EJERCICIO PRACTICO DE PROGRAMACION ORIENTADA A OBJETOS
2º ASIR C 27/12/2021 - IMPLANTACIÓN DE APLICACIONES WEB

Se pide la clase **Hipoteca.php**, de manera que se valida con el código **pruebaHipoteca.php**,

Hipoteca.php | pruebaHipoteca.php

Source | History

```php
<?php
require './Hipoteca.php';
echo "ATENCION:<br>".
     Hipoteca::MENSAJE1.'</br>';
echo Hipoteca::MENSAJE2;
echo "<hr>";

echo "hipoteca con VALORES MALOS<br>";
$h=new Hipoteca(-10000,25 , 1);
echo $h->ficha();
echo "<hr>";

echo "HIPOTECA CORRECTA";
$h=new Hipoteca(12000, 7, 5);
echo $h->ficha();
echo '<br>';
echo "<b>CUADRO AMORTIZACION</b>";
echo $h->TablaAmortizacion();
```

dando el siguiente resultado:

Navigator

- Hipoteca
  - MENSAJE1 'Los valores son informativos. El banco pasa!'
  - MENSAJE2 'Datos calculados al día actual.'
  - $maximoAnualidades
  - $minimoAnualidades
  - $tipoMinimo
  - $tipoMáximo
  - $capital00
  - $tipoInteres00
  - $anualidades00
  - __construct($capital, $tipoInteres, $anualidades)
  - getMaximoAnualidades()
  - getMinimoAnualidades()
  - getTipoMinimo()
  - getTipoMáximo()
  - setMaximoAnualidades($maximoAnualidades)
  - setMinimoAnualidades($minimoAnualidades)
  - setTipoMinimo($tipoMinimo)
  - setTipoMáximo($tipoMáximo)
  - getCapital()
  - getTipoInteres()
  - getAnualidades()
  - getTipoInteresReal()
  - getTipoInteresPorcentaje()
  - setCapital($capital)
  - setTipoInteres($tipoInteres)
  - setAnualidades($anualidades)
  - getCuotaAnualidad()
  - getTotalPagado()
  - getInteresTotal()
  - ficha()
  - calculaTablaAmortizacion()
  - TablaAmortizacion()
  - pintarTabla($tablaEntrada):string

ATENCION:
Los valores son informativos. El banco pasa!
Datos calculados al día actual.

hipoteca con VALORES MALOS

| | |
|---|---|
| CAPITAL FINANCIADO: | 0 |
| TIPO INTERES: | 15% |
| ANUALIDADES: | 30 |
| **CUOTA ANUAL:** | **0** |
| **TOTAL PAGADO:** | **0€** |
| INTERESES TOTALES: | 0 |

HIPOTECA CORRECTA

| | |
|---|---|
| CAPITAL FINANCIADO: | 12000 |
| TIPO INTERES: | 7% |
| ANUALIDADES: | 5 |
| **CUOTA ANUAL:** | **2926.69** |
| **TOTAL PAGADO:** | **14633€** |
| INTERESES TOTALES: | 2633 |

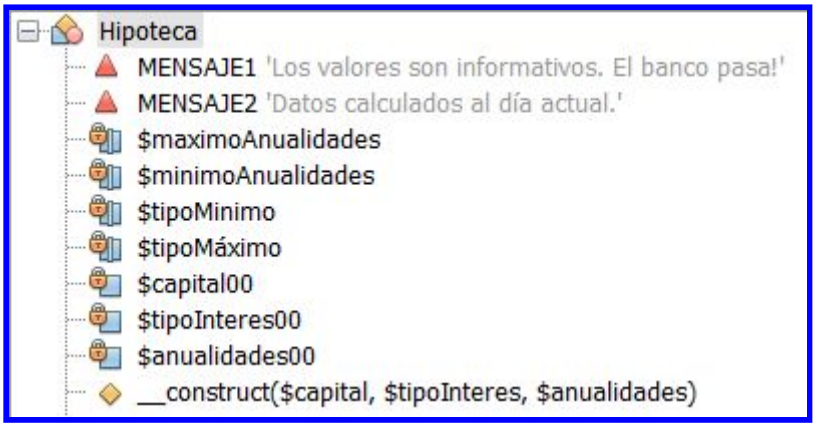
**CUADRO AMORTIZACION**

| AÑO | INTERES | AMORTIZ. | C.VIVO |
|---|---|---|---|
| 0 | 0 | 0 | 12000 |
| 1 | 840 | 2086.69 | 9913.31 |
| 2 | 693.93 | 2232.76 | 7680.56 |
| 3 | 537.64 | 2389.05 | 5291.51 |
| 4 | 370.41 | 2556.28 | 2735.22 |
| 5 | 191.47 | 2735.22 | 0 |

EJERCICIO PRACTICO DE PROGRAMACION ORIENTADA A OBJETOS
2º ASIR C 27/12/2021 - IMPLANTACIÓN DE APLICACIONES WEB

Especificaciones de la clase / ayuda de codificación:

- Constantes (texto en imagen)
- Campos de clase
- Campos de objeto (OJO! añadir el código asignado al final del nombre del campo, en el ejemplo es 00)
- Constructor.




```php
class Hipoteca {

//constante de mensajes informativos
    CONST MENSAJE1 = 'Los valores son informativos. El banco pasa!';
    CONST MENSAJE2 = 'Datos calculados al día actual.';

//campos de clase con el maximo y minimo de Anualidades, inicializados a 30 y 10
    private static $maximoAnualidades = 30;
    private static $minimoAnualidades = 10;
//campos de clase con tipos de interes minimo y maximo, inicializados a 1 y 15
    private static $tipoMinimo = 1;
    private static $tipoMáximo = 15;

//campos privados de objeto de la clase hipoteca se le añade el código
//de alumno/a asignado al final del nombre de campo (en este caso es 00)
//debéis sustituir el 00 por el código que se os ha asignado.
    private $capital00,
            $tipoInteres00, //se recibe en % (p.e. 5.5 %)
            $anualidades00;

//constructor que recibe el valores de los campos privados
//obligatorio usar los setters
    function __construct($capital, $tipoInteres, $anualidades) {...5 lines }
```

AQUI VAN LAS CONSTANTES

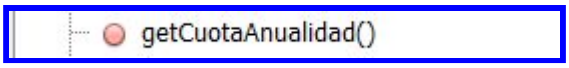
CAMPOS DE CLASE

CAMPOS DE CLASE

EJERCICIO PRACTICO DE PROGRAMACION ORIENTADA A OBJETOS
2º ASIR C 27/12/2021 - IMPLANTACIÓN DE APLICACIONES WEB

- SETTERS Y GETTERS AUTOMÁTICOS DE CLASE
- GETTERS AUTOMATICOS DE OBJETO
- getTipoInteresReal(), getTipoInteresPorcentaje()
- y setters de objeto (3) con restricciones

```
35  //SETTERS Y GETTERS AUTOMATICOS CLASE (8)
70
71  //GETTERS AUTOMATICOS DE OBJETO (3)
86
87      //devuelve el tipo de interes numérico (de 5.5 -> 0.055)
88      function getTipoInteresReal() {...3 lines }
91
92      //devuelve una cadena con el tipo y % (de 5.5 ->"5.5%")
93      function getTipoInteresPorcentaje(){...3 lines }
96
97      //establece el capital, si es menor que 0 lo pone a 0
98      function setCapital($capital) {...6 lines }
104
105     //establece el tipoInteres, tiene que estar entre el mínimo y el máximo
106     //en caso contrario lo pone al máximo
107     function setTipoInteres($tipoInteres) {...6 lines }
113
114     //establece las anualidades, tiene que estar entre el mínimo y el máximo
115     //en caso contrario lo pone al máximo
116     function setAnualidades($anualidades) {...6 lines }
```




este método se os dá tal y como hace falta incluirlo en la clase, solo tenéis que copiar y pegar en la clase:

```
//calcula la cuota anual en función de los datos de la hipoteca (NO ROUND)
 //NO MODIFICAR ESTE CÓDIGO
 function getCuotaAnualidad() {
    // C0 = CF * (i * (1+i)^t) /( (1+i)^t-1)
    $CF = $this->getCapital(); //Capital Finaciado
    $i = $this->getTipoInteresReal(); //interes real (ya dividido por 100)
    $t = $this->getAnualidades(); //anualidades que se quieren
    $bicho = pow(1 + $i, $t);
    $C0 = $CF * ($bicho * $i) / ($bicho - 1);
    return $C0;
 }
```

métodos getTotalPagado(), getInteresTotal() y método ficha().

```php
//devuelve el número de anualidades  * cuota anual (2 decimales)
    function getTotalPagado() {...3 lines }

//devuele la cuota anual para pagar (2 decimales), para ello se suma el capital
//inicial y el total de intereses y se divide por las anualidades
    function getInteresTotal() {...3 lines }

//devuelve una tabla html con la ficha de la hipoteca
    function ficha(){...9 lines }
```

método calculaTablaAmortizacion() y TablaAmortizacion()

```php
//vamos almacenando en una tabla de dos dimensiones
//cada fila es una anualidad, y por cada fila hay cuatro columnas
//1º columna indica la anualidad (0 a anualidades)
//2ª columna lo que pagamos de $interes
//3º columna lo que pagamos de $amortizacion
//4º columna lo que nos queda de $capitalVivo
    function calculaTablaAmortizacion() {
//inicialmente el $capitalVivo es el capital del objeto
        $capitalVivo = $this->getCapital();
//ponemos la cabecera de la tabla
        $tabla[]=['AÑO','INTERES','AMORTIZ.','C.VIVO'];
        $tabla[] = [0,0, 0, $capitalVivo];
//lanzamos el bucle desde 0 hasta las anualidades
        for ($fila = 0; $fila < $this->getAnualidades(); $fila++) {
//calculamos los $intereses de ese $capitalVivo
            $intereses = $capitalVivo * $this->getTipoInteresReal();
//los restamos a la cuota anual, y eso será la $amortizacion
            $amortizacion = $this->getCuotaAnualidad() - $intereses;
//recalculamos el $capitalVivo, le restamos la $amortización
            $capitalVivo = $capitalVivo - $amortizacion;
//almacenamos la $tabla[$fila] con las cuatro columnas
            $tabla[] = [$fila+1,
                round($intereses,2),
                round($amortizacion,2),
                round($capitalVivo,2)];
        }
        return $tabla;
    }
```

```php
//devuelve un HTML con el cuadro de amortizacion
//NO MODIFICAR ESTE CODIGO
    public function TablaAmortizacion() {
        return self::pintarTabla($this->calculaTablaAmortizacion());
    }
```

Por último este es el código del método privado de clase pintarTabla($tablaEntrada):

```
//se le pasa una tabla de dos dimensiones y devuelve un
// HTML listo para hacer echo (hecho en el primer trimestre
// NO MODIFICAR ESTE CODIGO
    private static function pintarTabla($tablaEntrada) {
        $tablaHTML = "<table>";
        for ($f = 0; $f < count($tablaEntrada); $f++) {
            $tablaHTML = $tablaHTML . "<tr>";
            for ($c = 0; $c < count($tablaEntrada[$f]); $c++) {
                $tablaHTML = $tablaHTML . '<td style="border: 1px solid black">'
                        . $tablaEntrada[$f][$c] .
                        "</td>";
            }
            $tablaHTML = $tablaHTML . "</tr>";
        }
        $tablaHTML = $tablaHTML . "</table>";
        return $tablaHTML;
    }
```