

Programa Analítico

1. OBJETIVOS:

Al finalizar la asignatura los alumnos deben saber:

- el paradigma de la programación orientada a objetos
- diseñar clases
- editar, compilar y depurar programas
- aplicar adecuadamente las estructuras de programación
- modular los programas
- aplicar adecuadamente colecciones de objetos y arreglos lineales
- los métodos de ordenación, intercalación y búsqueda
- desarrollar aplicaciones informáticas simples

2. CONTENIDOS:

UNIDAD 1. Algoritmos y lenguajes programación.

- 1.1 - Concepto de algoritmo y programa.
- 1.2 - Concepto de lenguajes de máquina, ensambladores y de alto nivel.
- 1.3 - Código fuente. Edición, compilación y depuración de un código fuente escrito en un lenguaje de alto nivel.
- 1.4 - Errores de sintaxis y errores de ejecución.

UNIDAD 2. Principales conceptos de la Programación Orientada a Objetos.

- 2.1 - Objetos y clases. Creación de objetos de clases predefinidas. Instancias. Estado de un objeto. Interacción entre objetos.
- 2.2 - Utilización de métodos. Concepto de parámetros y de valores de retorno.
- 2.3 – Introducción a los tipos de datos.

UNIDAD 3. Diseño e Implementación de Clases.

- 3.1 - Componentes de una clase: atributos y métodos.
- 3.2 - Métodos constructores, modificadores y de acceso.

3.3 - Variables locales.

3.4 - Sentencia de asignación. Cálculos simples. La sentencia condicional simple (if ...else...). Sentencia de impresión.

UNIDAD 4. Tipos de datos primitivos y cadenas de caracteres

4.1 - Tipos primitivos int, double, char y boolean.

4.2 - Expresiones aritméticas, relacionales y lógicas.

4.3 - Conversión de tipo de dato.

4.4 - Diferencia entre un tipo primitivo y una cadena.

4.5 - Operaciones con cadenas.

UNIDAD 5. Abstracción y modularización.

5.1 - Diagramas de clases y de objetos.

5.2 - Objetos que crean objetos. Constructores múltiples. Variables de instancia.

5.3 - Desarrollo de aplicaciones que utilizan varias clases. Uso del depurador.

UNIDAD 6. Estructuras de decisión

6.1 - Sentencias de decisión unarias y binarias.

6.2 - Sentencias anidadas.

6.3 - Estructura if-else-if.

6.4 - Sentencia switch.

UNIDAD 7. Estructuras de repetición

7.1 - Estructura de repetición while.

7.2 - Contadores y sumadores.

7.3 - Estructuras de repetición anidadas.

7.4 - Sentencia for.

7.5 - Números aleatorios.

UNIDAD 8. Colecciones de objetos.

8.1 - Agrupar objetos en colecciones de tamaño flexible (ArrayList).

8.2 - Clases genéricas.

8.3 - Numeración dentro de las colecciones.

8.4 - Agregar y eliminar elementos de una colección.

8.5 - Objetos anónimos.

8.6 - Procesamiento de una colección: buscar un valor, buscar el mínimo o el máximo valor, mostrar los valores de una colección, etc.

8.7 - Menus de opciones.

UNIDAD 9. Arreglos lineales.

- 9.1 - Definición.
- 9.2 - Diferencias con ArrayList.
- 9.3 - Procesamiento de arreglos. Aplicaciones.
- 9.4 - Algoritmos de búsqueda, ordenación e intercalación.

3. BIBLIOGRAFIA

3.1 BASICA

Programación orientada a objetos con Java. Una introducción práctica usando BlueJ. Barnes-Kölling. Pearson . 2007

3.2 Adicional

Programación en Java 5.0. Cohoon-Davidson. Mc. Graw Hilll. 2006.

4. METODOLOGIA DE LA ENSEÑANZA

Durante las clases los alumnos deberán diseñar e implementar aplicaciones de software utilizando los conceptos de la programación orientada a objetos, así también como las distintas estructuras de programación y colecciones de objetos.

Se realizarán en cada módulo clases teóricas y actividades prácticas. Estas últimas consistirán de prácticas de resolución de problemas y/o prácticas de laboratorio según corresponda con el tema teórico.

Las guías de trabajos prácticos y apuntes de interés se publican en el portal de la Universidad.

5. CRITERIOS DE EVALUACION

Además de los exámenes parciales dispuestos por la Universidad se evaluará a los alumnos a través de pruebas semanales o quincenales y entrega de trabajos.

Para la aprobación de la cursada se tendrá en cuenta los siguientes conceptos:

- Calificación obtenida en los exámenes parciales.
- Evaluación de los trabajos prácticos realizados (individuales o grupales).

Lo anterior junto con

- El resultado de las evaluaciones breves que los profesores toman como seguimiento.
- Asistencia, puntualidad, respeto y participación en clase.

Conformarán una nota de concepto.

Para la aprobación de la materia el alumno deberá rendir un examen final que consiste en la resolución de diferentes problemas utilizando el lenguaje de programación utilizado durante el dictado del curso. La calificación final estará conformada por la nota del examen y la nota de concepto.