

Planeacion

De: "Juan J. Carreón G." <juan.carreon@gmail.com>
Fecha: miércoles, 07 de marzo de 2012 07:16 p.m.
Para: "Ing. Alberto Templos Carbajal" <teca61@fi-b.unam.mx>; "Ángel César Govantes" <govantes@unam.mx>; "Sandoval Montaña Laura" <lsm@servidor.unam.mx>; "Alejandro Velázquez Mena" <mena@fi-b.unam.mx>; "Jorge Valeriano Assem" <assem@unam.mx>; "Maria Jaquelina López Barrientos" <jaqui@fi-b.unam.mx>; "Juan Manuel Gómez González" <juan@control.fi-b.unam.mx>; "Damian Federico Vargas Sandoval" <fevasa@fi-b.unam.mx>; <planeacion@planfi.unam.mx>; "Orlando Zamorategui Zaldívar" <zazor1@fi-b.unam.mx>; "Gilberto Silva Romo" <silvarg@unam.mx>; "Fernando Solorzano" <solojf@servidor.unam.mx>
Asunto: Re: IMPORTANTE--> REVISION DE PLANES DE ESTUDIO DE LA FI
Colegas

El 2 de marzo de 2012 13:09, Ing. Alberto Templos Carbajal <teca61@fi-b.unam.mx> escribió:

...queremos saber cómo ve el Plan de estudios en general, considera que estamos cubriendo las necesidades del mercado de trabajo; debemos agregar, reorganizar o quitar alguna asignatura, los módulos que se tienen actualmente cubren las expectativas de los estudiantes y los empleadores, etc.

Creo que el plan tiene un rezago considerable, en particular, creo que asignaturas como *Computación para Ingenieros*, y *Programación Avanzada y Métodos Numéricos*, no son tan eficaces y eficientes según los objetivos, tanto actuales, como los que se plantean para la carrera.

Llevo diez años escribiendo y publicando al respecto (con mucho gusto puedo hacer la antología respectiva), como no creo que me hubieran leído, va a continuación un resumen a vuelo de pluma de algunas ideas que he expresado al respecto en foros nacionales e internacionales:

Computación para ingenieros

Independientemente de cómo se le llame, dicho curso debe ser una **introducción a la computación y a la programación**, su objetivo principal debe ser introducir a los estudiantes a los principios de la solución sistemática de problemas mediante la programación y las reglas básicas del cómputo.

Dicho curso no debe suponer ninguna experiencia de programación previa, ya que la mayor parte de los estudiantes de primer ingreso de la FI provienen del propio sistema universitario de educación media (secundaria y preparatoria), donde he participado como presidente de la comisión dictaminadora del Colegio de Informática, a lo largo de casi una década, por lo que me consta que los programas de secundaria y preparatoria universitarios, no tienen considerado enseñar a programar a sus estudiantes.

Esa asignatura debería ser obligatoria tanto para estudiantes de Computación, como de las demás Ingenierías, cuyos comités de carrera pueden considerarlo así o no; lo cual no nos exime

de buscar lo mejor para los estudiantes tanto de nuestra carrera, como de las demás que se imparten en la FI.

Los temas principales de mi propuesta para dicha asignatura son:

- 1) Introducción
- 2) Datos atómicos y compuestos
- 3) Uniones
- 4) Uniones autorreferenciales
- 5) Expresiones locales
- 6) Recursión general
- 7) Abstracción
- 8) Recursión generativa
- 9) Acumuladores
- 10) Refinamiento iterativo

Bibliografía

1. Cómo diseñar programas, FI, UNAM, 2009

2. How to Design Programs, Second Edition, Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi, Draft, MIT Press, 2012,
<http://www.ccs.neu.edu/home/matthias/HtDP2e/index.html>

3. How to Design Classes, Draft, 2012

<http://www.ccs.neu.edu/home/matthias/htdc.html>

En el caso de *Programación Avanzada y Métodos Numéricos*, mi propuesta es que sea un curso orientado al **diseño de programas con base en clases**, así como al diseño de abstracciones que apoyen el diseño de software *reusable* y el de bibliotecas. Debe cubrir los principios del diseño de programas orientados a objetos, las reglas básicas de la evaluación de programas, así como la relación entre algoritmos y estructuras de datos, a la vez que técnicas básicas para analizar la complejidad algorítmica.

La meta es ayudar a que los estudiantes comprendan los principios del diseño de programas con base en clases, empleando lenguajes orientados a objetos, no sólo Java. El cual se empleará para aprender cómo dichos principios se emplean en aplicaciones prácticas, proporcionando así una oportunidad para discutir las fortalezas y debilidades de diversos lenguajes y paradigmas.

Ese curso debe orientarse tanto a estudiantes de Computación como de otras ingenierías, pero supone que el estudiante maneja los principios básicos del diseño de programas.

- 1) Datos, clases, uniones, introducción a Eclipse
- 2) Datos complejos, métodos condicionales, *dispatch*
- 3) Métodos para datos-complejos, Mundos, errores en verificación de tipos
- 4) Clases abstractas, constructores, excepciones, *privacy*
- 5) Igualdad, objetos función, *singleton pattern*
- 6) Datos circulares, métodos, cambios de estado; *fromStringData*
- 7) Más métodos imperativos, genéricos, diseño a interfaces, *getters, setters, predicados*.
- 8) *Traversals*, Colecciones, *Traversals* con efectos; *ArrayList*
- 9) Ciclos, ADT's, *Visitors*
- 10) *Map, HashMap*, Igualdad; *Array*
- 11) *Big Oh* y algoritmos
- 12) Más algoritmos...

Bibliografía

How to Design Classes, Viera K Proulx, 2011

<http://www.ccs.neu.edu/home/matthias/htdc.html>

How to Design Programs (Second Edition), Felleisen, Findler, Flatt, y Krishnamurthi. Draft, 2012.

Cómo Diseñar Programas, FI, UNAM, 2009

Esas dos materias cuando menos para el caso de los estudiantes de Computación deben ser obligatorias y de la máxima importancia, lo cual se debe reflejar en los créditos y los recursos asignados.

Dado que esta información se requiere para hoy, ahí va.

Atentamente

Mtro Juan J Carreón G, Profesor Titular Definitivo Sistemas Inteligentes, Ing. en Comp.
FI, UNAM