Комитет по образованию Правительства Санкт-Петербурга  
Государственное бюджетное учреждение  
среднего профессионального образования  
«Колледж информационных технологий»

МДК 03.01

«Технология разработки программного обеспечения»

**отчёт**

по практической работе №6

Выполнил студент 474 гр.:

Черных Артём

Преподаватель:

Фомин А.В.

Санкт-Петербург 2020

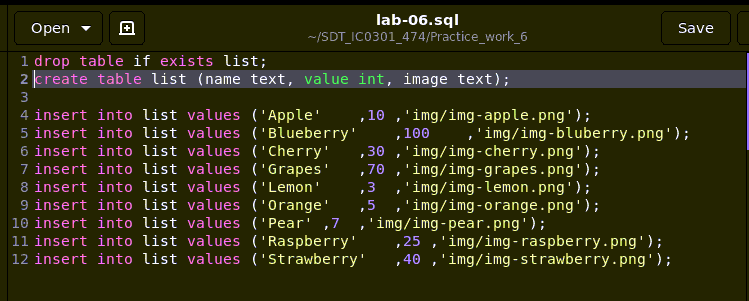
1. Структуру интерфейса «Интеграция с базой данных SQLite3»:

Черных Артём



1. Сценарий на языке запросов SQL, для создания таблицы в sqlite3:

Черных Артём



1. Выполним запрос в sqlite3:

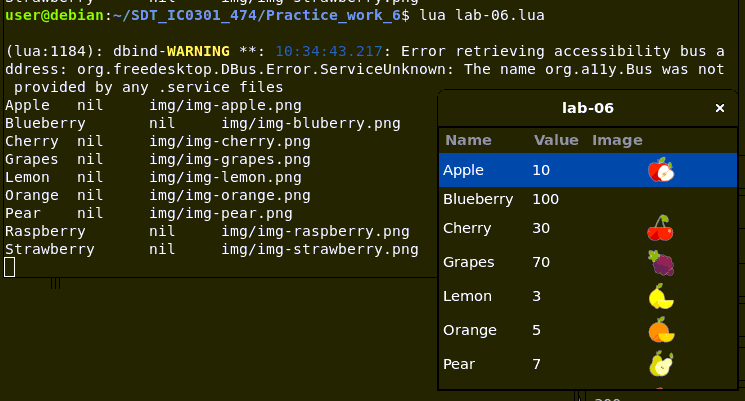
1)В терминале зайдём в БД: «sqlite3»

2)Считаем файл: «.read lab-06.sql»

3)Сохраним файл: «.save lab-06.db»

1. Программный код, в котором подключили библиотеку«lsqlite3»:
2. #!/usr/bin/lua5.3
3. lgi = require 'lgi'
4. --package.loaded[lsqlite3]
5. sqlite = require 'lsqlite3'
6. gtk = lgi.Gtk
7. pb = lgi.GdkPixbuf.Pixbuf
8. gtk.init()
9. bld = gtk.Builder()
10. bld:add\_from\_file('lab-06.glade')
11. ui = bld.objects
12. rdr\_txt = gtk.CellRendererText{}
13. rdr\_pb = gtk.CellRendererPixbuf{}
14. c1 = gtk.TreeViewColumn { title = 'Name', { rdr\_txt, { text = 1 }} }
15. c2 = gtk.TreeViewColumn { title = 'Value', { rdr\_txt, { text = 2 }} }
16. c3 = gtk.TreeViewColumn { title = 'Image', { rdr\_pb, { pixbuf = 3 }} }
17. ui.tree:append\_column(c1)
18. ui.tree:append\_column(c2)
19. ui.tree:append\_column(c3)
20. db = sqlite.open('lab-06.db')
21. --print(db)
22. for row in db:nrows('select \* from list') do
23. print(row.name, row.valie, row.image)
24. --local pb = pixbuf.new\_from\_file(row.image)
25. img = pb.new\_from\_file(row.image)
26. el = ui.tree\_mdl:append()
27. ui.tree\_mdl[el] = { [1] = row.name, [2] = row.value, [3] = img}
28. end
29. db:close()
30. ui.wnd.title = 'lab-06'
31. ui.wnd.on\_destroy = gtk.main\_quit
32. ui.wnd:show\_all()
33. gtk.main()
34. Приложение в работе:

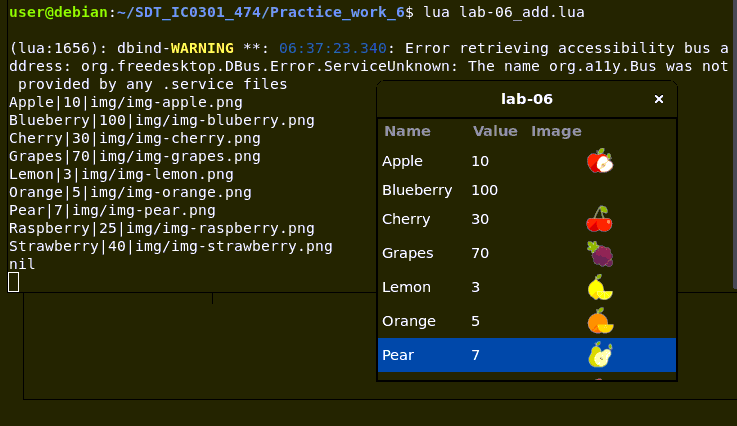
Черных Артём



Дополнительное задание

1. Интерфейс программы остаётся тот же.
2. Программный код (использование функции io.popen() для запуска дочернего процесса sqlite3 и передачей в эту функцию запроса на языке SQL и получением результатов):
3. #!/usr/bin/lua5.3
4. lgi = require 'lgi'
5. --package.loaded[lsqlite3]
6. sqlite = require 'lsqlite3'
7. gtk = lgi.Gtk
8. pb = lgi.GdkPixbuf.Pixbuf
9. gtk.init()
10. bld = gtk.Builder()
11. bld:add\_from\_file('lab-06.glade')
12. ui = bld.objects
13. rdr\_txt = gtk.CellRendererText{}
14. rdr\_pb = gtk.CellRendererPixbuf{}
15. c1 = gtk.TreeViewColumn { title = 'Name', { rdr\_txt, { text = 1 }} }
16. c2 = gtk.TreeViewColumn { title = 'Value', { rdr\_txt, { text = 2 }} }
17. c3 = gtk.TreeViewColumn { title = 'Image', { rdr\_pb, { pixbuf = 3 }} }
18. ui.tree:append\_column(c1)
19. ui.tree:append\_column(c2)
20. ui.tree:append\_column(c3)
21. --[[db = sqlite.open('lab-06.db')
22. --print(db)
23. for row in db:nrows('select \* from list') do
24. print(row.name, row.valie, row.image)
25. --local pb = pixbuf.new\_from\_file(row.image)
26. img = pb.new\_from\_file(row.image)
27. el = ui.tree\_mdl:append()
28. ui.tree\_mdl[el] = { [1] = row.name, [2] = row.value, [3] = img}
29. end
30. db:close()]]
31. local base = 'lab-06'
32. local sql = 'select \* from list'
33. f = io.popen('sqlite3 ' .. base .. '.db "' .. sql .. '"')
34. while true do
35. local row = f:read('\*l')
36. print(row)
37. if row == nil then break end
38. local \_, \_, name, value, image = row:find('(%w+)|(%d+)|(.-)$')
39. img = pb.new\_from\_file(image)
40. el = ui.tree\_mdl:append()
41. ui.tree\_mdl[el] = { [1] = name, [2] = value, [3] = img}
42. end
43. f:close()
44. ui.wnd.title = 'lab-06'
45. ui.wnd.on\_destroy = gtk.main\_quit
46. ui.wnd:show\_all()
47. gtk.main()
48. Приложение в работе:

Черных Артём



1. Для большей автоматизации напишем shell-script для создания базы данных и запуска приложения:
2. #!/usr/bin/sh
3. rm -f lab-06.db
4. cat lab-06.sql | sqlite3 lab-06.db
5. Запустим shell-script (как видим ошибку не выдал, значит успешно сработал):

Черных Артём

