



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Secretaría de Postgrado

Magíster en Ingeniería de Software

Técnicas y Herramientas
Prof. Dr. Alejandro Fernández

Técnicas y Herramientas **Profesor: Dr. Alejandro Fernández**

Programa

Presentación de la materia

Técnicas y Herramientas es una materia de carácter presencial, basada en su mayoría en clases teóricas semanales (viernes 18:00 a 20:00 horas). Para el desarrollo de ciertos trabajos prácticos, se programarán clases prácticas especiales.

La materia no tiene modalidad "a distancia". Sin embargo, se permiten alumnos no presenciales si creen que pueden llevar la materia con la bibliografía de referencia. Se publicarán minutas de los temas tratados en cada clase, donde, en la medida de lo posible, se recomendará lectura de referencia.

La aprobación de la materia es con un proyecto de implementación en el lenguaje Java (cuya especificación se publicará llegada la fecha) y un examen final presencial (solo teórico, o teórico y práctico dependiendo de los resultados del proyecto).

Objetivo

El objetivo de la materia es introducir a los alumnos en técnicas modernas de la programación, en particular en aspectos de la programación visual, orientada a objetos y diseño orientado a objetos. Se enfatizará en la construcción de arquitecturas de software modulares, extensibles y reusables, conceptos claves para aplicaciones de gran porte. Se introducirá también al alumno en el uso de un lenguaje de modelado gráfico orientado a objetos (UML), que le permitirá construir diagramas especificando distintos aspectos de un sistema. Los trabajos prácticos se realizarán usando el lenguaje de modelado y diferentes lenguajes de implementación, tales como Smalltalk, Java, etc. que son los más apropiados de acuerdo a estos objetivos.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Secretaría de Postgrado

Magíster en Ingeniería de Software

Técnicas y Herramientas
Prof. Dr. Alejandro Fernández

Contenidos

1. La crisis del software. Problemas de las técnicas tradicionales (procedurales). Resolución de problemas complejos. El problema de la extensibilidad, el reuso y el mantenimiento.
2. Conceptos básicos: Tipos Abstractos de Datos. Encapsulamiento. Information hiding. Objetos y Programa Orientado a Objetos. Comportamiento de un Objeto. Mensaje y Método. Clasificación: Clases e Instancias. Instanciación. Jerarquías de Clases. Relación isA. Generalización / Especialización. Herencia, Herencia Simple. Clases Abstractas. Hacia mayor genericidad de código: polimorfismo y binding dinámico.
3. Diseño de objetos complejos. Relaciones entre Objetos. Relación de conocimiento. Relación isPartOf.
4. Lenguajes orientados a objetos: variantes. El lenguaje Java. Modelo de programación. Tipos de Mensajes. Variables de instancia. PseudoVariables: this y super. Constructores. Biblioteca de clases y jerarquías pre-definidas. El lenguaje Smalltalk: bloques y estructuras de control como objetos.
5. Estructuras de datos como Objetos. Objetos contenedores. Colecciones de Objetos. Protocolo estándar. Iteración.
6. Tipado de variables. Interfaces. Generics. Tipos primitivos vs. clases (autoboxing).
7. Lenguajes de modelado orientados a objetos: historia y variantes. El lenguaje de Modelado Unificado (Unified Modeling Language).
8. Diagramas del UML. Diagramas de Estructura Estática: Diagramas de Paquetes y Diagramas de Clases. Diagramas Dinámicos ó de Comportamiento: Diagramas de Interacción (Diagramas de Secuencia y Diagramas de Colaboración), Diagramas de Estado y de Actividades. Diagramas de Casos de Uso.
9. Aplicación de técnicas orientadas a objetos para la construcción de aplicaciones con interfaces visuales. Pattern observer. Interfaces: Model-View-Controller.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA
Secretaría de Postgrado

Magíster en Ingeniería de Software

Técnicas y Herramientas
Prof. Dr. Alejandro Fernández

Bibliografía Recomendada

Teóricos

- Timothy Budd *“An Introduction to Object-Oriented Programming”*
- J. Martin – J. Odell *“Análisis y Diseño Orientado a Objetos”*
- R.. Wirfs-Brock – B. Wilkerson – L. Wiener *“Designing Object Oriented Software”*

Prácticos

- Kathy Sierra, Bert Bates *“Head First Java”*
- David Flanagan *“Java in a Nutshell”*