



Tarea #2

Programación II

Nombre del Profesor	Fabio Sandí Sánchez.
Nombre del Estudiante	Fabricio Mora Gómez.
Tarea	Conceptos Fundamentales de Java.
Objetivo	Reforzar los conocimientos que envuelven POO en Java.
Fecha de entrega	Lunes 30 de septiembre 2024 hasta las 5:59 pm

Instrucciones

1-Defina los siguientes conceptos:

Variables estáticas.

Métodos estáticos.

Palabra reservada "This".

Palabra reservada "Super".

Clase Abstracta.

Interfaces

Métodos Nativos.

Paquete.

2-Crear una clase,objeto y métodos referentes a la clase 2 del 23 de septiembre:

Forme una clase separada del "main" y cree cualquier objeto cualquiera con mínimo 3 atributos y dos métodos. Presentar en código

Ejemplo:

Objeto: persona

Atributos: nombre, edad, nacionalidad.

Métodos: mostrarDetalles(), saludar().

I- Defina los siguientes conceptos

- **Variables estáticas:** Son variables que pertenecen a la clase en lugar de a las instancias de la clase. Se declaran utilizando la palabra clave `static`. Estas variables son compartidas por todas las instancias de la clase.
- **Métodos estáticos:** Son métodos que también pertenecen a la clase en lugar de a una instancia específica. Se pueden llamar sin crear un objeto de la clase y se declaran con la palabra clave `static`. No pueden acceder a variables de instancia directamente.
- **Palabra reservada “This”:** Se refiere al objeto actual de la clase. Se utiliza principalmente para diferenciar entre los atributos de instancia y los parámetros del método cuando tienen el mismo nombre.
- **Palabra reservada “Super”:** Se usa para hacer referencia a la clase padre o superclase. Permite acceder a los métodos y constructores de la clase padre desde una clase hija.
- **Clase Abstracta:** Es una clase que no puede ser instanciada directamente y que puede contener métodos abstractos (sin implementación) y métodos concretos (con implementación). Se utiliza como base para que otras clases la hereden y completen su funcionalidad.
- **Interfaces:** Es un contrato que define un conjunto de métodos que una clase debe implementar. A diferencia de las clases abstractas, las interfaces no pueden contener implementación de métodos (hasta Java 8, donde se introdujeron métodos por defecto).
- **Métodos Nativos:** Son métodos que están implementados en un lenguaje de programación diferente, como C o C++, y se invocan desde el código Java mediante la palabra clave `native`. Estos métodos interactúan con el sistema operativo o hardware.
- **Paquete:** Es una forma de organizar las clases y las interfaces en módulos separados. Se declaran al inicio de un archivo de código y ayudan a evitar conflictos de nombres entre clases.

2- Crear una clase, objeto y métodos

Profe, decidí hacerlo sobre un juego llamado Valorant, el cual tiene distintos agentes con habilidades y una habilidad definitiva (ultimate) que requiere cargar puntos para poder utilizarse. En este documento se encuentra la descripción, pero el código estará en su respectiva carpeta.

Clase: Agente (representa un agente de *Valorant*).

Atributos:

- **nombre:** El nombre del agente (ejemplo. "Jett").
- **rol:** El rol del agente en el equipo (ejemplo. "Duelista").
- **habilidadUltimaCarga:** La cantidad de puntos que tiene cargada su habilidad definitiva, con un máximo de 7.

Métodos:

- **mostrarDetalles():** Muestra la información básica del agente.
- **usarUltima():** Intenta usar la habilidad definitiva del agente si está completamente cargada.
- **cargarUltima():** Añade puntos a la habilidad última del agente hasta que esté completamente cargada.

Bibliografía

CampusMVP. (n.d.). *Paquetes en Java: ¿Qué son, para qué se utilizan y cómo se usan?* CampusMVP. <https://www.campusmvp.es/recursos/post/paquetes-en-java-que-son-para-que-se-utilizan-y-como-se-usan.aspx>

freeCodeCamp. (2022, 12 de abril). *Interfaces en Java explicadas con ejemplos*. freeCodeCamp. <https://www.freecodecamp.org/espanol/news/interfaces-java-explicadas-con-ejemplos/>

Instituto Tecnológico de La Piedad. (n.d.). *Métodos nativos en Java*. Instituto Tecnológico de La Piedad. <http://www.itlp.edu.mx/web/java/Tutorial%20de%20Java/Cap1/nativo.html>

Martínez, J. (n.d.). *Java: Constructores y super*. Arquitectura Java. <https://www.arquitecturajava.com/java-constructores-y-super/>

Platzi. (n.d.). *Super y this en Java*. Platzi. <https://platzi.com/clases/1629-java-ooop/21575-super-y-this/#:~:text=La%20clase%20padre%20se%20denomina,en%20la%20que%20se%20utiliza>