**GUÍA DE CLASES Y OBJETOS**

|  |  |
| --- | --- |
| **OBJETIVO DE APRENDIZAJE** | **OA 4**. Construir aplicaciones computacionales basadas en programación orientada a objetos, de manera de cumplir con las exigencias técnicas y de los usuarios. |
| **APRENDIZAJE ESPERADO** | **AE 5.** Construye programas, utilizando los componentes del paradigma orientado a objetos, a partir del análisis de problemáticas y requerimientos, modelando el comportamiento del sistema. |
| **CRITERIOS DE EVALUACIÓN** | **5.1.** Construye los componentes a utilizar en el programa a partir del análisis de requerimientos para resolución de problema, resguardando los principios de POO. |
| **OBJETIVOS DE APRENDIZAJE GENÉRICOS** | **A – B – C – G - H** |

**¿QUÉ ES UN PARADIGMA DE PROGRAMACIÓN?**

Hace referencia a una manera o estilo de programación de software. Existen diversas maneras de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores. Por ejemplo, existe el paradigma imperativo, declarativo, programación orientada a objetos, programación reactiva, entre otros.

Respecto a la programación orientada a objetos (**POO**), esta tiene como objetivo el desarrollo computacional a través de la interrelación de una colección de objetos que trabajan conjuntamente para resolver un problema (UNAL, 2017). Es un tipo de paradigma de programación muy popular en la actualidad (IEEE Spectrum, 2020).





**A continuación, se abordarán en mayor profundidad, los conceptos de objeto, clase e instancia.**

**OBJETO**

La idea fundamental en **POO** es empaquetar en una unidad tanto los datos como las funciones que operan (las que manipulan esos datos). Este empaquetamiento ocurre en un objeto.

Un objeto es un conjunto de **atributos** y **métodos** relacionados entre sí. Es algo tangible y juega un papel clave en el dominio del problema de programación. Por ejemplo, un objeto puede ser el auto que vemos a continuación.



Imagen extraída de Quaternius. (2018).

**CLASE**

Una clase es un modelo o molde de varios objetos del mismo tipo. Es una plantilla que define variables y métodos comunes para todos los objetos de cierto tipo. En el caso del objeto auto mencionado anteriormente, una clase podría contener estos elementos:

|  |
| --- |
| AUTO |
| String Marca |
| String Modelo |
| Int Año |
| Double Peso |
| Boolean Tuning |

En Java, para la definición de una clase se utiliza la palabra reservada ***class***, seguida del nombre de la clase. A continuación, veremos otro ejemplo en donde se crea una clase Bebé incluyendo sus atributos y métodos.

public class Bebe  
{  
 *//Atributos de la Clase Bebe*  
 String nombre;  
 boolean esMujer;  
 double peso;  
 double decibeles;  
 int numPopo = 0;  
  
 *//Métodos de la Clase Bebe*  
 void popo()  
 {  
 numPopo += 1;  
 System.out.println("¡Que alguien me cambie el pañal!");  
 }  
}

**INSTANCIA**

Ya definida la clase, es posible crear objetos a partir de esta **¡como si fuera un molde!** Esta parte del proceso se llama **INSTANCIAR** la clase. Cada uno de los objetos o instancias, tienen su propia copia de los atributos y métodos, por lo tanto, la asignación de valores es totalmente independiente de las demás.

Para instanciar un objeto, en Java se utiliza el operador ***new*** seguido del nombre de la clase y un par de paréntesis.

public static void main(String[] args)  
 {  
 *//Instanciación de una clase*  
 Bebe chucky = new Bebe();  
  
 *//¡chucky se hizo popó!*  
 chucky.popo();  
 }

**BUENAS PRÁCTICAS**

1. Por lo general las clases siempre son un sustantivo y comienzan con mayúsculas. Si contiene más de una palabra, cada una de ellas comienza con mayúscula (Ejemplo: StringBuffer, Perro, Cuenta). ¡Las clases deben ir sin tilde!
2. Para llevar un orden en el código, cada clase debe estar alojada en un archivo con el mismo nombre de la clase. En el caso de la clase Bebe, su archivo será Bebe.java.

**¿TE ATREVES A PROBAR POO? COMPLETA ESTE DESAFÍO**

Haz una lista de diversos tipos de objetos que logres identificar a tu alrededor. Anótalos en el cuadro a continuación.

1. ¿Cuáles son sus características? ¿Qué acciones o comportamientos tiene cada objeto?

**OBJETOS**

**CARACTERÍSTICAS / ACCIONES O COMPORTAMIENTOS**

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Ponte en este caso. Has sido contratado(a) por un hotel para perros para crear un programa que gestione las reservas de habitaciones.

1. Nombra **DOS CLASES** que debería tener el programa.
2. Nombra **DOS ATRIBUTOS** que debería tener la clase.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**¿AHORA SÍ? VEAMOS CÓMO TE VA CON POO RESOLVIENDO ESTE DESAFÍO O RETO**

Sigue los pasos:

* + - 1. **¿Cuál es la idea general del Reto?**

**R:** El paradigma de la programación orientada a objetos.

* + - 1. **¿Cuál es nuestra pregunta esencial para responder al reto?**

**R:** ¿Es posible construir videojuegos usando esta forma de programar?

* + - 1. **¿Cuál es nuestro reto o desafío?**

**R:** Desarrollador de un videojuego INDIE (ver diapositiva 30 en adelante).

* + - 1. **¿Qué preguntas, actividades y recursos necesitamos para resolver el reto?**

**R:** Realicen preguntas entre ustedes que les ayude a resolver el reto y generen actividades y/o recursos para representar el conocimiento necesario para desarrollar exitosamente la solución. Pueden apoyarse de la primera parte de esta guía así como también de estas instrucciones:

1. Sitúense en este caso. Trabajan para una empresa de videojuegos y el equipo está muy ansioso en desarrollar el próximo gran éxito para móviles. Para eso deberán:
2. Crear un equipo de trabajo de 3 personas. Cada integrante del equipo estará encargado de tareas específicas del trabajo:

* Guardián(a): Encargado del cumplimiento de las etapas de la actividad.
* Hechicero(a): Presenta la solución del equipo.
* Mercante: Encargado(a) de intercambiar información con otros grupos. ***(recuerda que eres un negociador(a), y tienes que entregar información valiosa si deseas recibir lo que necesitas).***

1. En equipos, seleccionar uno de estos juegos:

* <http://flappybird.io/>
* <http://terrycavanaghgames.com/hexagon/>
* <http://www.mathsisfun.com/games/sokoban.html>

1. Escoger uno e identificar al menos 3 objetos, incluyendo sus características y mecánicas **(COMPORTAMIENTOS).**
   * + 1. **¿Cuáles son las posibles soluciones?**

**R:** Luego de analizar y seleccionar uno de los juegos propuestos, detallen cuáles son las posibles soluciones para el reto encomendado y presenten las condiciones del entregable. A continuación se detallan las instrucciones:

1. Crear al menos 3 clases, declarando sus atributos y métodos. Consideraciones:

* Para el método solo basta crear un Println() que detalle la mecánica. Por ejemplo:

System.out.println("Al apretar Espacio la nave dispara");

* ¿Qué acción le agregarías al juego? ¿Cómo esta acción se traduce en el lenguaje orientado a objetos?
* **Bonus:** Investiga cómo se programa alguno de esos juegos y entrega un pseudo-código.

1. Entregable



1. Consideraciones









* + - 1. **¿Cuáles son los mecanismos de validación de lo efectuado?**

**R:** Creen un método que les permita validar la solución encontrada. En esta etapa trabajen en conjunto con el/la docente. Pueden utilizar los instrumentos de evaluación disponibles y autoevalúen sus desempeños como una oportunidad mejora y autorregulación de los aprendizajes.

* + - 1. **¿Cómo documentamos o publicamos nuestra solución?**

**R:** Para presentar o publicar la solución al reto deben realizar lo siguiente:











**MATERIAL DE CONSULTA**

1. Codigofacilito. Programación orientada a objetos explicada: <https://www.youtube.com/watch?v=Nka4JSBgf7I&feature=emb_title>
2. Programación ATS. Curso completo de programación en Java: <https://www.youtube.com/watch?v=2ZXiuh0rg3M&list=PLWtYZ2ejMVJkjOuTCzIk61j7XKfpIR74K>
3. TutorialesYa. Curso completo de programación en java: <http://www.tutorialesprogramacionya.com/javaya/>
4. La Geekipedia de Ernesto. Curso de Java desde cero:  
   <https://www.youtube.com/watch?v=L1oMLsiMusQ&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB>
5. Java desde cero: <https://javadesdecero.es/>
6. w3schools: <https://www.w3schools.com/java/default.asp>
7. CodingBat: <https://codingbat.com/java>
8. Learn Java: <https://www.learnjavaonline.org/>
9. CodeGym: <https://codegym.cc/>
10. Programming Skills: <http://www.pskills.org/corejava.jsp>
11. Tutorials Point: <https://www.tutorialspoint.com/java/java_generics.htm>
12. Java Tests: <https://www.studytonight.com/tests/?subject=java>
13. JavatPoint: <https://www.javatpoint.com/core-java-quiz>
14. Recopilación de ejercicios Java POO: <https://www.academia.edu/36182478/Recopilaci%C3%B3n_de_Ejercicios_Java_POO_Programaci%C3%B3n_Orientada_a_Objetos>
15. Curso de programación orientada a objetos: <https://www.curriculumnacional.cl/614/w3-article-209545.html>
16. Curso de desarrollo de videojuegos: <https://www.curriculumnacional.cl/614/w3-article-209538.html>
17. Curso de desarrollo de aplicaciones móviles: <https://www.curriculumnacional.cl/614/w3-article-209536.html>
18. Makigas. Tutorial de GIT: <https://www.youtube.com/watch?v=jSJ8xhKtfP4&list=PLTd5ehIj0goMCnj6V5NdzSIHBgrIXckGU>

**REFERENCIAS**

1. Quaternius. (2018). Low-poly Cars. itch.io. Recuperado 20 de agosto de 2020, de <https://quaternius.itch.io/lowpoly-cars>
2. MIT. (2010). Introduction to Programming in Java, Lecture 4: Classes and Objects. Recuperado 20 de agosto de 2020, de https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/lecture-notes/MIT6\_092IAP10\_lec04.pdf
3. UNAL. (2017). Paradigmas de programación. Recuperado 26 de agosto de 2020, de <https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoria/index.html>
4. IEEE Spectrum. (2020). Interactive: The Top Programming Languages. Recuperado 20 de agosto de 2020, de https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2019