

Tema: V. Interfícies d'usuari i execució diferida de codi. Accés a BB.DD. des d'aplicacions de escriptori Pràctiques.

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant

Curs 2014-2015 , Copyleft © 2011-2015 .
Reproducció permesa sota els termes de la llicència de documentació
lliure GNU.



1 / 13

- ① Requisites
- ② Apartat “stack”
- ③ Apartat “stack”
- ④ Apartat “stack”
- ⑤ Apartat “stack” - funcionament
- ⑥ Apartat “sql”. Primera part
- ⑦ Apartat “sql”. Primera part
- ⑧ Apartat “sql”. Segona part
- ⑨ Apartat “sql”. Segona part
- ⑩ Lliura
- ⑪ Objectius...



2 / 13

Requisits

- Respecta noms de directoris, arxius i identificadors.
- Aquesta pràctica consta de dos apartats: **stack** i **sql**.
- Descàrrega de la secció de pràctiques del Campus Virtual el material “hada-p3.tgz”. En descomprimir-ho crearà un directori anomenat “hada-p3” i dins d'ell dues directoris anomenats “stack” i “sql”. Cada un d'ells es correspon amb un dels dos apartats d'aquesta pràctica.



3 / 13

Apartat “stack”

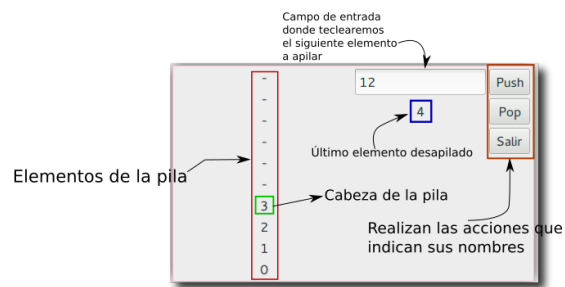
Aquest apartat de la pràctica 3 es basa en la pràctica 2.

- Dins del directori “stack” crea els arxius que contindran tot el codi font demanat: “hada-p3-modelo.vala”, “hada-p3-vista.vala”. El fitxer “model-view.vala” ja ho tens creat allí.
- Anem a usar la classe “Hada.Stack” creada en la pràctica 2, solament que ara derivarà de la classe “Mvc.Model” que tens en “model-view.vala”. Tot el codi d'aquesta classe estarà en l'arxiu “hada-p3-model.vala”.
- Sobre aquest model de la pila “Fada.Stack” anem a crear una Vista -l'interfície d'usuari- en Gtk amb l'ajuda de glade. Aquest interfície estarà en l'arxiu “hada-p3.ui”.
- Pots crear el teu propi programa principal de prova en un arxiu anomenat “main.vala”. **No has de lliurar-ho.**



4 / 13

- En l'arxiu "hada-p3-vista.vala" crearem una classe anomenada "Stack_vista", com el seu nom indica serà una vista sobre el model representat per "Hada.Stack". Implementarà el *interfície* "Mvc.View".
- Tindrà dos mètodes¹: "void push_clicked (Button)" y "void pop_clicked (Button)" que seran els manejadores de les senyals corresponent dels botons "Push" i "Pop" del *interfície*.
- Tindrà aquest aspecte:



Perquè el autocorrector funcioni amb l'interfície gràfic de la teva aplicació tingues en compte el següent:

- Els "elements de la pila" i el "últim element desapilado" són widgets de tipus "Label". Els noms dels widgets "elements de la pila" seran "label0", "label1" ... "label19", i el del "últim element desapilado" serà "label_pop".
- El camp d'entrada serà de tipus "Entry" i permetrà teclejar el següent element a apilar. Es dirà "entry_push".
- El botó "Pop" es dirà "button_pop".
- El botó "Push" es dirà "button_push".
- El botó "Sortir" es dirà "button_terminar".
- La finestra que conté a tots es dirà "ventana1".

Apartat "stack" - funcionament

- El programa principal que construeixis per provar la teva interfície sobre la classe "Hada.Stack" permetrà que l'usuari vagi fent operacions Push i Pop.
- Reflectirà l'estat de la pila amb els widgets label0 a label19 així com l'últim element tret de la mateixa en label_pop.
- Seguirà emetent els mateixos senyals que en la pràctica-2 quan correspongui.
- Finalitzarà l'execució quan l'usuari premi el botó sortir.

Apartat "sql". Primera part

- Dins del directori "sql" disposes dels següents fitxers:
 - "bd-vista.vala" -codi de la vista -
 - "clientecad.vala" -codi del component de accés a dades-
 - "clienteen.vala" -codi de l'entitat de negoci-
 - "create-db.sh" -guió del shell que recrea la bb.dd. amb els valors originals-
 - "Makefile" -compila i crea la bb.dd. original-
 - "model-view.vala" -codi de la classe Mvc.Model i de l'interfície Mvc.View-
 - "vala-bd-gtk.ui" -interfície d'usuari-
 - "vala-bd-gui.vala" -programa principal-

- Aquest codi no compila, està encara per acabar, per tant la primera part d'aquest apartat consisteix que:
 - Facis els afegits/modificacions perquè compili².
 - Una vegada que compili l'aplicació obtinguda, la qual fa ús de gtk i sqlite, et permetrà moure't avanci i enrere per tots els registres de la bb.dd. de prova (hadadb) que s'haurà creat en el directori actual al executar “make” o “make crea-bd”.

Això es fa cridant al CAD directament des del constructor de la “vista” a manera d'exemple perquè tinguis una aplicació funcional facil de comprendre.
- Disposes d'un programa principal de prova en l'arxiu llamado “vala-bd-gui.vala”. **No has de lliurar-ho.**

Per realitzar aquesta segona part és indispensable que hakis aconseguit resoldre la primera.

- Afegeix dos botons a l'interfície d'usuari. Es diran “w_crear”, “w_borrar”.
- Al senyal *clicked* del primer hauràs de connectar-li el mètode d'instància de la classe *BD_vista1* anomenat “public void on_crear_clicked (Button)”, i al segon “public void on_borrar_clicked (Button)”, també de instància i pertanyent a la mateixa classe.

²Fixa't en els errors que dóna en compilar para que t'ajudi en aquest apartat

- “on_crear_clicked” fent ús del CAD per als clients, crearà un nou registre en la bb.dd. que contindrà el nom, adreça i ciutat que l'usuari hagi teclejat en aquest moment en els camps d'entrada de text corresponents de nostra aplicació.
- “on_borrar_clicked” fent ús del CAD per als clients i tenint en compte el contingut que hi hagi en aquest moment solament en el widget “w_nombre”, esborrarà aquells registres de la bb.dd. el camp del qual “nom” coincideixi exactament amb el que l'usuari hagi teclejat en “w_nombre”.
- Tingues en explica que les trucades al CAD s'han de fer des de l'EN, per tant primer hauràs de crear una EN a partir de les dades necessaris de la vista, i aquesta EN demanarà al CAD que dugui a terme l'operació demanada. Hauràs de per tant crear també les operacions necessàries en les EN perquè cridin als CAD.
- El lliurament consistirà en un fitxer anomenat “hada-p3.tgz” que contindrà el directori principal i tot el seu contingut.
- No ocuparà mes de 512KB.

L'alumne sap:

- ☐ Crear un interfície d'usuari amb un constructor gràfic d'interfícies d'usuari.
- ☐ Carregar-ho dinàmicament en temps d'execució para mostrar-ho a l'usuari.
- ☐ Usar l'arquitectura MVC en aplicacions amb interfície gràfic d'usuari.
- ☐ Usar un model de capes per comunicar-se amb un gestor de bases de dades relacionals des d'aplicacions d'escriptori.
- ☐ Crear una aplicació dotada d'interfície gràfic de usuari que usant el model de capes i MVC és capaç de comunicar-se amb el gestor de bases de dades relacionals.