

PROGRAMACION -PRACTICA 5

Objetivos

- uso de colecciones (tema 8)
- uso de clases predefinidas (tema 8)

Recursos

Proyecto eclipse que contiene, entre otros ficheros y carpetas:

- Paquete `es.upm.dit.prog.practica5`
- `PruebaPractica5.java` programa que prueba si funcionan bien las clases
- `PruebaInteractiva5.java` programa que recibe órdenes por teclado para probar el funcionamiento.

Descripción

Partiendo de la práctica anterior, sustituiremos los arrays por Colecciones de capacidad ilimitada, y las operaciones de manejo de objetos por los métodos de las Colecciones, con lo que se simplificará el código enormemente. Estos elementos se tratan en el tema 8 de la asignatura.

Creación del proyecto

1. Descargue de moodle el fichero zip que encontrará en la entrada Práctica 4: proyecto para empezar. En Eclipse debe hacer `File->Import->General->Existing Projects into Workspace` e indicar la localización del fichero zip usando la opción `Select archive file`. El fichero se llama `PROG24practica4` y contiene varios archivos; dentro de la carpeta `src` se encuentra el paquete `es.upm.dit.prog.practica4` que es donde deberán estar las nuevas clases que desarrolle.
2. Copie las clases `Posicion`, `Dron`, `Mision`, `CentroControl` y los `SelectorMision` de la práctica 4 en el paquete `es.upm.dit.prog.practica5`.

Programación

1. En la clase `CentroControl` elimine los atributos y métodos que tenía, y ponga sólo dos atributos (será necesario importar `List` y `ArrayList` de `java.util`):

```
private List<Dron> drones;  
private List<Mision> misiones;
```
2. Añada un método constructor que crea las listas inicialmente vacías con `ArrayList`. Genere el método `String toString()` que pone eclipse.
3. Escriba un método `public void addDron(Dron d)`, que llama al método `this.drones.add(d)` si `d` es diferente de `null`.
4. Escriba un método `public void addMision(Mision m)`, que llama al método `this.misiones.add(m)` si `m` es diferente de `null`.

5. Escriba un método `public List<Dron> getDrones()` que devuelve una nueva lista con los contenidos de `this.drones`. Para ello puede llamar al constructor de `ArrayList` pasando otro como parámetro.
6. Escriba un método `public List<Mision> getMisiones(SelectorMision sm)` que devuelve una nueva `List<Mision>` compuesta por todas las de `this.misiones` que cumplen la condición del `SelectorMision`. Si el `SelectorMision` que se pasa como parámetro es `null`, use un `SelectorMisionTrue`.
7. Escriba el método `public void update(long t)`, que llama a `mover(t)` con todos los drones y a `update(t)` con todas las misiones.

Pruebas

1. Cuando haya terminado de programar las clases, puede pasar a probar su funcionamiento.
2. Si no tiene errores de compilación, puede ejecutar las pruebas empaquetadas que hay en `PruebaPractica5` (marque este fichero y ejecute el programa: `Run->Run Java application.`). Se crean objetos de las clases que estamos probando, se llama a sus métodos y se comprueba que los efectos y resultados son los correctos. No entregue la práctica hasta haber superado con éxito todas estas pruebas. Deberá leer los mensajes para comprobar que todo funciona como se espera. Este programa proporciona una nota provisional aproximada.
3. Otra forma de probar que funcionan bien es mediante el programa `PruebaInteractiva5`; para ejecutarlo marque el fichero y haga: `Run->Run Java application`. En la consola de eclipse aparecerá un mensaje de saludo, y un cursor para que pueda escribir alguna de estas órdenes; el resultado se mostrará en la siguiente línea, con el cursor preparado para la siguiente orden:

ORDEN	FUNCIONAMIENTO
hello	saluda
status	muestra los valores de las variables
help	muestra la lista de órdenes
exit	acaba el programa
clear	inicia las variables
pos x y z	crea <code>this.pos</code> que es un vector posicion con <code>x:double y:double z:double</code>
vel x y z	crea <code>this.vel</code> que es un vector velocidad con <code>x:double y:double z:double</code>
dron id t control	crea <code>this.dron1</code> nuevo con <code>id:String this.pos t:long this.vel control:boolean</code> , guarda el anterior en <code>this.dron2</code>
addvel	añade vel a <code>this.dron1</code>
mover1 t	mueve el <code>this.dron1</code> con <code>t:long</code>
mover2 t	mueve el <code>this.dron2</code> con <code>t:long</code>

peligro	comprueba si this.dron1 está en peligro por this.dron2
mision id n	crea una misión con id:String, this.dron1 y n posiciones
addpos t	añade this.pos y y t:long a this.mision
update t	actualiza this.mision con t:long
show	muestra un diagrama con los drones de la estacion filtrados con el último selector
adddron	añade this.dron1 a this.cc
getdrones	obtiene drones no nulos de this.cc
addmision	añade this.mision1 a this.cc
getmisiones	obtiene misiones no nulas de this.cc
updatecc t	actualiza todos los elementos de this.cc con t:long
sim tini tfin	muestra una simulación del movimiento de los drones de la estacion con: tini:long tfin:long use los botones para parar, avanzar t+1 o continuar la simulación: > >>
seltrue	devuelve todas las misiones
selactivas	devuelve las misiones activas
selalejadas d	devuelve las misiones con drones a mayor distancia de d:double de this.pos
selaterrizando	muestra las misiones con drones aterrizando
seldespegando	muestra las misiones con drones despegando
selenpeligro	muestra las misiones con drones en peligro

Puede copiar las órdenes del fichero comandos5.txt y comprobar los resultados.

Entrega y evaluación

No entregue la práctica hasta haber superado con éxito todas estas pruebas. Aunque el programa PruebaPractica5 proporciona una nota aproximada, si aparece el siguiente mensaje o falta alguna clase o interfaz, su código no es correcto por lo que cuando se ejecute en el servidor para la corrección final aparecerán errores en la compilación y la nota será 0.0.

Error en las cabeceras de métodos de CentroControl, Posicion, Dron o Mision

No se considerará entregada la práctica hasta que no la suba a moodle -dentro del plazo indicado-; para hacer la entrega de la práctica a través de moodle debe:

1. Con el ratón sobre el proyecto PR0G24practica5, en Eclipse seleccione el menú: File->Export...->General->Archive File (también puede aparecer en español como Fichero zip)
2. Seleccione la carpeta src de su proyecto, y allí los ficheros Java; en "To Archive file" indique la carpeta de su sistema de ficheros donde quiere colocar el fichero zip y el nombre de este fichero que debe ser

PROG24practica5.zip. Asegúrese de que se guarda en formato zip con las carpetas necesarias.

3. Fuera de Eclipse, localice el fichero PROG24practica5.zip y súbalo a la tarea Práctica 5 del Moodle de Programación.

Cuando termine el plazo de entrega de la práctica, se recogerán y comprobarán todas las entregas. Se asignará la nota 0.0 a las entregas que no se hayan recibido a tiempo, que no contengan los ficheros indicados o que no compilen correctamente. Para cada una se pasarán pruebas parecidas a las de la nota provisional. Las notas definitivas aparecerán en la entrada de moodle un tiempo después del cierre de la entrega, cuando los profesores corrijan todas las prácticas y revisen que no hay copias de código.

Se recuerda que la copia de código supone el suspenso automático de la práctica y de la asignatura, tanto para quien copia el código como para quien lo cede sin distinción. Además, se emprenderán medidas disciplinarias contra dichos alumnos ante la Escuela y la Universidad.

Si ha llegado hasta este punto, le doy la enhorabuena por haber finalizado las prácticas de programación del curso 2023-24. Si le han gustado, le agradecería que lo dijera enviándome un mensaje a mi dirección de correo.