



Programación declarativa

Curso 2011-2012

CARÁCTER: Optativa de segundo ciclo.

CRÉDITOS: 6 créditos (3 créditos teóricos + 3 créditos prácticos en laboratorio)

HORARIO: En el primer cuatrimestre:

- Teoría: martes, de 17:30 a 19:30.
- Prácticas: lunes, de 18:30 a 20:30

PROFESOR: Roberto Rodríguez Echeverría (rre@unex.es)

TUTORÍAS: Se puede consultar en la web de la Escuela Politécnica y en la puerta de profesor. Se pueden concertar reuniones fuera del horario de tutorías. También se responden a preguntar mediante los espacios de comunicación del aula virtual.

PÁGINA WEB: <http://epcc.unex.es>, siguiendo los enlaces Asignaturas | Ingeniería Informática | Programación declarativa.

También hay un aula virtual dentro del Campus virtual de la Universidad.

OBJETIVO GENERAL:

El objetivo de esta asignatura es conocer los paradigmas de programación lógica y programación funcional y aprender a resolver problemas con ellos.

OBJETIVOS ESPECÍFICOS:

- Presentar el paradigma de programación lógica.
- Aprender a resolver problemas con el paradigma de programación lógica.
- Aprender a escribir programas en Prolog, el principal lenguaje de programación lógica.
- Presentar el paradigma de programación funcional.
- Aprender a resolver problemas con el paradigma de programación funcional.
- Aprender a escribir programas en Haskell, un lenguaje de programación funcional.
- Ser capaz de escribir informes y programas que cumplan los requisitos y plazos establecidos.

NORMAS GENERALES:

- La convocatoria oficial de los exámenes de teoría será fijada por la Subdirección Académica del Centro. Los profesores sólo fijarán la hora de comienzo de los mismos.
- De acuerdo a la normativa del Centro, para aquellos alumnos que la fecha del examen oficial coincida con la fecha del examen de otra asignatura, deberán solicitar el cambio de fecha del examen mediante instancia presentada en la secretaría del centro, en el período establecido por la misma. **No** se realizará ningún examen extra si no se ha seguido el procedimiento oficial establecido.
- Cualquier duda, caso no contemplado en este programa o problema debe ser consultado con suficiente antelación al profesor de la asignatura.

CRITERIOS DE EVALUACIÓN:

Se hará hincapié en una metodología de evaluación continua a lo largo del cuatrimestre, aunque se arbitrarán criterios de evaluación basados en un examen final.

- Se utilizan distintos instrumentos de evaluación que intentan asegurar la adquisición por parte del estudiante de todos los objetivos propuestos, tanto conocimientos como competencias.

- Esta asignatura se compone de dos grandes bloques: programación lógica (PL) y programación funcional (PF), que será necesario aprobar por separado.
- La evaluación de cada uno de los bloques se llevará a cabo mediante un portafolio de evaluación continua (PO) y la entrega, evaluación y defensa de un proyecto de programación (PY).
- Además, también habrá que realizar un informe escrito sobre un tema propuesto (INF).
- Se informará de cualquier ajuste e información adicional a estos criterios de evaluación en el aula virtual de la asignatura.

Evaluación de la parte teórica: Evaluación continua:

- A lo largo del curso se irán proponiendo distintas actividades para realizarlas en clase o fuera de ellas. Esas pruebas formarán el portafolio del estudiante. Al finalizar el bloque, se evaluará el portafolio completo con las muestras de trabajo realizado y su evolución
- Para considerar superado el portafolio de cualquiera de estos dos bloques se deberán cumplir los siguientes requisitos:
 - Haber realizado y entregado el 75 % de las pruebas.
 - Tener una nota igual o superior a 4.

Evaluación de la parte teórica: Examen final:

- Para las personas que no hayan seguido la evaluación continua o no cumplan los requisitos mínimos se realizará un examen final de cada uno de los dos bloques.
- Para considerar superado el portafolio de un bloque, la nota media del examen debe ser superior a 4.

Parte práctica:

- Para cada uno de los bloques se deberá entregar, en fechas anunciadas con antelación la solución documentada a un proyecto de programación (PY) propuesto por el profesor, cumpliendo las normas indicadas. Además, se realizará una defensa de las prácticas entregadas delante del profesor.
- Las entregas realizadas dentro de la fecha límite indicada se puntuarán de 0 a 10; la nota máxima de las prácticas entregadas posteriormente será 8 (siempre antes de la celebración del examen oficial de la convocatoria correspondiente).
- Para considerar superada la parte práctica (PY) de un bloque se deberá tener una nota mínima de 4.

Informe:

- A lo largo del curso se propondrán varios temas y habrá que elegir uno para escribir un breve informe (y las correspondientes implementaciones que puedan ser necesarias). Adicionalmente, se podrá proponer la presentación del contenido del informe al resto de los compañeros de la asignatura.
- La entrega del informe realizada dentro de la fecha límite indicada se puntuarán de 0 a 10; la nota máxima de los informes entregados posteriormente será 8 (siempre antes de la celebración del examen oficial de las convocatorias correspondientes).
- La nota de una parte superada (ya sea PO, PY o INF) se guardará hasta la convocatoria de septiembre.
- La nota final de la asignatura se calculará de la siguiente forma, siempre que cada nota individual sea igual o superior a 4:
$$\text{NotaFinal} \leftarrow 0,2 * \text{NotaPO-PL} + 0,2 * \text{NotaPY-PL} + 0,2 * \text{NotaPO-PF} + 0,2 * \text{NotaPY-PF} + 0,2 * \text{Inf}$$

Si no se cumple alguno de los requisitos mínimos en algún bloque, la nota final será SUSPENSO (3)
- La detección de la copia de cualquiera de los trabajos (en su totalidad o en parte), supondrá una nota de SUSPENSO (0) en la convocatoria, y que no se guardará ninguna de las partes superadas por separado, además de tomar las medidas disciplinarias según la normativa vigente. Se usará software de detección de copias en las prácticas e informes entregados.

PROGRAMA TEÓRICO:

Bloque 0: Introducción

Definición de lenguaje de programación ; Historia de los lenguajes de programación; Paradigmas de programación

Bloque 1: Programación lógica

Tema 1. La programación lógica y Prolog

La lógica como sistema de programación; Unificación y resolución; Cláusulas, objetivos y programas; Semántica de programas; Semántica procedural de programas; Semántica para la negación; El modelo de ejecución de Prolog

Tema 2. Introducción a Prolog

Introducción; Comparación con lenguajes de programación convencionales; átomos, predicados y hechos; Comprobación de hechos; Variables; Objetivos compuestos; Instanciación, vinculación y backtracking; Reglas; Recursividad

Tema 3. Unificación y aritmética en Prolog

Unificación; Tipos de datos; Predicados de comparación; Operadores aritméticos; Predicados aritméticos evaluables; Predicados de lectura y escritura

Tema 4. Estructuras de datos: listas

Estructuras; Listas; Operaciones con listas; Optimización de la recursividad

Tema 5. Control en Prolog

Control de la vuelta atrás: corte y fallo; Usos comunes del corte; Uso de la negación en sustitución del corte; Negación como fallo; Problemas con el corte

Tema 6. Gestión de términos y de la base de conocimiento

Clasificación de términos: Construcción y acceso a componentes de estructuras; Estructuras de datos incompletas; Gestión de la base de conocimiento

Tema 7. Otros temas de Prolog

Bloque 2: Programación funcional

Tema 1. Introducción a la programación funcional

Programación funcional; Funciones; Características de los lenguajes funcionales actuales; Funciones en los lenguajes convencionales; Lenguajes funcionales puros e impuros; Historia y evolución de los lenguajes funcionales; Ventajas e inconvenientes de los lenguajes funcionales; Definición de funciones; Especificación e implementación; Comparación de patrones (Pattern Matching); Guardas; Parcialización (currying); Funciones NO estrictas; Órdenes de evaluación. Evaluación Eager y Lazy; Funciones infijas; Funciones de orden superior

Tema 2. Tuplas y listas

Tuplas; Listas; Secuencias de números; Listas por comprensión; Cadenas de caracteres; funciones de selección de listas

Tema 3. Tipos de datos

Tipos de datos primitivos; Tipos de datos definidos por el usuario; Tipos de datos polimórficos; Chequeo de tipos; Sistema de inferencia de tipos; Clases de tipos predefinidas

Tema 4. Patrones de computación

Patrones de computación; map; foldr; foldl; filter

Tema 5. Evaluación perezosa

Evaluación perezosa; listas infinitas

Tema 6. Tipos

Tipos; Tipos enumerados; Tipos definidos por el usuario; Tipos recursivos; valuación perezosa; Árboles

Tema 7. Otros temas de Haskell



PROGRAMA DE PRÁCTICAS:

Realmente no hay una división entre clases de teoría y de prácticas en esta asignatura. En cada clase se propondrá un guión de laboratorio donde se incluirán explicaciones de conceptos teóricos y realización de problemas.

BIBLIOGRAFÍA

- W.F. Clocksin, C.S. Mellish. “Programación en Prolog”. Gustavo Gili, 1991.
- Leon Sterling, Ehud Shapiro. “The Art of Prolog: Advanced Programming Techniques, 2nd Ed.”. The MIT Press, 1994.
- R. Bird, P. Wadler. “Introduction to functional programming”. Prentice Hall, 1988.
- Richard A. O’Keefe. “The Craft of Prolog”. The MIT Press, 1990.
- Field, P. Harrison. “Functional programming”. Addison Wesley, 1988.
- J.W. Lloyd. “Foundations of Logic Programming” (2nd edition). Springer-Verlag, 1987.
- Christopher John Hogger. “Introduction to Logic Programming”. Academic Press, 1984.