Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

MinTIC

Unidad 2

05 - Convenciones de código

Hechos QUE CONECTAN

Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

MinTIC

Convenciones de código

Hechos QUE CONECTAN

Clases

- **La primer letra debe ser mayúscula**
- **Utiliza nomenclatura** UpperCamelCase
- **Para las** clases**, los nombres deben de ser sustantivos (Sujeto) y van después de la palabra reservada *class***
- **Para las** interfaces**, los nombres deben de ser adjetivos (Califica el sustantivo) y van después de la palabra reservada *interface***
- **Ejemplos**
  - **class ClasePrincipal**
  - **interface ActionListener**

Paquetes

- Deben ser escritos todo en minúscula.
- Van después de la palabra reservada *package*
- Si se van a usar paquetes dentro de otros paquetes, se unen mediante un punto (.)
- Finalizan con punto y coma (;)
- Ejemplos
  - package ventanas;
  - package imagenes.iconos;

Métodos

- **La primer letra debe ser** minúscula
- **Utiliza nomenclatura** lowerCamelCase
- **Los nombres deben conformarse por el par *verbo + sustantivo***
- **El nombre va después del tipo de método (void, int, double, String)**
- **Al finalizar el nombre del método debe indicarse mediante paréntesis con o sin argumentos ()**
- **Ejemplos**
  - **int sumaEnteros(int a, int b)**
  - **boolean retornaPermisos(int tipoUsuario)**

Variables

- La primer letra debe ser minúscula
- Utiliza nomenclatura lowerCamelCase
- El nombre va después del tipo de dato (int, String, double, boolean)
- Es recomendable utilizar nombres con un *significado explícito*, y en lo posible, cortos
- Defina cada variable en una línea independiente.
- Ejemplos
  - int edad
  - String nombre
  - String direccionResidencia

Constantes

- Todas las letras de cada palabra deben estar en mayúsculas
- Se separa cada palabra con un guión bajo (_)
- Se declaran similar a las variables, con la diferencia de que el tipo de dato va después de la palabra reservada final.
- Defina cada variable en una línea independiente.
- Ejemplos
  - final Double PI
  - final String CODIGO_CIUDAD

Estilos de código

- Se deben emplear cuatro espacios como unidad de indentación.
- Evite las líneas de más de 80 caracteres ya que no son interpretadas correctamente por muchas terminales y herramientas.
- Cuando una expresión no entre en una línea debe separarla de acuerdo con los siguientes principios:
  - Romper después de una coma.
  - Romper antes de un operador.
  - Preferir roturas de alto nivel (más a la derecha que el «padre») que de bajo nivel (más a la izquierda que el «padre»).
  - Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.
  - Si las reglas anteriores provocan que su código parezca confuso o que éste se acumule en el margen derecho

```
unMetodo(expresionLarga1, expresionLarga2, expresionLarga3,
         expresionLarga4, expresionLarga5);

var = unMetodo1(expresionLarga1,
                unMetodo2(expresionLarga2,
                          expresionLarga3));
```

Sentencias

- **Simples**: Cada línea debe contener como máximo una sentencia
- **Compuestas**: son sentencias que contienen listas de sentencias encerradas entre llaves «{ sentencias }».
  - Las sentencias encerradas deben indentarse un nivel más que la sentencia compuesta.
  - La llave de apertura se debe poner al final de la línea que comienza la sentencia compuesta; la llave de cierre debe empezar una nueva línea y ser indentada al mismo nivel que el principio de la sentencia compuesta.
  - Las llaves se usan en todas las sentencias, incluso las simples, cuando forman parte de una estructura de control, como en las sentencias if-else o for. Esto hace más sencillo añadir sentencias sin incluir accidentalmente bugs por olvidar las llaves.
- **Retorno**: Un return con un valor no debe usar paréntesis a no ser que que hagan el valor de retorno más obvio de alguna manera.

```
argv++;              // Correcto
argc--;              // Correcto
argv++; argc--;      // EVITAR!
```

```
if (condicion) {
    sentencia;
} else if (condicion) {
    sentencia;
} else {
    sentencia;
}
```

```
for (inicializacion; condicion; actualizacion) {
    sentencias;
}
```

Universidad Tecnológica de Pereira

Mision TIC 2022

El futuro digital es de todos

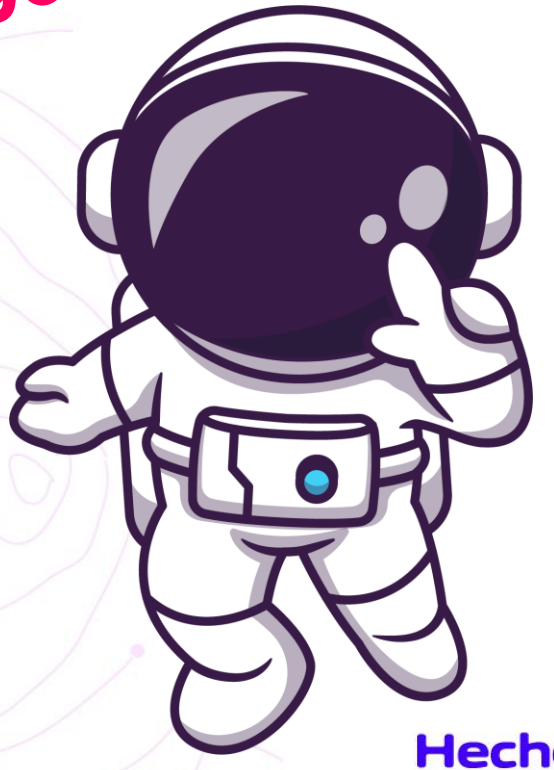
MinTIC

Ejercicios

Hechos QUE CONECTAN

Vamos al código

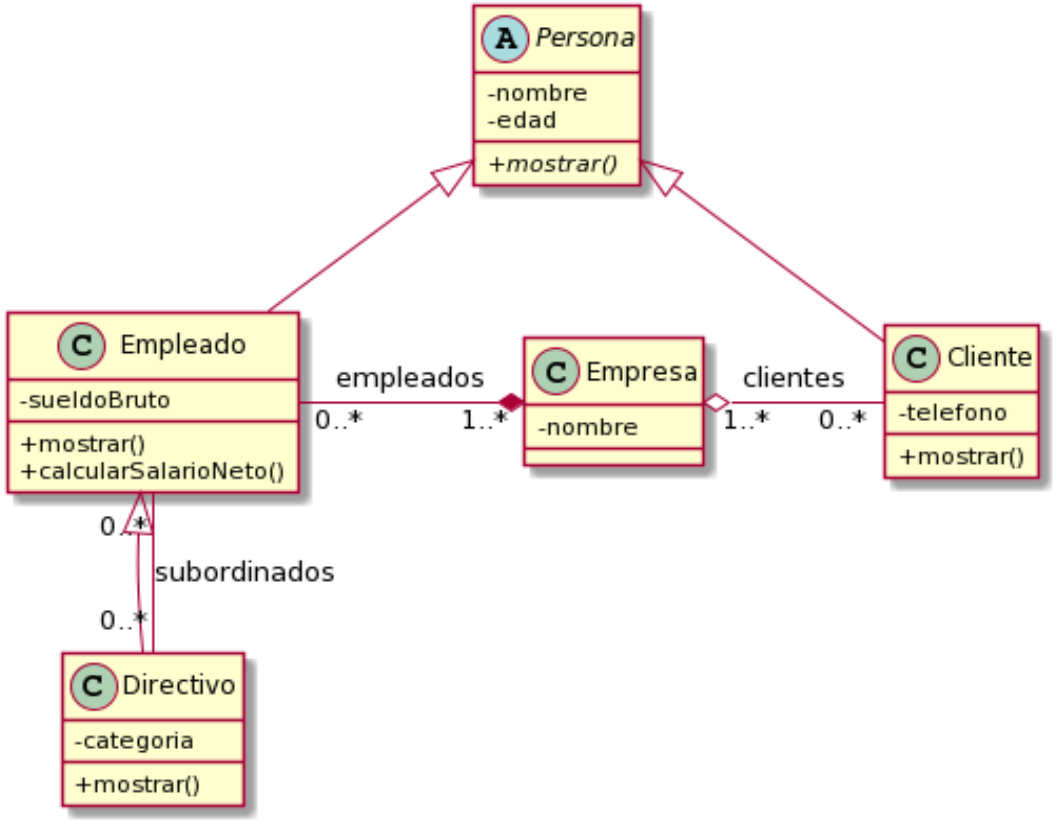
- Crear un proyecto Maven para los ejercicios de la clase 5
  - Ctrl + Shift + P
  - *› Java: Create Java Project...*
  - *Maven*
  - *maven-archetype-quickstart, ‹versión más reciente›*
  - group Id: co.edu.utp.misiontic2022.c2
  - artefact Id: clase04-empleados
- Crear las clases en Java del proyecto asignado



Empleados de una empresa

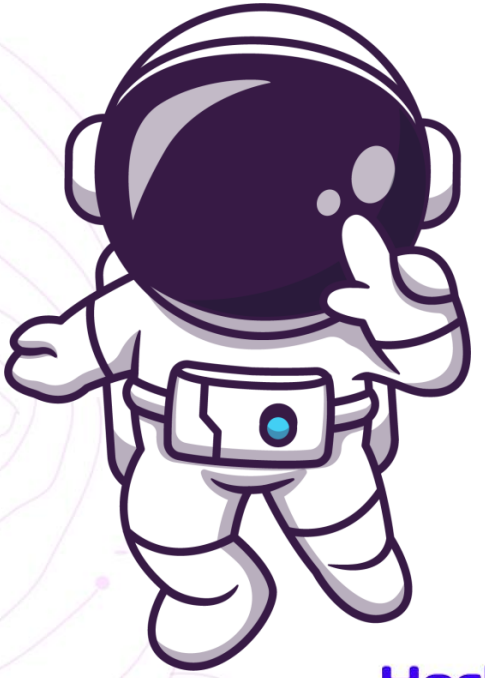
- Una aplicación necesita almacenar información sobre empresas, sus empleados y sus clientes. Ambos se caracterizan por su nombre y edad.
- Los empleados tienen un sueldo bruto, los empleados que son directivos tienen una categoría, así como un conjunto de empleados subordinados.
- De los clientes además se necesita conocer su teléfono de contacto.
- La aplicación necesita mostrar los datos de empleados y clientes

Solución Empleados



Vamos al código

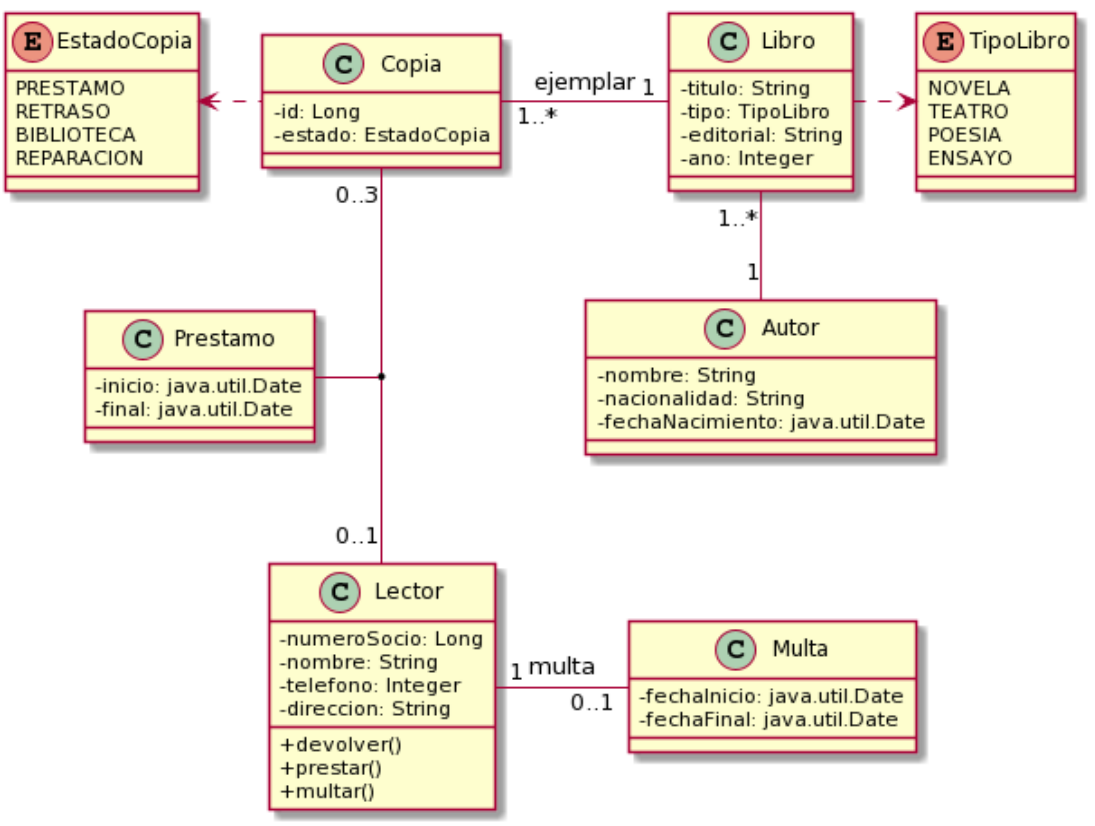
- Crear un proyecto Maven para los ejercicios de la clase 5
  - Ctrl + Shift + P
  - *› Java: Create Java Project...*
  - *Maven*
  - *maven-archetype-quickstart, ‹versión más reciente›*
  - group Id: co.edu.utp.misiontic2022.c2
  - artefact Id: clase04-biblioteca
- Crear las clases en Java del proyecto asignado



Gestión de Biblioteca

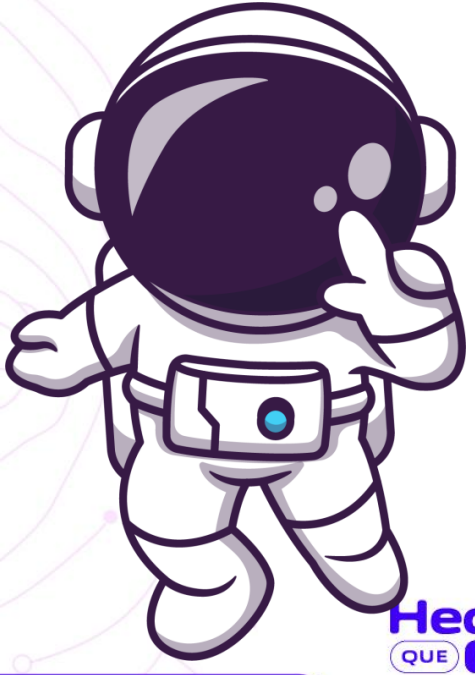
- Una biblioteca tiene copias de libros. Estos últimos se caracterizan por su nombre, tipo (novela, teatro, poesía, ensayo), editorial, año y autor.
- Los autores se caracterizan por su nombre, nacionalidad y fecha de nacimiento.
- Cada copia tiene un identificador, y puede estar en la biblioteca, prestada, con retraso o en reparación.
- Los lectores pueden tener un máximo de 3 libros en préstamo.
- Cada libro se presta un máximo de 30 días, por cada día de retraso, se impone una "multa" de dos días sin posibilidad de coger un nuevo libro.
- Realiza un diagrama de clases y añade los métodos necesarios para realizar el préstamo y devolución de libros.

Solución Biblioteca



Vamos al código

- Crear un proyecto Maven para los ejercicios de la clase 5
  - Ctrl + Shift + P
  - *› Java: Create Java Project...*
  - *Maven*
  - *maven-archetype-quickstart, ‹versión más reciente›*
  - group Id: co.edu.utp.misiontic2022.c2
  - artefact Id: clase04-proyecto
- Crear el diagrama UML y las clases en Java del proyecto



Hechos QUE CONECTAN

Proyecto

Un estudio de arquitectura desea crear una base de datos para gestionar sus proyectos. Nos dan las siguientes especificaciones:

- Cada proyecto tiene un código y un nombre. Un proyecto tiene uno y solo un jefe de proyecto y un jefe de proyecto sólo puede estar involucrado en un proyecto o en ninguno.
- De cada jefe de proyecto se desean recoger sus datos personales (código, nombre, dirección y teléfono). Un jefe de proyecto se identifica por un código. No hay dos nombres de jefe de proyecto con el mismo nombre.
- Un proyecto se compone de una serie de planos, pero éstos se quieren guardar de modo independiente al proyecto. Es decir, si en un momento dado se dejara de trabajar en un proyecto, se desea mantener la información de los planos asociados.
- De los planos se desea guardar su número de identificación, la fecha de entrega, los arquitectos que trabajan en él y un dibujo del plano general con información acerca del número de figuras que contiene.
- Los planos tienen figuras. De cada figura se desea conocer, el identificador, el nombre, el color, el área y el perímetro. Además, de los polígonos se desea conocer el número de líneas que tienen, además de las líneas que lo forman. El perímetro se desea que sea un método diferido; el área se desea implementarlo como genérico para cualquier tipo de figura, pero además se desea un método específico para el cálculo del perímetro de los polígonos.
- De cada líneas que forma parte de un polígono se desea conocer el punto de origen y el de fin (según sus coordenadas, X e Y), así como la longitud. Cada línea tiene un identificador que permite diferenciarlo del resto. La longitud de la línea se puede calcular a partir de sus puntos origen y final.

Para la próxima sesión...

- Terminar los ejercicios que no se terminaron... (si aplica)
- Ver los videos del Curso Video JUnit 5