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 №9

Выполнил студент 474 гр.:

Черных Артём

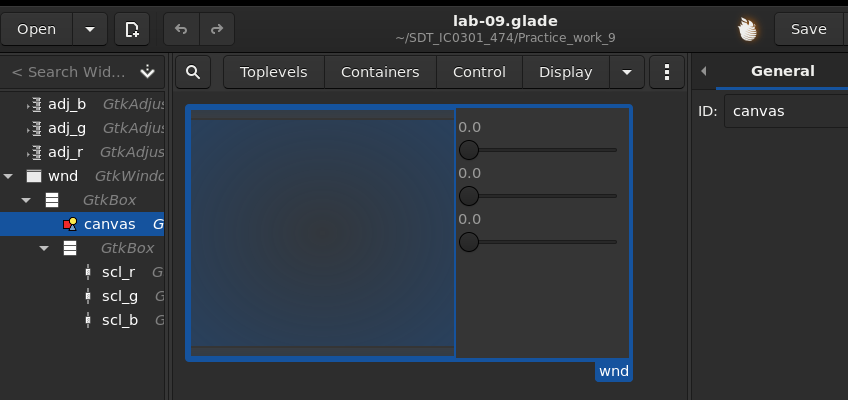
Преподаватель:

Фомин А.В.

Санкт-Петербург 2020

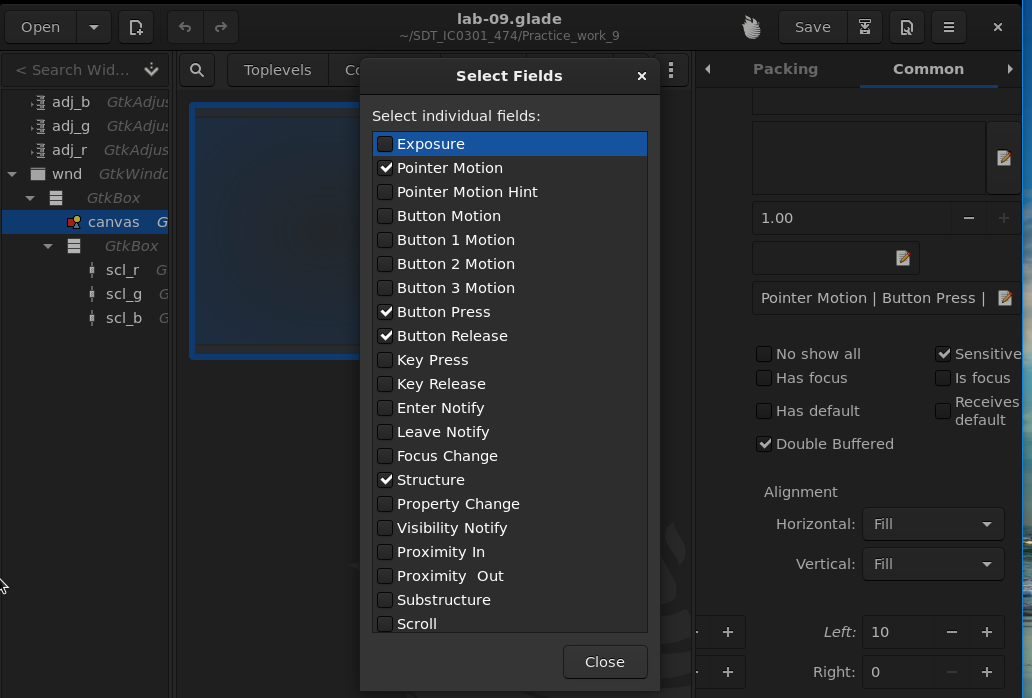
1. Интерфейс приложения:

Черных Артём



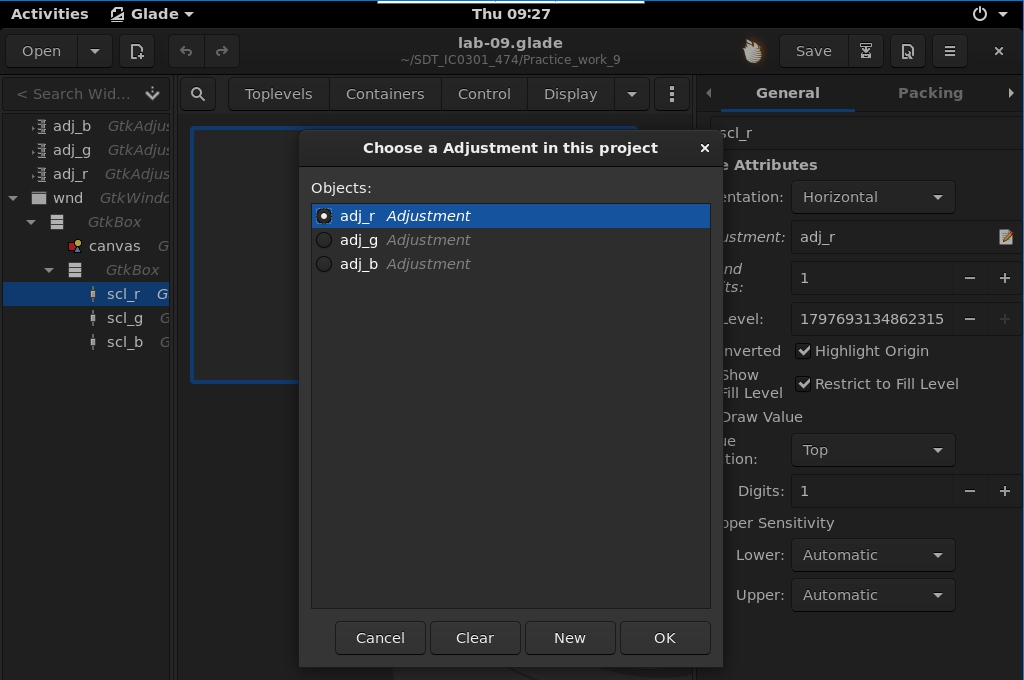
1. Разрешение компоненту GtkDrawingArea получение событий: Pointer Motion, Button Press и Button Release изменив св-во **Events** на вкладке **Common**:

Черных Артём



1. Настройка отображение текущего значения у GtkScale, создав элементы GtkAdjustment:

Черных Артём



1. Для всех GtkAdjustment настройка параметров Maximum Value, Step Increment и Page Increment:

Черных Артём



1. Программный код, где реализована обработка событий мыши button\_press\_event, button\_release\_event и motion\_notify\_event. Отрисовка прямоугольника в месте курсора мыши, а при зажатой ЛКМ увеличение его в размерах. Цвет прямоугольника выбирается с использование компонентов GtkScale:
2. Приложение в работе:
3. Проверить работоспособность приложения можно, используя клиент Redis(redis-cli). Команды для проверки: keys \* (рисунок 2.1) и get-имя\_ключа (рисунок 2.2) для получения информации о ключах. Чтобы узнать информацию о текущих соединениях и количестве выполненных запросов – запустите клиент с параметром –stat (рисунок 2.3) :

Черных Артём

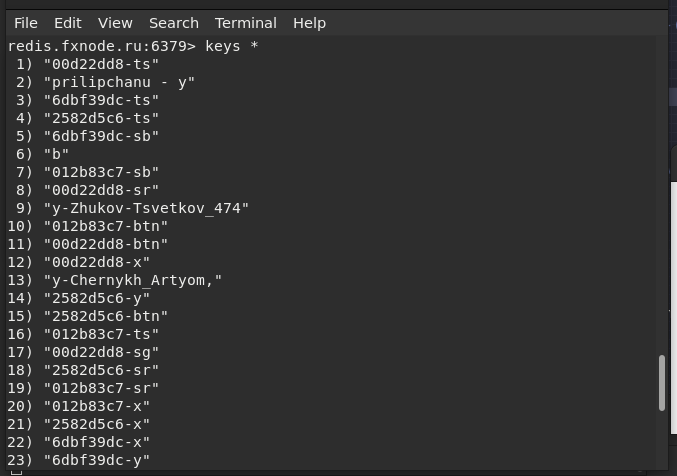


Рисунок 2.1 – Вывод всех ключей

Черных Артём

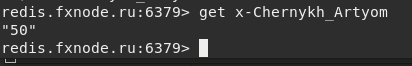


Рисунок 2.2 – Вывод конкретного ключа

Черных Артём

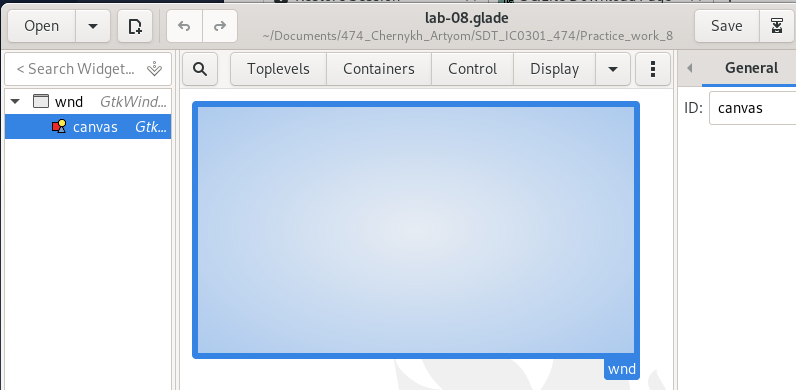


Рисунок 2.3 – Текущие запросы и соединения

**Пользовательское приложение**

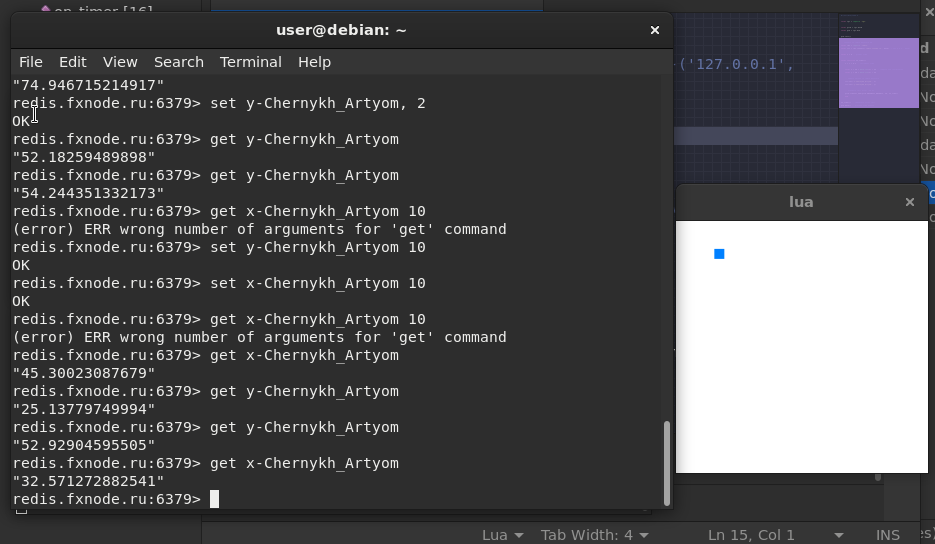
1. Интерфейс приложения:

Черных Артём



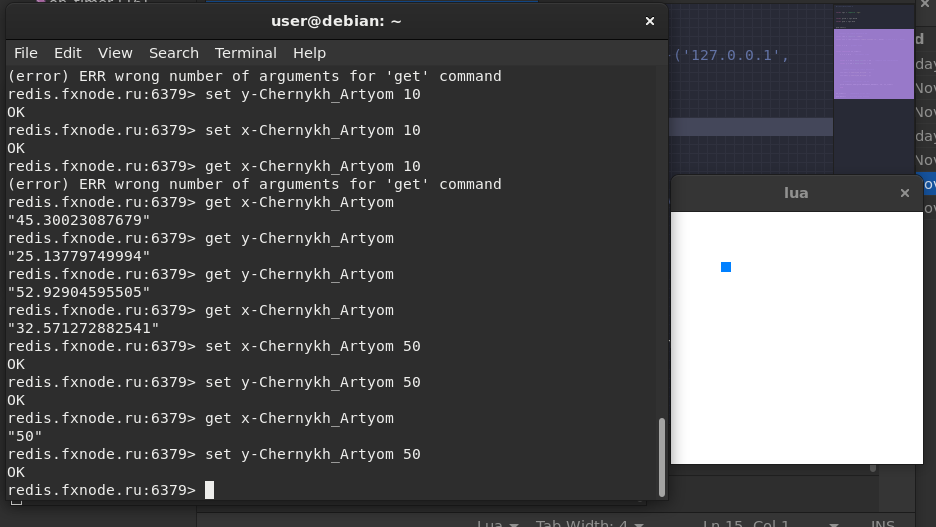
1. Программный код приложения для получения значений ключей в хранилище Redis и вывода изображения с объектом (квадратик), который будет менять координаты по заданным в этих ключах:
2. #!/usr/bin/lua5.3
3. rds = require 'redis' -- connect to redis keystore
4. cli = rds.connect('redis.fxnode.ru', 6379)
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:

Черных Артём



До ввода значений ключей

Черных Артём



После ввода значений ключей