

PARAMETRIZACIÓN

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020



San Marcos

Contenido

Introducción	2
Clases	3
Clases concretas	3
Clases abstractas	3
Clases parametrizadas	4
Mecanismos de abstracción para crear código genérico	6
Bibliotecas de clases parametrizadas	7
Clases parametrizadas de la biblioteca	8
Conclusiones y recomendaciones	10
Referencias bibliográficas	10



Introducción

El crecimiento de la demanda en creación de sistemas informáticos genera nuevos retos a los programadores y diseñadores, es común observar equipos en la búsqueda de maneras de mejorar el proceso de producción, no solo en calidad y eficiencia, si no también en la rapidez de construir rutinas que apoyen el crecimiento del proyecto.

El diseño parametrizable es un paradigma que ayuda en mucho en la fabricación de estructuras flexibles y modificables. Es un eslabón importante para la creación de código genérico. Las estructuras genéricas aportan fluidez y calidad en la creación de rutinas de software.

En la siguiente lectura se analizará diferentes elementos que ayudan tanto a la generación de código genérico como a la aplicación del concepto de parametrización, algunos de los temas contemplados son: clases concretas, abstractas y parametrizadas, mecanismos de abstracción para la creación de código genérico, bibliotecas de clases, entre otros.

Clases

Las clases son el bastión de los lenguajes de promoción que utilizan la técnica o el paradigma orientados a objetos, esta forma de desarrollar sistemas busca emular al mundo real mediante la creación de objetos y atributos. La utilización de clases también da sustento a otros conceptos importantes como: abstracción, encapsulación, la herencia y el polimorfismo.

Las clases ofrecen una representación abstracta de los objetos, por lo que permite hacer múltiples operaciones sobre ellos. Las clases pueden ser concretas o abstractas dependiendo del nivel de implementación de las funcionalidades de sus métodos.

Clases concretas

Una clase concreta es aquella que implementa todos sus métodos, puede extender una clase abstracta o implementar una interfaz. Es una clase completa que se puede instanciar a placer.

Clases abstractas

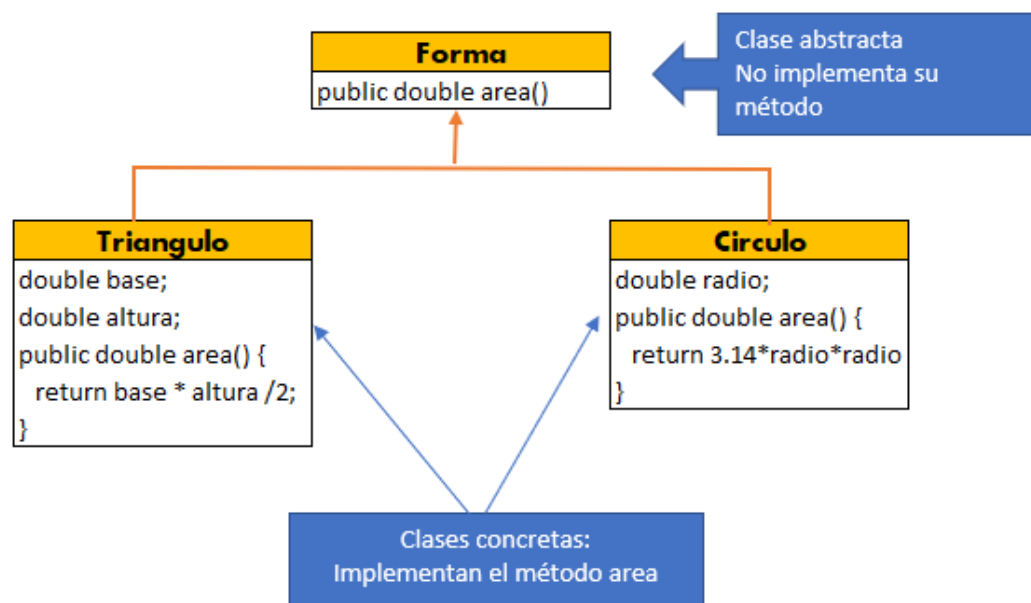
Las clases abstractas se declaran usando la palabra clave `Abstract` (por ejemplo, en Java). Por lo general, las clases abstractas, también conocidas como clases base abstractas, no se pueden instanciar (no se puede crear una instancia de esa clase). Por lo tanto, las clases abstractas solo tienen sentido si el lenguaje de programación admite la herencia.

Las clases abstractas generalmente representan un concepto abstracto o entidad con implementación parcial o nula. Por lo tanto, las clases abstractas actúan como clases principales a partir de las cuales se derivan las clases secundarias para que la clase secundaria comparta las características incompletas de la clase principal y se pueda agregar funcionalidad para

completarlas.

Las clases abstractas pueden contener métodos abstractos. Las subclases que extienden una clase abstracta pueden implementar estos métodos abstractos (heredados). Si la clase secundaria implementa todos los métodos abstractos, se convierte en una clase concreta. Pero si no es así, la clase secundaria también se convierte en una clase abstracta.

Imagen 1 representación de clases



Fuente: Elaboración propia

En la imagen anterior se demuestra la teoría de las clases vistas con anterioridad, la clase Forma es una clase de tipo abstracta que no implementa ningún método, su función es heredar a otras clases, mientras tanto las clases Triangulo y Circulo, si implementan el método área, por consiguiente, son del tipo concretas.

Clases parametrizadas

También llamadas clases genéricas, son relativamente nuevas y su objetivo es reducir errores y agregar una capa extra de abstracción sobre los tipos. Se diferencian de las demás porque sus nombres tienen un formato especial ya que incluyen los símbolos de mayor y menor.

Según (Gavidia Calderon , 2020) Los generics o clases parametrizadas son importantes ya que permiten al compilador informar de muchos errores de compilación que hasta el momento solo se descubrirían en tiempo de ejecución, al mismo tiempo permiten eliminar los cast, reducir las repeticiones y aumentar la legibilidad el código.

El cast es un procedimiento para transformar una variable primitiva de un tipo a otro, o transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas.

Los tipos genéricos permiten usar tipos para parametrizar las clases, interfaces y métodos, generando con esto muchos beneficios como:

- Permite la comprobación de tipos más fuerte en tiempo de compilación.
- Eliminación de casts.
- Mayor legibilidad del código.
- Abre la posibilidad de implementar algoritmos genéricos, con tipado seguro.

Un ejemplo de clase parametrizada es la clase ArrayList que dentro de la biblioteca de clases de Java viene de la siguiente forma: ArrayList<E> o la clase HashMap<K, V>. La información que contienen entre los símbolos de menor y mayor indican que necesitan uno o más parámetros para funcionar correctamente, de ahí su nombre de clases parametrizadas.

Es posible también crear clases parametrizadas para crear elementos como listas, colas, árboles, etc.



Mecanismos de abstracción para crear código genérico

Según (Interpolados, 2020) Una abstracción es una forma de interpretar y conceptualizar lo que resulta más importante de una entidad compleja, sin tener en cuenta todos sus detalles. Permite obtener lo esencial descartando lo que es accesorio. Abstraer es construir una idea general a partir de casos individuales. Abstracción simboliza organizar la realidad para hacerla más aprehensible, reunir la atención en las características notables al problema a resolver, y disponerlas del modo más provechoso para su mejor aplicación. Existen diferentes mecanismos para lograr este propósito:

- **Clasificación:** este es el más habitual, define un objeto como perteneciente a un tipo o clase mediante el mecanismo de instanciación. Es el que permite definir a “San José” como una provincia y a “Walter” como una persona.
- **Generalización:** ayuda a definir una clase o tipo como un caso general de varias clases o tipos. Está relacionado con el concepto de especies o familias. Por ejemplo, se puede definir cuadriláteros a partir de los paralelogramos, rectángulos y rombos. La operación inversa se llama especialización, y es la que permite definir clases como casos particulares de otras más generales.
- **Agregación:** Define a un objeto como una simple agrupación de otros. Por ejemplo, podríamos definir a un trabajo como el conjunto de trabajadores y un jefe. En esencia, representa la relación de un objeto con otros, teniendo los segundos existencia propia. Ejemplo: un objeto “libro” tiene datos de su autor, almacenados en un objeto “autor”. Sin embargo, la eliminación de un libro no implica la desaparición del objeto autor, que puede tener otros libros escritos.
- **Composición:** similar al mecanismo anterior, permite definir un objeto

como compuesto por otros objetos, de modo que si el objeto compuesto deja de existir los objetos componentes dejan de tener sentido. Es el que permite definir a una empresa como un conjunto de elementos empleados. Si la empresa deja de existir, los empleados dejan de ser tales.

Bibliotecas de clases parametrizadas

Una biblioteca de clases es una colección de plantillas de programación orientada a objetos precodificada. Tanto las aplicaciones de web como las de escritorio utilizan bibliotecas de clases. Las bibliotecas de clases contienen código para elementos de interfaz gráfica de usuario (GUI) como botones, iconos, barras de desplazamiento y ventanas, así como otros componentes que no son GUI.

Todos los lenguajes POO tienen bibliotecas de clases. Los lenguajes con .NET framework utilizan Base Class Library (BCL). Los lenguajes Java utilizan Java Class Library (JCL). Las bibliotecas de clases mejoran la reutilización del código al proporcionar implementaciones de trabajos repetitivos.

Escribir aplicaciones de programas desde cero puede ser un proceso extremadamente detallado y costoso. Las bibliotecas de clases incluyen todas las clases esenciales en un formato codificado previamente escrito, que no solo simplifica la programación, sino que también aumenta la calidad del código. La personalización de la plantilla de clase se implementa según los requisitos de programación específicos.

Las bibliotecas de clases se actualizan continuamente y se prueban con el tiempo para verificar su confiabilidad para reducir las limitaciones del lenguaje de programación. Esto es especialmente cierto en lenguajes independientes de la plataforma como Java.

Clases parametrizadas de la biblioteca

Java posee una gran variedad de bibliotecas o paquetes, suministrando con esto, una gran variedad de funcionalidades para crear nuevas aplicaciones. Además de servir como bibliotecas, definen una Application Program Interface (API), que permite al programador extender las clases de estos paquetes para adecuarlos a las necesidades primordiales de un sistema. La lógica de Java ordena los paquetes o bibliotecas en componentes jerárquicos a partir de dos directorios principales. El primero es java, que es parte esencial de lo que actualmente se conoce como el API 1 de Java y el segundo es API2,

Imagen 2 Bibliotecas básicas de Java

Paquete	Contenido
Java.applet	Clases para implementar <i>applets</i> , correspondientes a aplicaciones que corren en los browsers.
Java.awt	Clases para gráficas, componentes Graphic User Interface (GUI) y administradores de control de ventanas, además de clases más especializadas como para procesamiento de imágenes Abstract Window Toolkit (AWT).
java.beans	Clases e interfaces para construir <i>JavaBeans</i> , correspondientes a GUI independientes de plataformas.
java.io	Clases para control de entradas y salidas, tales como archivos y streams.
java.lang	Clases que componen el núcleo del lenguaje.
java.math	Clases para aritmética avanzada, incluyendo manejo de precisión numérica arbitraria.
java.net	Clases relacionadas con el manejo de redes, tales como datagramas y <i>sockets</i> .
java.rmi	Clases para el manejo de métodos remotos.
java.security	Clases para aspectos de seguridad, tales como criptografía.
java.sql	Clases para acceso a base de datos con el lenguaje Standard Query Language (SQL). ¹
java.text	Clases para internacionalización del idioma, independiente del lenguaje particular.
java.util	Clases adicionales, tales como estructuras de datos avanzadas y compresión de datos.

Fuente: <https://interpolados.wordpress.com/2017/03/07/bibliotecas-de-java/>

El API 2 de java, incluye el paquete de java y el paquete javax, este último es

donde se encuentran componentes más avanzados, como se muestra en la siguiente tabla:

Imagen 3 Bibliotecas extendidas de java

Paquete	Contenido
<code>javax.accessibility</code>	Clases que definen contratos entre componentes de interfaces de usuario y una tecnología asistente que provee acceso a esos componentes.
<code>javax.activation</code>	Clases que definen activación de los componentes de <i>JavaBeans</i> .
<code>javax.ejb</code>	Clases para el manejo de <i>Enterprise Java Beans (EJB)</i> .
<code>javax.jms</code>	Clases para el manejo de <i>Java Message Server (JMS)</i> .
<code>javax.mail</code>	Clases para el manejo de correo.
<code>javax.naming</code>	Clases para el acceso de los servicios de nombres.
<code>javax.rmi</code>	Clases para la invocación de métodos remotos incluyendo <i>CORBA</i> .
<code>javax.servlet</code>	Clases para el manejo de <i>servlets</i> y <i>Java Server Pages (JSP)</i> .
<code>javax.sql</code>	Clases para el acceso a base de datos con <i>SQL</i> .
<code>javax.swing</code>	Clases que proveen un conjunto de componentes para GUI que trabajan en cualquier plataforma.
<code>javax.transaction</code>	Clases para el manejo de transacciones entre componentes.

<https://interpolados.wordpress.com/2017/03/07/bibliotecas-de-java/>

Conclusiones y recomendaciones

- Las clases abstractas son superclases incompletas debido a que no se pueden instanciar y una característica muy importante es que tienen la capacidad de soportar polimorfismo.
- Un nuevo jugador ingreso al juego de los tipos de clase, estas son las clases parametrizadas, utilizan una lógica que estiliza la función de las clases y apoya la creación de código genérico.
- Las bibliotecas de clases tienen la particularidad de ser un contenedor muy provechoso de estructuras de código reutilizables, que son ampliamente probadas y están en constante actualización y modernización, lo que las convierten en importantes asistentes en la producción de software

Referencias bibliográficas

- Buriticá, O. T. (2017). *Lógica de programación*. Bogotá: Ediciones de la U.
- Gavidia Calderon , C. (01 de 12 de 2020). *Certified. Apuntes sobre Ingeniería de Software*. Obtenido de <http://certified-es.blogspot.com/2009/09/clases-abstractas-y-concretas.html>
- Herrera Morales, J., Gutiérrez Posada, J., & Pulgarín Giraldo, R. (2017). *Introducción a la lógica de programación*. Armenia: ELIZCOM S.A.A.
- Interpolados*. (01 de 12 de 2020). Obtenido de <https://interpolados.wordpress.com/2017/03/07/bibliotecas-de-java/>



www.usanmarcos.ac.cr

San José, Costa Rica