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 №8

Выполнил студент 474 гр.:

Черных Артём

Преподаватель:

Фомин А.В.

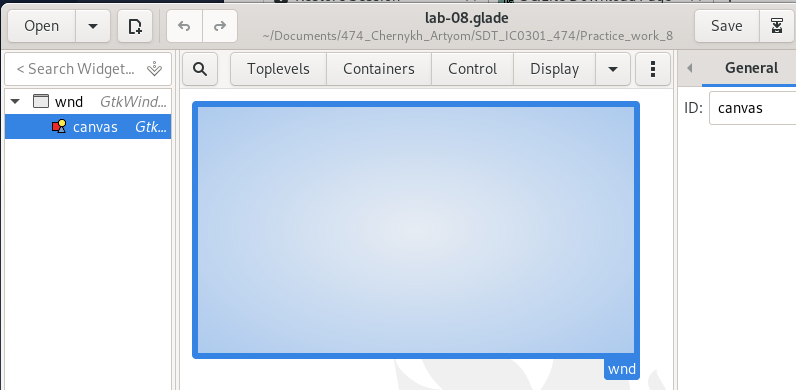
Санкт-Петербург 2020

**Серверное приложение**:

1. Сценарий, который изменяет перeодически значения в хранилище Redis:
2. #!/usr/bin/lua5.3
3. local lgi = require 'lgi'
4. local glib = lgi.GLib
5. local gtk = lgi.Gtk
6. gtk.init()
7. -- connect to redis keystore db
8. local rds = require 'redis'
9. local cli = rds.connect('redis.fxnode.ru', 6379)
10. local t = 0 -- global time
11. local function on\_timer()
12. t = t + 0.1 -- increment time
13. local x = 50 + math.cos(t) \* 25 -- compute new coordinates
14. local y = 50 + math.sin(t) \* 25
15. -- update values in redis keystore
16. cli:set('x-Chernykh\_Artyom', x)
17. cli:set('y-Chernykh\_Artyom', y)
18. -- schedule call after ~10 ms
19. glib.timeout\_add(glib.PRIORITY\_DEFAULT, 10, on\_timer)
20. end
21. on\_timer() -- immediate first call
22. gtk.main() -- process time events
23. Проверить работоспособность приложения можно, используя клиент Redis(redis-cli). Команды для проверки: keys \* и get имя\_ключа для получения информации о ключах. Чтобы узнать информацию о текущих соединениях и количестве выполненных запросов – запустите клиент с параметром –stat :

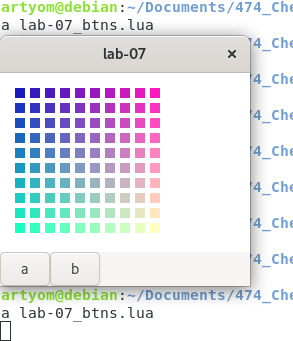
**Пользовательское приложение**

1. Интерфейс приложения:



1. Программный код приложения для получения значений ключей в хранилище Redis и вывода изображения с объектом (квадратик), который будет менять координаты по заданным в этих ключах:
2. #!/usr/bin/lua5.3
3. rds = require 'redis' -- connect to redis keystore
4. cli = rds.connect('redis.fxnode.ru', 6379)
5. lgi = require 'lgi'
6. glib = lgi.GLib
7. gtk = lgi.Gtk
8. gtk.init()
9. bld = gtk.Builder()
10. bld:add\_from\_file('lab-08.glade')
11. ui = bld.objects
12. x = 0 -- last know coordinates
13. y = 0
14. function ui.canvas:on\_draw(cr)
15. cr:set\_source\_rgb(1,1,1) -- fill background
16. cr:paint()
17. cr:set\_source\_rgb(0, 0.5, 1, 1) --draw smal rectangle at given position
18. cr:rectangle(x, y, 10, 10)
19. cr:fill()
20. return true
21. end
22. function ui.wnd:on\_destroy()
23. gtk.main\_quit() -- leave event loop
24. end
25. function on\_timer()
26. x = cli:get('x-Chernykh\_Artyom')
27. y = cli:get('y-Chernykh\_Artyom')
28. ui.canvas:queue\_draw()
29. glib.timeout\_add(glib.PRIORITY\_DEFAULT, 10, on\_timer) -- schedule next run
30. end
31. ui.wnd:show\_all()
32. on\_timer() -- immediate firs call
33. gtk.main() -- event loop
34. Запуск серверного приложения и клиентского (желательно протестировать на двух компьютерах). Для передачи данных используем сервер Redis размещенный по адресу (redis.fxnode.ru) и ключами (x-Chernykh\_Artyom, y-Chernykh\_Artyom). Пример работы приложения:
35. Ва
36. вы

Черных Артём



При нажатии на кнопку «b»:

Черных Артём

