

DEFINIR ESTÁNDARES DE CODIFICACIÓN DE ACUERDO A PLATAFORMA DE DESARROLLO
ELEGIDA **GA7-220501096-AA1-EV02**

Aprendiz:

DIANA PACHECO MORENO

Instructor:

EDINSON RUEDA DUARTE

ANÁLISIS Y DESARROLLO DE SOFTWARE FICHA 2834937

SERVICIO NACIONAL DE APRENDIZAJE

CENTRO AGROTURÍSTICO

13/09 DE 2024

INTRODUCCION

La programación orientada a objetos es un paradigma de programación que permite modelar objetos del mundo real, sus propiedades y sus interacciones en un programa de computadora. En la actualidad, este paradigma se ha vuelto cada vez más popular debido a su capacidad para facilitar la creación de programas más complejos y escalables, lo que ha llevado a su aplicación en una amplia variedad de campos, desde el desarrollo de software hasta la inteligencia artificial y la robótica.

En la programación orientada a objetos, los objetos se modelan como entidades que tienen un estado, un comportamiento y una identidad. Estos objetos interactúan entre sí mediante el intercambio de mensajes, lo que permite la creación de programas más modulares y reutilizables. Además, la programación orientada a objetos proporciona herramientas para la abstracción, encapsulamiento, herencia y polimorfismo, que son conceptos fundamentales para la creación de programas más complejos.

Objetivo General

La aplicación del paradigma orientado a objetos tiene como objetivo principal modelar objetos del mundo real en un programa de computadora y permitir su interacción de manera más natural y sencilla.

objetivos específicos

Se encuentran la facilitación de la creación de programas más complejos, la mejora de la escalabilidad, la eficiencia en el desarrollo, y la reutilización del código.

ESTÁNDARES DE CODIFICACIÓN:

Nombramiento de variables: Una clase en java se compone por las siguientes declaraciones:

- Paquete
- Comentarios
- Definición de clases
- Atributos
- Constructores
- Métodos

Java determina en sus estándares de codificación convenciones para nombres de clases, métodos y atributos. Para las clases se define lo siguiente:

1. La primera letra debe ser mayúscula.
2. Utilizar nomenclatura camelCase (por ejemplo: un nombre compuesto por varias palabras: clase "CuentaAhorros").
3. Los nombres deben ser sustantivos.

Ejemplo: Estructura:

```
public class NombreDeLaClase {}
```

Donde la palabra ‘public’ indica que el alcance de la clase será público.

La palabra ‘class’ indica que es una clase. Y las llaves {} se usan para agrupar los bloques de código.

Todos los identificadores Java presentan unas condiciones en su nomenclatura que se resumen en los siguientes aspectos:

- Deben tener nombre claro y conciso.
- Se permite el uso del carácter subrayado (_), exceptuando la posición final e inicial del identificador No se permiten dos caracteres subrayados consecutivos.
- El nombre de la clase, Publicación debe ser un identificador válido en Java y por convención, empieza por letra mayúscula.
- Una declaración de clase crea un nombre de tipo en Java. De esta forma las referencias a objetos Publicación se pueden declarar con un simple:

Nombre de la clase:

Las variables de una clase se llaman campos. La clase Publicación tiene tres campos, idPublicacion que identifica de forma invoca cada publicación, título que es el título de la publicación y autor que ha firmado la publicación.

La creación de un objeto en Java es equivalente a decir que se está instanciando una clase. Cuando se hace la instancia se comparten los atributos y métodos de esta. Una sola clase puede tener varias instancias.

Declaración de un método:

Un método consta de un encabezado y un cuerpo. Para declarar el encabezado de un método, basta con escribir el tipo que retorna, seguido del nombre del método y entre paréntesis la lista de parámetros. Ejemplos: void imprimir (); este método no retorna nada y no tiene parámetros.

Organización de ficheros:

Las clases en Java se agrupan en paquetes. Estos paquetes se deben organizar de manera jerárquica, Dentro del paquete principal las clases se organizarán en subpaquetes en función del área. Por ejemplo, si estamos desarrollando un servicio web de inscripción a un curso de programación Java del IMFE las clases de dicho servicio se incluirían en el paquete. Un fichero consta de secciones que deben estar separadas por líneas en blanco y comentarios opcionales que identifiquen cada sección.

Deben evitarse los ficheros de gran tamaño que contengan más de 1000 líneas. En ocasiones, este tamaño excesivo provoca que la clase no encapsule un comportamiento claramente definido, albergando una gran cantidad de métodos que realizan tareas funcional o conceptualmente heterogéneas.

Fichero fuente Java (.java)

Cada fichero fuente Java debe contener una única clase o interfaz pública. El nombre del fichero tiene que coincidir con el nombre de la clase. Cuando existan varias clases privadas asociadas funcionalmente a una clase pública, podrán colocarse en el mismo fichero fuente que la clase pública. La clase pública debe estar situada en primer lugar dentro del fichero fuente.

En todo fichero fuente Java distinguimos las siguientes secciones:

- Comentarios de inicio.
- Sentencia de paquete.
- Sentencias de importación.
- Declaraciones de clases e interfaces.

CONCLUSIONES

En conclusión, la programación orientada a objetos es un paradigma de programación poderoso que permite modelar objetos del mundo real en un programa de computadora y facilita la creación de programas más complejos y escalables. En la actualidad, su aplicación se ha vuelto cada vez más popular debido a sus ventajas en la creación de programas más modulares, reutilizables y mantenibles.

BIBLIOGRAFÍA

Rumbaugh, J., Jacobson, I. & Booch, G. (2004). El Lenguaje Unificado de Modelado. Elements, 30. <http://portal.acm.org/citation.cfm?id=993859&dl=>

Corcuera, P. (2017). Introducción a la tecnología Java. Universidad de Cantabria, 7-8. https://personales.unican.es/corcuerp/java/slides/intro_tecjava.pdf