

# Enseñanza de patrones de diseño utilizando robots con alumnos de la materia Orientación a Objetos II de la Universidad Nacional de Río Negro

Mauricio Tassara, Mauro Cambarieri, Marcelo Petroff, Claudio Iturburu, Horacio Muñoz  
Laboratorio de Informática Aplicada - Universidad Nacional de Río Negro  
{mtassara, mcambarieri, mpetroff, citurburu, hmuñoz}@unrn.edu.ar

**Abstract.** Este trabajo presenta los resultados y conclusiones preliminares de una experiencia educativa en la enseñanza de patrones de diseño de la materia Orientación a Objetos II de la Licenciatura en Sistemas de la Universidad Nacional de Río Negro (UNRN) - Sede Atlántica. Los patrones de diseño revolucionaron el campo de la arquitectura de software permitiendo agilizar los procesos de producción mediante la estandarización de soluciones, sin embargo el aprendizaje resulta ser complejo para alumnos de carreras informáticas por el alto nivel de abstracción. Es por ello que se hizo uso de robots para una visualización concreta y tangible de la solución aplicada facilitando la apropiación del conocimiento adquirido por el alumnado.

**Keywords.** Patrones de Diseño, Orientación a Objetos, Enseñanza, Robótica.

## 1. Introducción

El conocimiento y dominio de los patrones de diseño son de gran importancia en el ámbito de la ingeniería de software y en particular en el diseño de software orientado a objetos. No hay una definición generalmente aceptada de un patrón, pero tal vez el mejor lugar para empezar es Christopher Alexander, una fuente de inspiración para muchos entusiastas de los patrones: “Cada patrón describe un problema que ocurre infinidad de veces en nuestro entorno, así como la solución al mismo, de tal modo que podemos utilizar esta solución un millón de veces más adelante sin tener que volver a pensarla otra vez” [1] [2]. La correcta aplicación de los patrones de diseño puede permitir alcanzar diseños de software más flexibles, modulares, reutilizables y comprensibles, objetivos muy importantes de la orientación a objetos.

Sin embargo, el aprendizaje de dichos patrones de diseño suele resultar una tarea no sencilla para los estudiantes universitarios de carreras vinculadas al desarrollo de

software. Es de vital importancia, para lograr una mejor comprensión de los mencionados patrones de diseño, la aplicación a problemas de diseño concretos, a situaciones reales en las que los estudiantes descubran los beneficios de aplicarlos durante la actividad de diseño del software. Es necesario para un completo entendimiento que se enfrenten a las dificultades propias de su implementación y que tengan la posibilidad de observar los resultados durante la ejecución de la solución obtenida.

Con el objetivo de alcanzar las metas mencionadas, se propuso a los alumnos como actividad práctica final de la asignatura Orientación a Objetos 2, correspondiente al tercer año de la Licenciatura en Sistemas de la Universidad de Río Negro, diseñar e implementar la solución a un problema vinculado a la programación de Robots, integrando y aplicando los patrones de diseño aprendidos durante el cursado de la asignatura.

La elección de la programación de robots como una herramienta para la ejercitación no es azarosa. En el artículo “Experiences With Educational Robotics” [3] se hace referencia a la importancia de la aplicación de la tecnología en los procesos de aprendizaje y en particular de los robots. “La tecnología debería promover el desarrollo de las competencias básicas y habilidades cognitivas de sus usuarios, explorando el aprendizaje en una forma interactiva y lúdica, permitiendo a la gente nuevos procesos educativos, nuevas experiencias, nuevos descubrimientos y nuevas formas de aprendizaje. Por lo tanto, el robot es atractivo como un medio, ya que invita a profesores y estudiantes a enseñar / aprender / descubrir / inventar en procesos colectivos, capaz de conectar la abstracción y el mundo concreto.”

En este trabajo se expone la experiencia educativa llevada adelante, describiendo detalladamente el proceso de diseño y desarrollo realizado por los alumnos. En la sección 2 se exponen y discuten los resultados obtenidos, detallando en la sección 3 las conclusiones de la experiencia educativa realizada y trabajos futuros.

## **2. Descripción de la experiencia educativa**

El proceso de enseñanza/aprendizaje llevado adelante durante el cursado de la asignatura Orientación a Objetos II de la Licenciatura en Sistemas de la Universidad Nacional de Río Negro (UNRN), incluyó el dictado de 9 clases por parte del docente en las que se explicaron los conceptos teóricos relacionados a los patrones de diseño, y se presentaron 10 de los patrones de diseño más ampliamente utilizados propuestos por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides (the Gang of Four, GOF) en su libro “Patrones de Diseño: Elementos de Software Orientado a Objetos Reutilizables” [4]. Se siguió un proceso de descubrimiento de los patrones de diseño, donde a partir de ejemplos prácticos, se arribó paulatinamente al diseño propuesto por cada uno de los patrones. Luego se profundizó cada patrón detallando sus aspectos particulares.

Para complementar el proceso de aprendizaje de dichos patrones, se propusieron trabajos prácticos en los que se planteaban problemas de baja complejidad a resolver aplicando los patrones de diseño aprendidos hasta el momento. Los ejercicios prácticos proponían al alumno no sólo el diseño de la solución, sino también su implementación en el lenguaje de programación Java. Si bien, no es un lenguaje de programación orientado a objetos puro, se optó por este lenguaje ya que permite aplicar los conceptos claves del mencionado paradigma y a partir del conocimiento y manejo del lenguaje por parte de todos los alumnos del curso.

En los primeros tres trabajos prácticos se indicó a los alumnos el patrón de diseño específico a aplicar en la solución del problema propuesto. Paulatinamente, a medida que los alumnos fueron madurando los conceptos y comprendiendo mejor la solución de diseño propuesta por los patrones, se dejó en manos de los alumnos descubrir el patrón de diseño más adecuado a aplicar en la solución del problema propuesto.

Para las últimas tres semanas del cursado de la asignatura, momento en el cual habían sido explicados a los alumnos la totalidad de los patrones de diseño propuestos por GOF y los alumnos habían ejercitado cada uno de ellos resolviendo problemas prácticos mostrando un nivel de madurez razonable en el manejo de los patrones de diseño, se propuso la aplicación de los patrones de diseño en el desarrollo de un software que dé solución a un problema concreto de mayor complejidad utilizando combinación de patrones. Por ello se les planteo un ejercicio integrador donde debían realizar la programación de un robot que se desplazaría por un circuito, marcado con una línea negra, y que en caso de salir del mismo emitiera una advertencia sonora.

En los siguientes párrafos se describen detalladamente el problema planteado y la solución propuesta por los alumnos.

## **2.1 Planteo del problema**

El problema propuesto a los alumnos indicaba el análisis, diseño e implementación en C++ de una solución, utilizando durante la actividad de diseño los patrones estudiados durante el cursado de la asignatura, que permita a un robot, recorrer un circuito marcado con una línea negra, y que en el caso de que el robot en su recorrido se salga del circuito, emita una advertencia sonora.

Es importante destacar que fué posible realizar esta experiencia educativa dado que los alumnos de la UNRN tienen acceso al Laboratorio de Informática Aplicada (LIA), donde cuentan con varios robots para poder experimentar.

## 2.2 Resolución del problema

### 2.2.1 Modelo de Dominio

El diseño de la solución comenzó con el diseño de un Modelo de Dominio. De acuerdo a la definición de Craig Larman [5], un modelo de dominio es una representación de las clases conceptuales del mundo real, no de componentes software. En el mismo se reflejan objetos del dominio o clases conceptuales, asociaciones entre las clases conceptuales y atributos de las clases conceptuales. A continuación se muestra el modelo de dominio definido para el problema planteado:

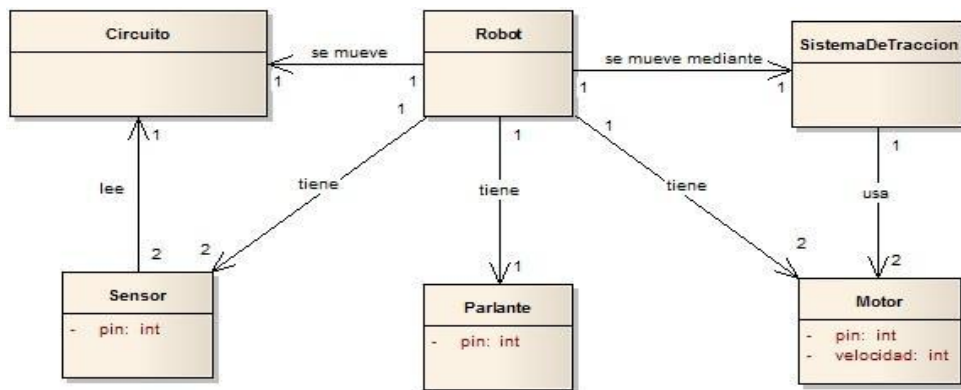


Figura 1. Modelo de Dominio

### 2.2.2 Diagrama de Clases de Diseño

A partir del modelo de dominio y de la identificación de las operaciones necesarias para dar solución al problema, se avanzó en la definición del modelo de diseño. Se identificaron las clases del diseño, y el modo en que los objetos se comunican y colaboran unos con otros.

Para formalizar este conocimiento se definieron los Diagramas de Clases de Diseño [5]. Estos diagramas muestran todas las especificaciones de las clases software y sus relaciones. En ellos podemos encontrar: clases, asociaciones y atributos; interfaces y sus operaciones; métodos; tipo y visibilidad de los atributos; navegabilidad; dependencias.

El siguiente es el diagrama de las clases principales derivadas del modelo de dominio, utilizando para su construcción el lenguaje de modelado UML [6]:

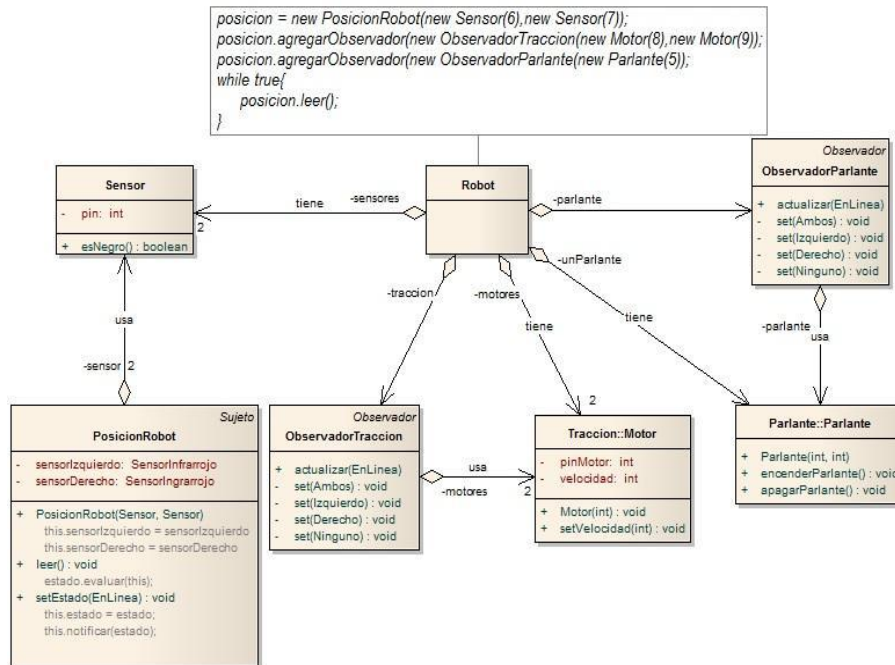


Figura 2. Modelo de Clases

El problema plantea la necesidad de poder indicar al robot que emita una advertencia sonora en caso de salirse del circuito demarcado por una línea negra. Para ello el robot cuenta con dos sensores infrarrojos, dos motores y un parlante.

Dado que las acciones del robot dependen de su ubicación con respecto al circuito demarcado se deben evaluar los valores leídos por ambos sensores. La siguiente tabla refleja en la primera fila los casos posibles del par de sensores, en la segunda fila una imagen representativa de los mismos y en la tercera fila como deben comportarse los actuadores según cada uno de los casos.

CASO 1 Sensor izquierdo y derecho sobre la línea	CASO 2 Sensor izquierdo sobre la línea y derecho no	CASO 3 Sensor derecho sobre la línea e izquierdo no	CASO 4 Ningún sensor sobre la línea
			
<b>Avanzar</b>	<b>Girar a izquierda</b>	<b>Girar a derecha</b>	<b>Retroceder y emitir sonido</b>

Figura 3. Tabla de los 4 casos posibles casos en los que pueden reflejar los sensores.

### 2.2.2.1 Diagramas de clases relacionados a los sensores

El par de sensores determinan 4 posibles estados, como se detalló en la Figura 3. Esta funcionalidad fue resuelta mediante el patrón State [7]. Los estados concretos reciben un mensaje mediante el método evaluar, de esta manera realizan una lectura de ambos sensores y de ser necesario cambian el estado. A continuación, la Figura 4, muestra el Diagrama de Clases:

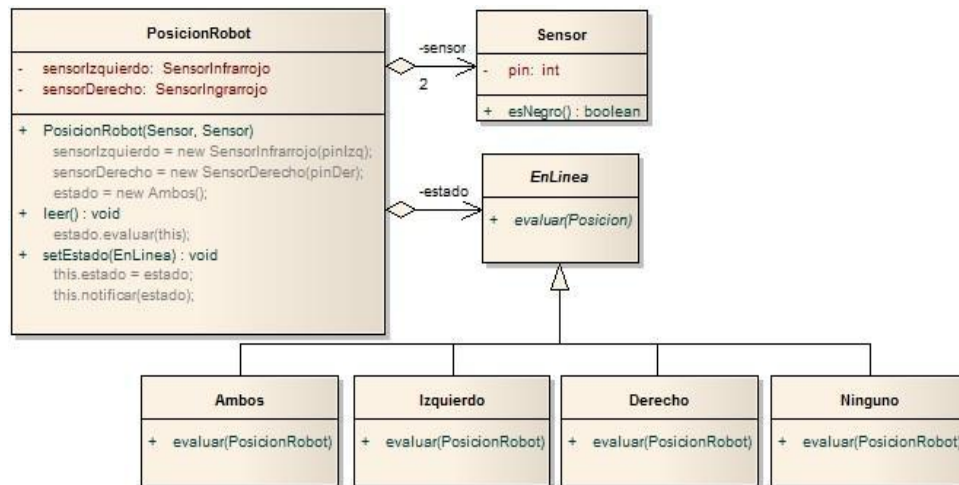


Figura 4. Diagrama de Clases - Sistema de Posicionamiento.

### 2.2.2.2 Diagramas de clases relacionados al movimiento

El robot puede realizar 4 tipos de movimientos, como se detalló en la Figura 3, esta funcionalidad fue resuelta nuevamente mediante el patrón State [7]. A continuación, la Figura 5, muestra el Diagrama de Clases:

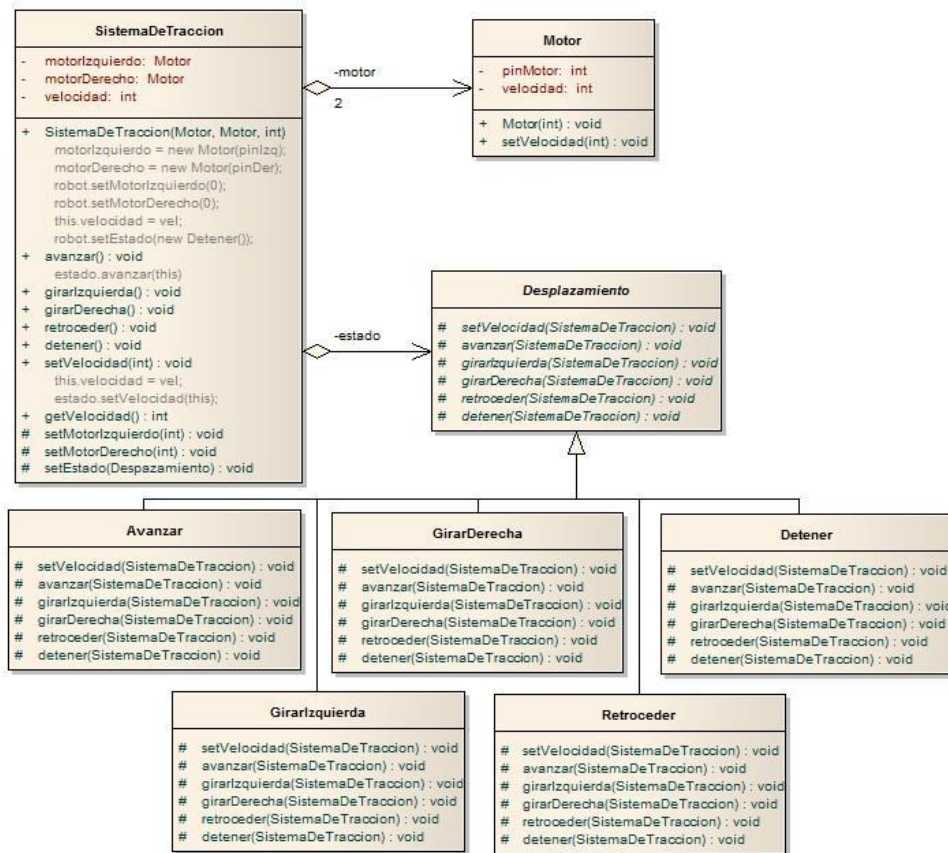


Figura 5. Diagrama de Clases - Sistema de Tracción

### 2.2.2.3 Diagramas de clases relacionados al sonido

El robot puede emitir un sonido, como se detalló en la Figura 3. Esta funcionalidad fue resuelta mediante el patrón State [7]. Los estados concretos reciben un mensaje mediante uno de dos métodos que representan el estado sonando o silenciado, de esta manera, de ser necesario el estado establece la configuración del parlante según lo requerido y cambia el estado del contexto. A continuación, la Figura 6, muestra el Diagrama de Clases:

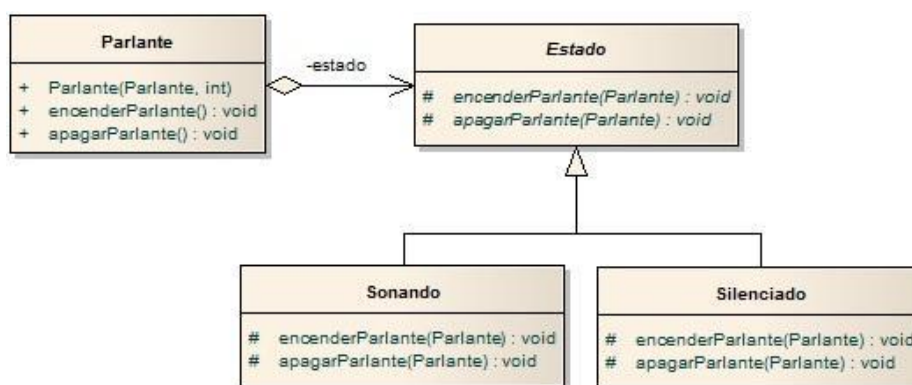


Figura 6. Diagrama de Clases - Sistema de Sonido

### 2.2.2.4 Diagramas de clases de la solución

El comportamiento de los motores y el parlante dependen del posicionamiento del robot con respecto al circuito, por ello tanto el sistema de tracción como el de sonido deben ser notificados de los cambios de estado del sistema de posicionamiento. Esta funcionalidad fue resuelta mediante el patrón Observer [7]. En este el sujeto a observar será el sistema de posicionamiento que en cada cambio de estado notificará a los observadores (sistema de tracción y de sonido) enviado como parámetro su propio estado para que los observadores actúen en consecuencia. A continuación, la Figura 7, muestra el Diagrama de Clases de la solución:



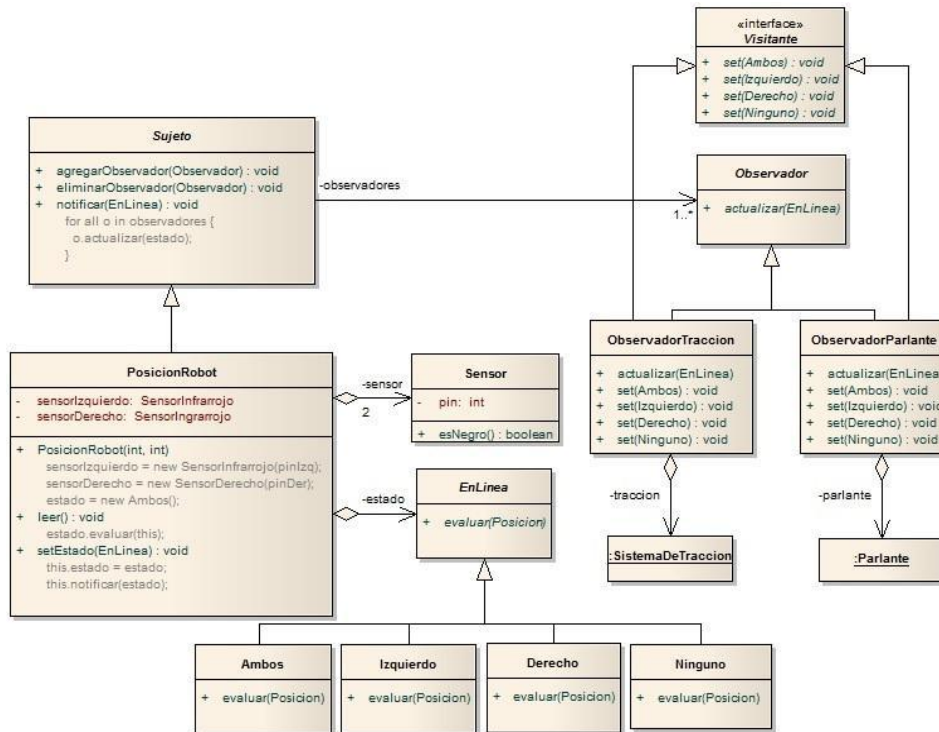


Figura 7. Diagrama de Clases - Integración de la Solución

### 3. Conclusiones y Trabajos Futuros

La experiencia educativa mostró resultados destacables desde dos puntos de vista. Por un lado, la implementación de una solución para el funcionamiento de robots permitió la visualización concreta y tangible de los resultados al observar el robot en movimiento, comportándose de acuerdo al diseño e implementación aplicada por los alumnos, logrando que pudieran relacionar de forma directa la interacción del robot en el medio con los patrones de diseños utilizados.

Experiencias previas en el dictado de la asignatura, donde no se hizo tanto hincapié en la ejercitación y la aplicación de los patrones de diseño a problemas concretos, mostraron dificultades en el aprendizaje por parte de los alumnos que se vieron reflejados en el resultado de los exámenes parciales y finales, y posteriormente en proyectos prácticos de

otras asignaturas, en los que los alumnos no utilizaron adecuadamente los patrones de diseño para alcanzar las mejores soluciones.

De un total de 14 alumnos que cursaron la asignatura durante el año 2014, 13 promocionaron con excelentes calificaciones, atribuyendo el éxito al fuerte componente de ejercitación práctica implementado entre los que se destaca la experiencia educativa integradora de la programación de los robots.

Por otro lado, la experiencia permitió que los alumnos pudieran descubrir los beneficios que se pueden obtener al aplicar adecuadamente patrones de diseño en la programación de los robots. Lograron eficientes implementaciones, factor importante teniendo en cuenta la escasa capacidad de almacenamiento y memoria de los robots y la necesidad de optimizar el uso de estos recursos. Soluciones a problemas similares sin aplicar patrones de diseño llevaban a soluciones más complejas e ineficientes.

Como trabajos futuros se prevé incorporar nuevos elementos tecnológicos vinculados a la robótica para el dictado de la materia. Se trabajará en el ajuste de la propuesta teniendo en cuenta las observaciones de los estudiantes y docentes de la cátedra.

#### **4. Referencias**

1. Alexander, et al. A Pattern Language. Oxford, (1977).
2. Martin Fowler. "Patterns of Enterprise Application Architecture", Addison-Wesley, (2002).
3. Anibal Lopes Guedes, Fernanda Lopes Guedes. Experiences With Educational Robotic. CACIC 2013. [http://sedici.unlp.edu.ar/bitstream/handle/10915/31687/Documento\\_completo.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/31687/Documento_completo.pdf?sequence=1) (Acceso 3 de Junio de 2014).
4. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional; 1 edition (November 10, 1994).
5. Craig Larman: UML y Patrones. 2da. Edición. Prentice Hall (November 2004).
6. Martin Fowler. UML Gota a Gota. Addison Wesley Longman (March 2000).
7. Stelting Maassen. "Patrones de diseño aplicados a Java", Prentice Hall, (2003).