Ruby (*en su primera ola con bastante fuerza*)
 - Pascal tiene poca relevancia, relegado al ámbito académico
 - En septiembre 2010 en el puesto 15 (0,771%)
 - Los lenguajes funcionales tienen una cuota bastante escasa
 - Lisp/Scheme, puesto 13, cuota 1.084%
 - Lua, en posición 24, cuota 0,556%

Primera tarea del curso

- Conectarse a <http://www.tiobe.com/tpci.htm>
 - Ver las tendencias de las distintas familias de lenguajes (OO, procedurales, funcionales, lógicos)
 - Hay también una tabla que compara la evolución de Statically Typed Languages y Dynamically Typed Languages
 - ¿Qué significa que un lenguaje sea tipado dinámico o estático?
 - ¿Y lo de que sea fuertemente tipado?
 - Identifica los tipos de los diez primeros lenguajes
 - ¿Qué lenguajes parecen tener mayor crecimiento a medio plazo?

Objetivos del curso

- Técnicas de programación orientada a objetos (POO) para la realización de aplicaciones de tamaño medio-grande
 - Aprender a pensar y diseñar con *objetos*
 - Acostumbrarse a usar *bibliotecas* de software
 - Nuevo software utilizando componentes software
 - Conocer la sintaxis y semántica del lenguaje de programación Java (y algo de C++)
 - Estudiar patrones de diseño
 - Cómo estructurar el software orientado a objetos
 - Limitaciones y alternativas de cada patrón
 - Integración de patrones de diseño
 - Armazones (frameworks)
- Esto es, con un toque de ingeniería de software...

Algunas palabrejas

Diseño dirigido por responsabilidades

herencia

encapsulado

sobrecarga

iteradores

acoplamiento

cohesión

javadoc

interface

clases

métodos

constructor

objetos

polimorfismo

Estructura del curso

- Siguiendo el método de Barnes y Kölling (**BlueJ**)
 - <http://www.bluej.org/>
- Contenidos
 - Fundamentos de POO
 - Objetos y clases. Definición de clases. Interacción de objetos. Agrupación de objetos. Bibliotecas de clases. Testing. Diseño de clases.
 - Estructura de las aplicaciones
 - Herencia. Abstracción. Gestión de errores. Patrones de diseño orientada a objetos. Ejemplos de frameworks: creación de interfaces de usuario avanzadas
 - Lenguaje Java
 - Prácticas: *Laboratorio de Programación de Sistemas*
 - Lenguaje C++
 - El lenguaje C++ se ve con más detalle en la asignatura de *Laboratorio de Programación 2*

Funcionamiento del curso

- Clases teóricas
 - Teoría: Miércoles, Jueves
 - Ejercicios: Viernes
- Ejercicios en el campus virtual
 - Trabajo individual o en grupo (usando los foros)
- Conviene desarrollar los ejemplos vistos en clase y variantes en un entorno de programación
 - BlueJ
 - Eclipse
 - Netbeans

Evaluación

- Convocatoria de febrero:
 - Durante el curso se realizarán ejercicios con calificación como parte de las clases (como máximo cinco ejercicios)
 - Cada ejercicio tendrá un peso de 10% en la nota final
 - La no realización del ejercicio equivale a calificación de cero en el mismo, sin posibilidad de recuperación
 - Al final del curso se realizará un examen final formado por un test y ejercicios
 - El peso será $(100-n*10)\%$
siendo n el número de ejercicios con calificación realizados durante el curso

- Intervenciones brillantes en clase también pueden ayudar a sumar puntos

- Convocatoria de septiembre: Examen final formado por un test y ejercicios (100%)

Bibliografía

- Básica:
 - D. J. Barnes y M. Kölling, ***Programación orientada a objetos con Java. Una introducción práctica usando BlueJ***. 3ª edición. Pearson-Prentice Hall, 2007

Bibliografía

- Complementaria:
 - T. Budd, An Introduction to Object-Oriented Programming, 3rd Edition, Addison-Wesley, 2002. (Existe versión en castellano de la primera edición).
 - E. Gamma et al., Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995. (Traducción al castellano (2002): Diseño de patrones. Pearson Educación)
 - Sánchez Allende, J., Huecas, G, Fernández Manjón, B., Moreno, P., Java 2: Iniciación y Referencia. Ed. McGraw-Hill, 2001.
 - También el libro de ejercicios
 - K. Arnold, J. Gosling, D. Holmes, El lenguaje de programación Java (tercera edición). Addison-Wesley, Pearson Educación, 2001.
 - B. Meyer, Construcción de Software Orientado a Objetos, 2ª Edición, Prentice Hall, 1999.
 - B. Stroustrup, El lenguaje de programación C++. Edición Especial, Pearson Educación, 2002.
 - Steve J. Metsker, Design Patterns Java Workbook, Addison-Wesley, 2002
 - <http://www.oozinoz.com/dpju.htm>

Más información sobre el curso

- En la página web de la asignatura:

<https://campusvirtual.ucm.es>