Práctica 2

Objetivos:

- Familiarizarse con los procesos de compilación y ejecución de programas
- Practicar con la creación de objetos y la invocación de sus métodos

Parte 1: compilar y ejecutar un programa desde la línea de comando

Parte 2: compilar y usar una clase desde `bluej`

Parte 3: crear un programa `main` que crea un objeto y usa sus métodos

Parte avanzada: Usar una clase externa para hacer una gráfica

Parte 1: compilar y ejecutar un programa desde la línea de comando

Compilar y ejecutar la clase `Hola` de los apuntes desde el intérprete de órdenes

Hacer una captura de pantalla que muestre la compilación y la ejecución, y meterla en el informe

Parte 2: usar una clase desde `bluej`

En la parte 2 usaremos una clase ya hecha, `Disolucion`, que estudia el fenómeno de la *difusión:*

- la distribución espacial de moléculas de un *soluto* en un medio *disolvente*.

Por simplicidad se estudia la difusión en un medio *unidimensional*

Ver el documento `ejercicio-disolucion.pdf` para una explicación más en detalle

Diseño de la clase Disolucion

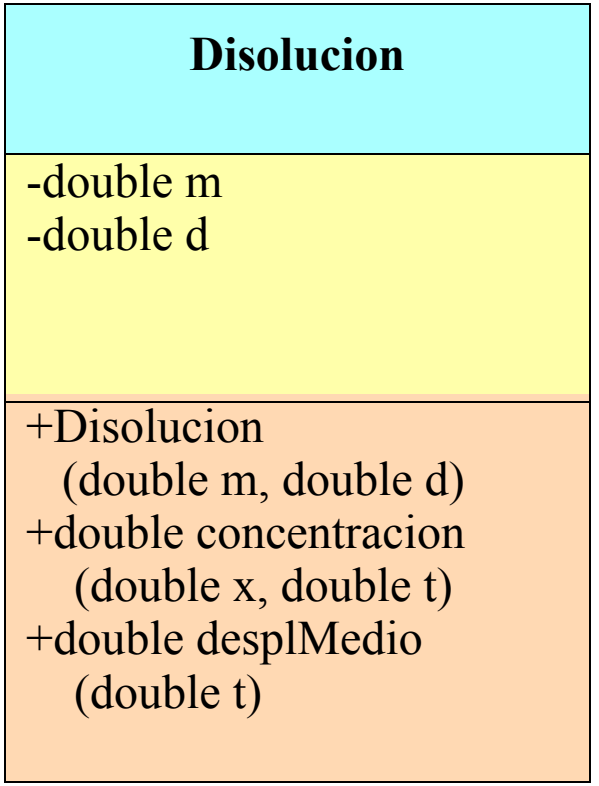
Descripción: Representaremos el sistema soluto-medio con objetos de la clase `Disolucion`

Atributos:

- masa: `m` (gr)
- coeficiente de disolución: `d` (cm^2/s)

Métodos:

- *constructor*: recibe como parámetros los valores iniciales de los atributos (masa y coeficiente de disolución) y los copia en los respectivos atributos



Diseño de la clase Disolucion (cont.)

- `concentracion(x,t)`: se le pasan como parámetros el desplazamiento y el tiempo. Retorna la concentración, en gr/cm
- `desplMedio(t)`: se le pasa como parámetro el tiempo. Retorna el desplazamiento medio, en cm

Parte 2: compilar y usar una clase desde `bluej`

Crear en `bluej` un nuevo proyecto:

- Mover a él y compilar la clase `Disolucion` que se suministra
  - ver explicación en el documento `ejercicio-disolucion.pdf`
- Con el ratón crear un objeto de la clase `Disolucion` con `m`=5gr y `d`=10^{-5}cm^2/s
  - este último valor corresponde a disolver alcohol en aire
- Hacer dos capturas de pantalla que muestren el resultado de usar el método `concentracion()`
  - una para `x`=0.5cm y `t`=0s
  - otra para `x`=0.5cm y `t`=36000s
- Y una 3ª captura para el método `desplMedio`()
  - en `t`=72000s

Parte 3: crear un programa `main` que crea un objeto y usa sus métodos

Crear en el proyecto una nueva clase con un método `main` que haga algo similar a lo que hemos hecho con el ratón en `bluej`:

- crear un objeto de la clase `Disolucion`
- mostrar en pantalla los resultados de invocar el método `concentracion()`
  - una para `x`=0.5cm y `t`=0s
  - otra para `x`=0.5cm y `t`=36000s
- y mostrar en pantalla los resultados del método `desplMedio`(), en `t`=72000s

Crear y usar objetos

Crear objetos de una clase:

- Sintaxis

```
Clase nombreObjeto=new Clase(parámetros);
// los parámetros son los del constructor
```

- Ejemplo, para una clase llamada `Piedra`

```
Piedra p=new Piedra(0.0,100.0,30.0,20.0);
// los parámetros indican los datos iniciales
```

Invocar un método del objeto:

- Sintaxis

```
nombreObjeto.nombreMétodo(parámetros);
// los parámetros son los que requiera el método
```

- Ejemplo

```
p.avanzaTiempo(0.5);
// invoca a avanzaTiempo() para un valor 0.5
```

Mostrar resultados en pantalla

Java tiene una operación de *concatenación* que permite unir a un string otro string o un dato primitivo cualquiera

- el operador de concatenación es `'+'`
- el resultado es otro string
- ejemplo, si `x` es una variable numérica:

```
"el valor de x es: "+x
```

Esto nos permite mostrar resultados en pantalla con la instrucción `System.out.println`. Ejemplo:

```
System.out.println(“desplazamiento= ”+x+” cm”);
```

Práctica 2: Parte avanzada

Crear una gráfica de la evolución del desplazamiento medio. Para ello:

Crear una nueva clase con otro `main` que haga:

- crea un objeto de la clase `Grafica`
- crea un objeto de la clase `Disolucion`
- crea una variable para el tiempo, `t`, con valor inicial `0.0`
- repite 10 veces (copiando y pegando instrucciones, pues aún no hemos aprendido los bucles):
  - inserta en la gráfica el tiempo `t` y el desplazamiento medio para ese tiempo
  - añade a `t` `100` segundos
- pinta la gráfica

Hacer una captura de la gráfica para el informe

Hacer una gráfica sencilla

Métodos de la clase `Grafica`, del paquete `fundamentos`

new Grafica (String titulo, String tituloX, String tituloY)	Constructor que pone los títulos de la ventana y de los ejes X e Y
void inserta (double x,double y)	Inserta el punto (x,y) en la gráfica actual
void pinta()	Pinta la gráficas

Ejemplo:

```
Grafica g = new Grafica ("Titulo","ejeX(unidades)","ejeY(unidades)");
g.inserta(x1,y1);
g.inserta(x2,y2);
...
g.pinta();
```

Nota: importar el paquete `fundamentos`:

```
import fundamentos.*; // al principio de la clase
```

Paquete fundamentos

Para usar la clase `Grafica` es necesario instalar el paquete `fundamentos`

Se puede *descargar* de la página moodle

- sección de *Recursos*

Luego hay que instalarlo en *bluej*

- ver apuntes del *capítulo 7*

Práctica 3: Entregar

Parte obligatoria

- La captura de pantalla de la parte 1
- Las tres capturas de pantalla de la parte 2
- El código de la clase principal de la parte 3
- La captura de pantalla de la parte 3

Parte avanzada, si se ha hecho

- El código de la nueva clase principal
- Una captura de pantalla con la gráfica