Tema: III.
Programació dirigida per esdeveniments
Pràctiques

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant

lsi Department of Software and Computing Systems

Contingut

Requisits

- Crea un directori anomenat "`hada-p2`".
- Dins d'ell crea l'arxiu que contindrà tot el codi font demanat: "`hada-p2.vala`".
- Pots crear el teu propi programa principal de prova en un arxiu anomenat "`main.vala`". No has de lliurar-ho.
- Pots compilar el codi així: "`valac hada-p2.vala main.vala`", això generarà un executable anomenat "`hada-p2`".
- El lliurament consistirà en un fitxer anomenat "`hada-p2.tgz`" que contindrà aquest directori i no ocuparà mes de 256KB.

Exercici 1

Crea en un arxiu anomenat "`hada-p2.vala`" una classe que representi una aplicació, p.i.;

```
1   class Application : GLib.Object {

3     public Application (string name) {
        m_name = name;
5     }

7     public void run () { } // Pone en marcha la aplicacion
      public void quit () { }// Termina la aplicacion
9
      private string m_name;
11  }
```

Aquesta classe pertanyerà a l'espai de noms '`Hada`'.
Escriu el codi de la funció **main** [1] que creu un objecte d'aquesta classe, li envie el missatge '`run`' i després '`quit`'.

[1] fixa't que no ha de ser un mètode de la classe.

Exercici 2

A partir d'aquí i fins al final de la pràctica, tots els senyals, quan correspongui, es connectaran en el constructor de la classe corresponent.

- Afegeix a la classe **Application** dos senyals: **void on_init()** y **void on_end()**.
- Aquests senyals hauran d'emetre's a l'inici i al final de la aplicació respectivament.
- Crea una funció independent (no un mètode d'una classe) llamada: **void al_inicio()**.
- Aquesta funció imprimirà per pantalla el text: "`"\nComenzamos...\n"`".
- Connecta-la al senyal **on_init**. Comprova que es executa quan comença l'aplicació i s'emet el senyal corresponent.

Exercici 3

- Crea una funció independent (no un mètode d'una classe) anomenada: **void al_final()**.
- Aquesta funció imprimirà per pantalla el text: "`"\nAcabamos...\n"`".
- Connecta-la al senyal **on_end**. Comprova que es executa quan acaba l'aplicació i s'emet el senyal corresponent.
- Crea una funció independent (no un mètode d'una classe) cridada: **void al_final2()**.
- Aquesta funció imprimirà per pantalla el text: "`"\nAcabamos de verdad...\n"`".
- Connecta-la al senyal **on_end**. Comprova ara que, a més d'executar-se la funció `al_final`, també es executa `al_final2` en acabar l'aplicació i emetre la senyal corresponent.

lsi Department of Software and Computing Systems

Exercici 4 (I)

- Afegida a l'exemple anterior crea una classe 'pila d'enters' (Stack) també pertanyent a l'espai de noms '`Hada`'.
- Aquesta classe tindrà un únic constructor a partir d'un 'string' que representarà el nom de la pila.
- El nombre màxim d'elements que tindrà serà '`10`'.
- Disposarà del mètode '`public void push (int)`' para apilar elements i del mètode '`public int pop ()`' para desapilar elements.
- Podrà emetre dos senyals: '`void stack_underflow ();`' i '`void stack_overflow ();`'.

Exercici 4 (II)

- Quan s'apili un element comprovarà si la pila està plena, si no ho està ho apilarà i si ho està emetrà el senyal `stack_overflow`.
- Quan es desapile un element comprovarà si la pila no està buida, si no ho està el desapilará i si ho està emetrà el senyal `stack_underflow`, després retornarà la constant '`kERROR`'[2] que reflectirà l'error.
- Crea les funcions independents (no mètodes d'una classe): **void on_overflow(...)** y **void on_underflow(...)**. Connecta-les a les funcions corresponents.
- Aquestes funcions imprimiran per pantalla el text: "`"Stack overflow %s index = %d\n"`" y "`"Stack underflow %s index = %d\n"`" respectivament, on els paràmetres ' %s' i '%d' de la cadena de format representen el nom de la pila i l'índex de la mateixa on s'ha intentat apilar o desapilar un element i ha provocat l'emissió d'aquest senyal.
- Quins paràmetres i de quin tipus han de rebre aquests senyals?

[2] `public const int kERROR = -100;`

Objectius...

L'alumne sap:

- ☐ Afegir senyals a una classe.
- ☐ Connectar a un senyal una funció.
- ☐ Connectar a un senyal un mètode.
- ☐ Crear una classe des de zero dotant-la de senyals i connectar-li a aquestes el codi que vol que s'executi en cada cas.