|  |  |
| --- | --- |
| **Предмет** | Технология разработки программного обеспечения |
| **Курс** | 3 |
| **Семестр** | 1 |
| **Работа** | 2 |
| **Дата** | 16.10.2020 |
| **Группа** | 474 |
| **Фамилия** | Алексеев |
| **Имя** | Игорь |
| **Отчество** | Алексеевич |

1. Создайте окно Окна – GtkWindow и добавьте в него GtkBox из Контейнеры. Сделайте 3 слота в GtkBox, пользуясь свойством Количество элементов. Добавьте в верхний слот GtkGrid из Контейнеры, а во второй и третий GtkBox. Измените количество слотов для кнопок в дочерних GtkBox до 5 и 4 штук, и количество колонок в GtkGrid на 2 штуки, пользуясь свойством Количество элементов.

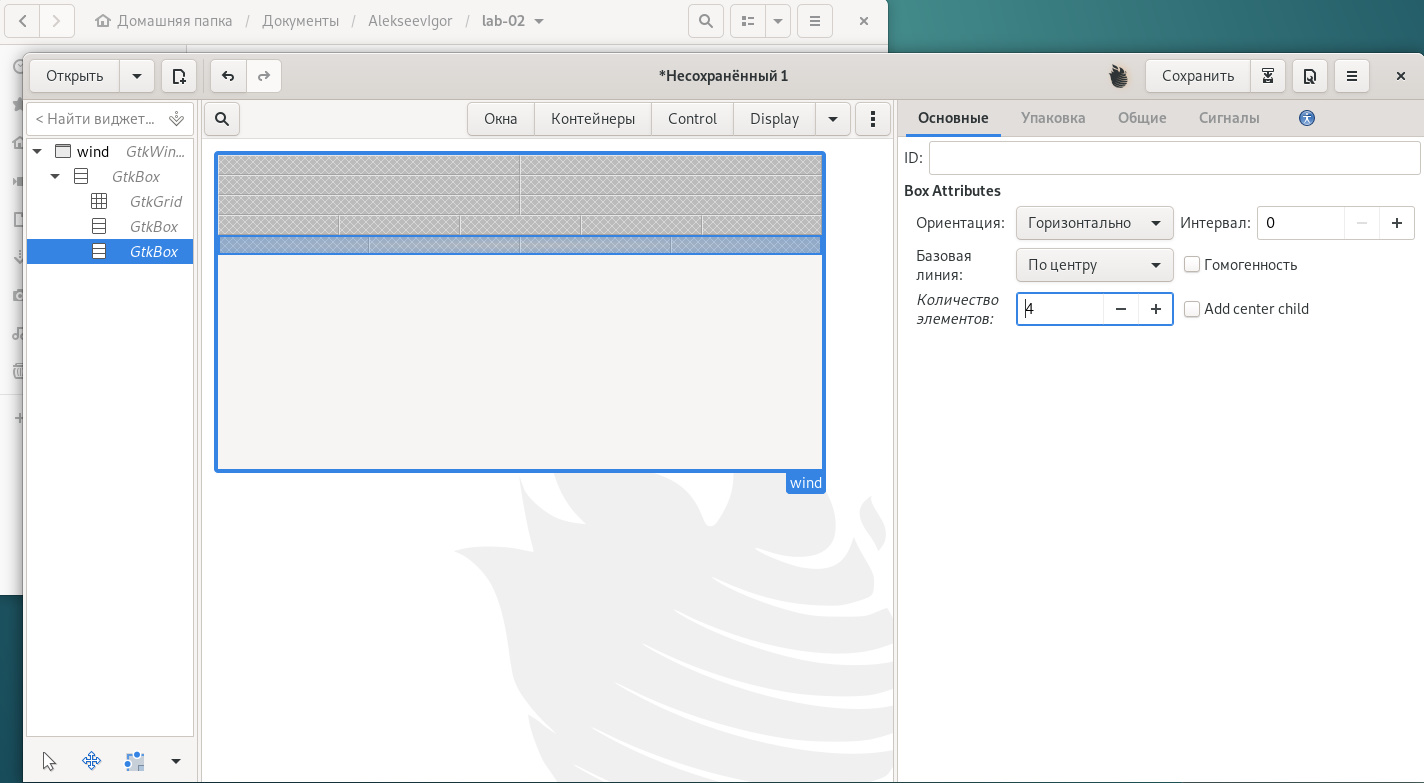


Рисунок 1 – Создание контейнеров

1. Расширьте место в верхнем слоте GtkBox для компонента GtkGrid используя свойство Расширяемость и Заполнение для сжатия самого GtkGrid под содержимое.

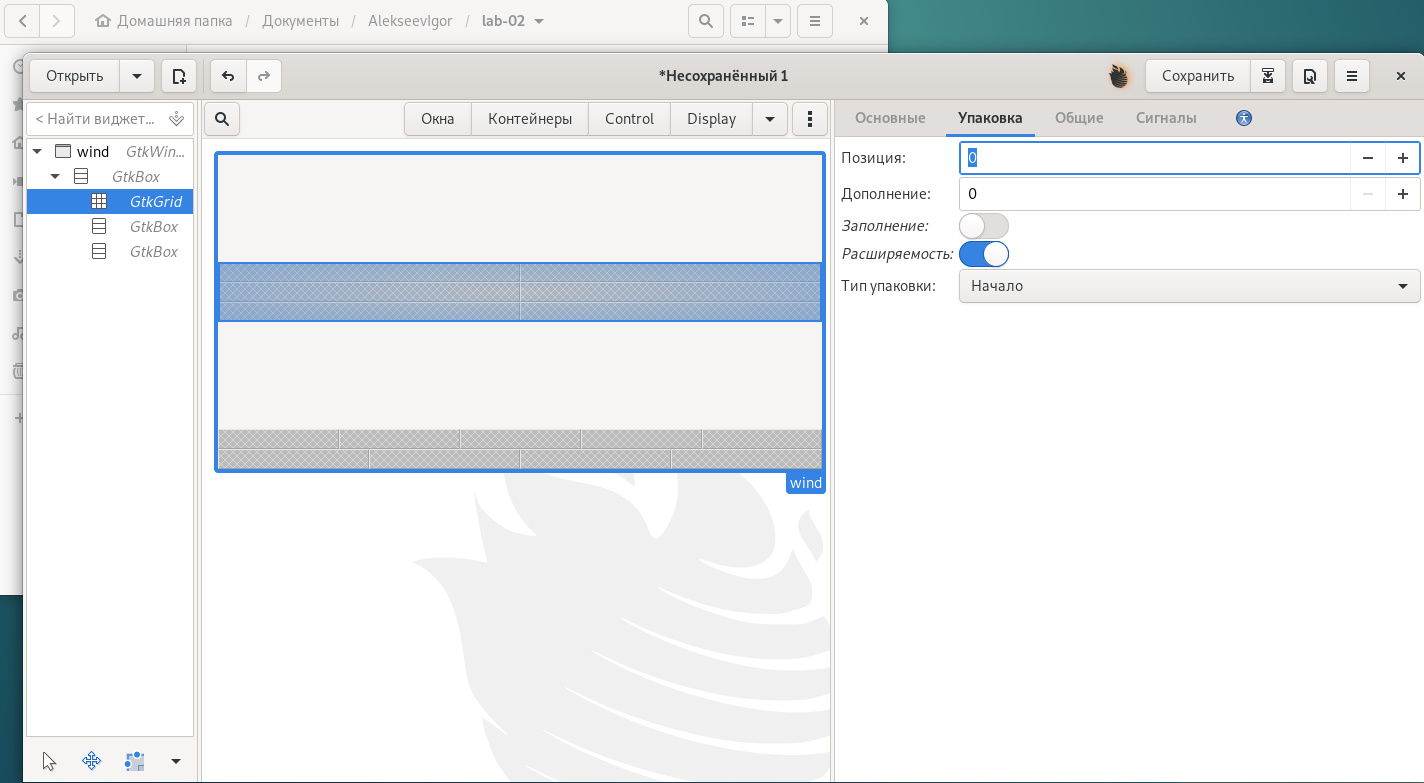


Рисунок 2 – Масштабирование контейнеров

1. Добавьте кнопки управления в дочерние GtkBox.

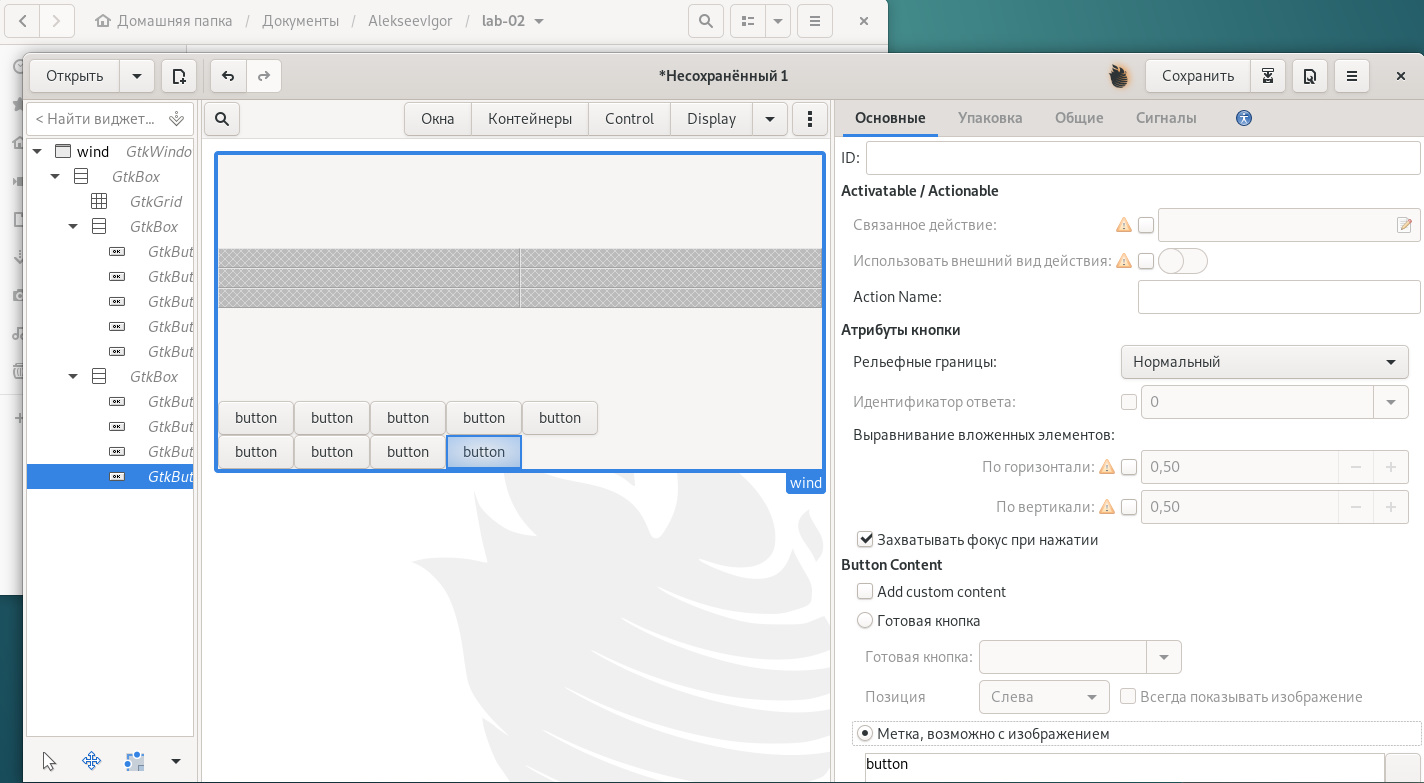


Рисунок 3 – Добавление кнопок

1. Выровняйте панели с кнопками по центру (по горизонтали). Используйте свойство Выравнивание По горизонтали в состояние По центру.

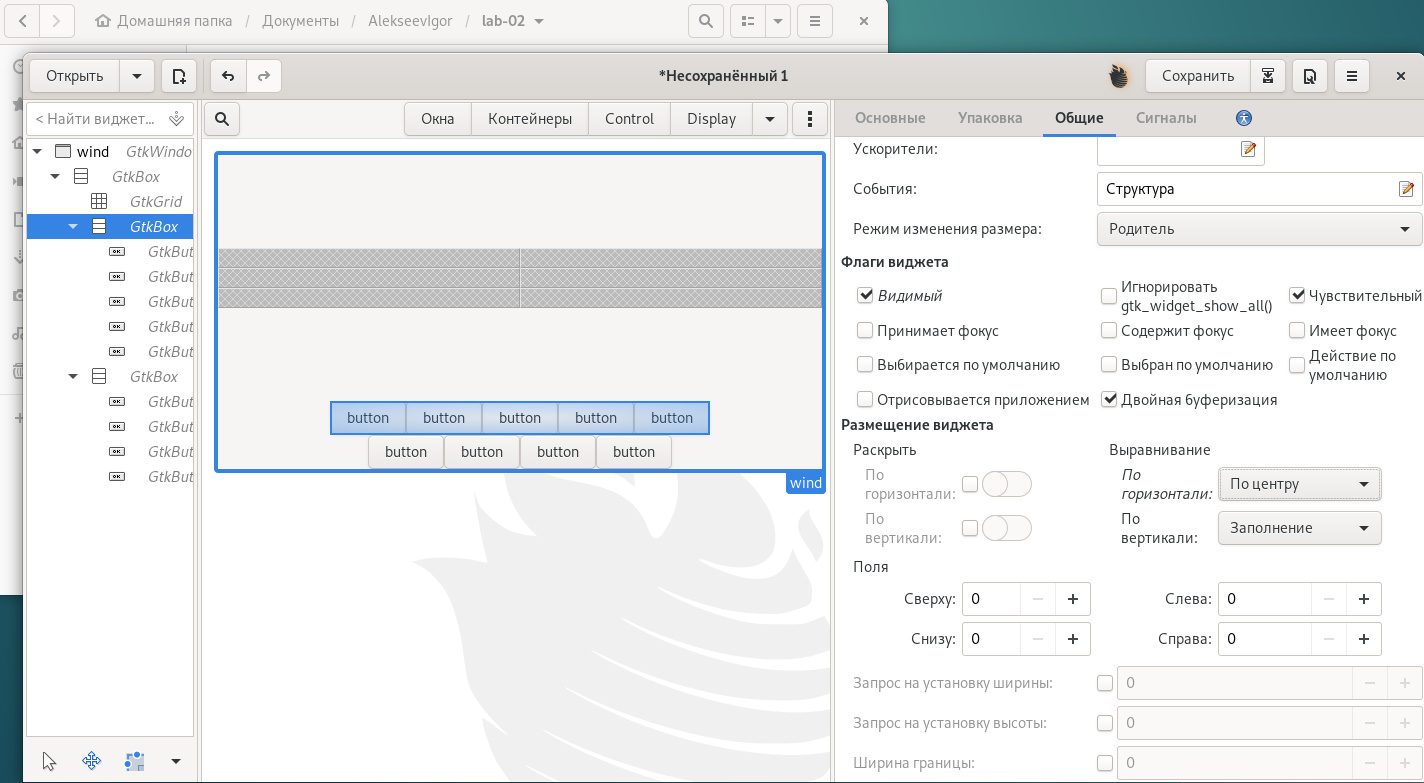


Рисунок 4 – Центрирование кнопок

1. Добавьте надписи из Display – GtkLabel и текстовые поля ввода из Control – GtkEntry в GtkGrid. Выровняйте элемент GtkGrid по центру (по горизонтали) относительно ячейки его родительского контейнера GtkBox. Для этого переключите свойство Выравнивание По горизонтали в состояние По центру. Сделайте ячейки для полей в GtkGrid одинаковой высоты установив свойство Строки: Homogeneous и установите расстояние между ячейками по горизонтали указав значение в свойстве Столбцы: Spacing.

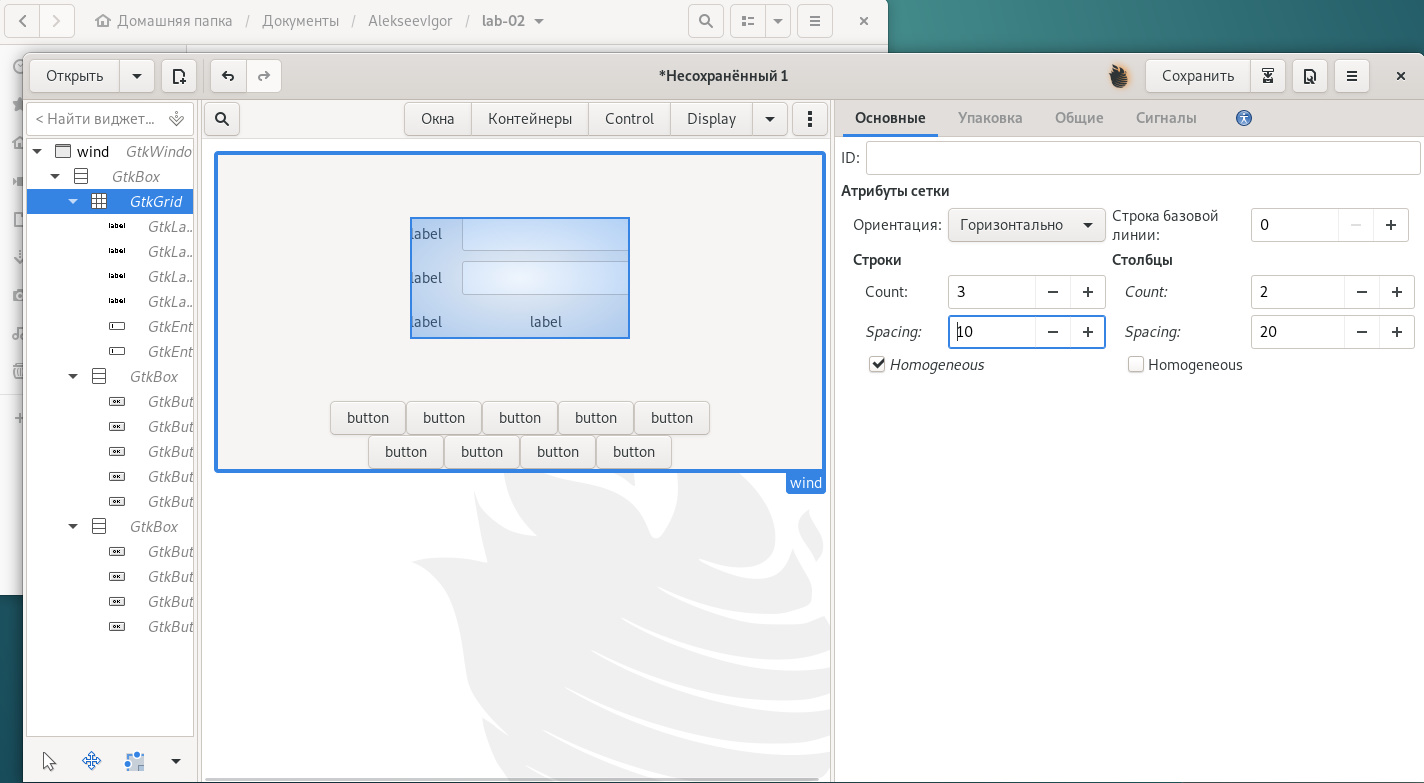


Рисунок 5 – Добавление элементов в сетку, и размещение её по центру

1. Сделайте ячейки для кнопок в GtkBox одинаковой ширины установив свойство Гомогенность и установите расстояние между ячейками указав значение в свойстве Интервал.

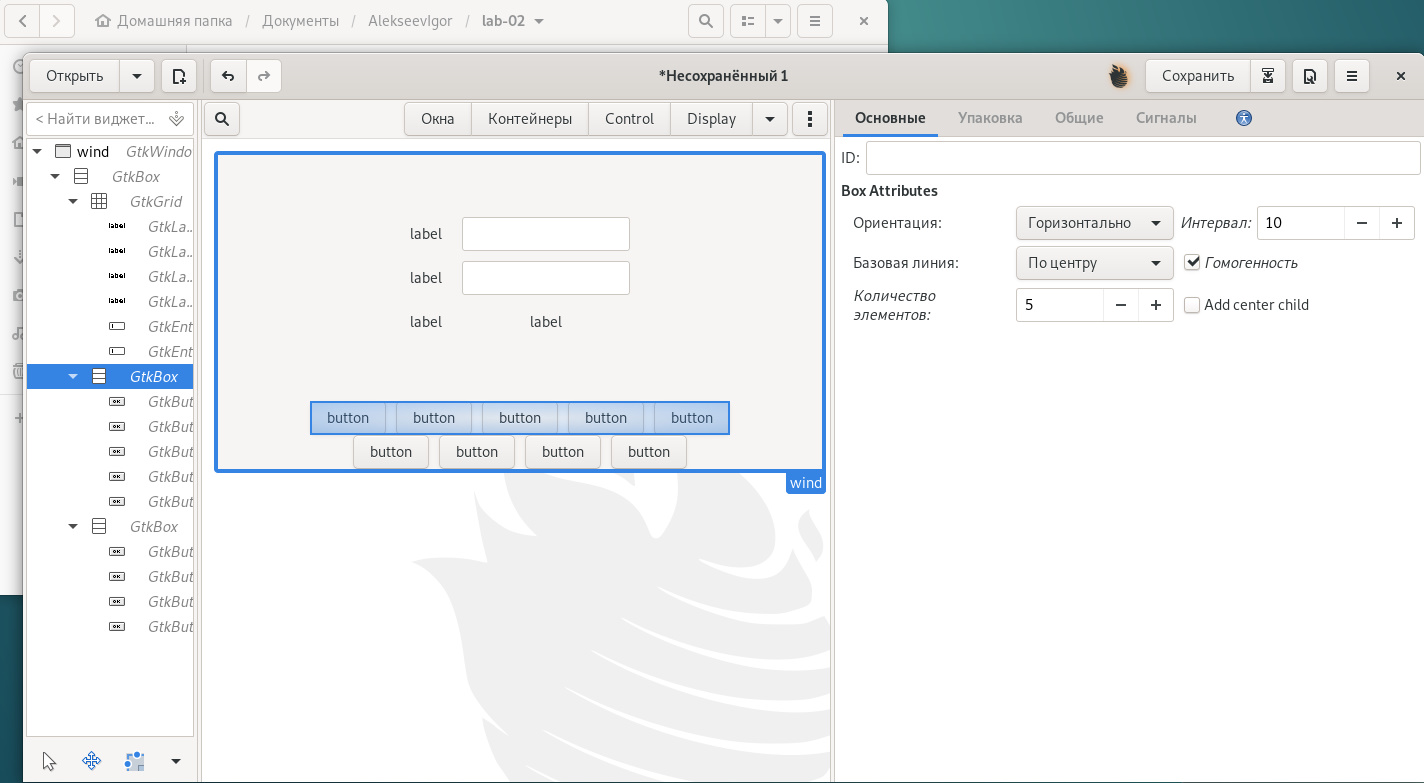


Рисунок 6 – Установить расстояние между кнопками

1. Определите отступ родительского GtkBox указав свойства Ширина границы.

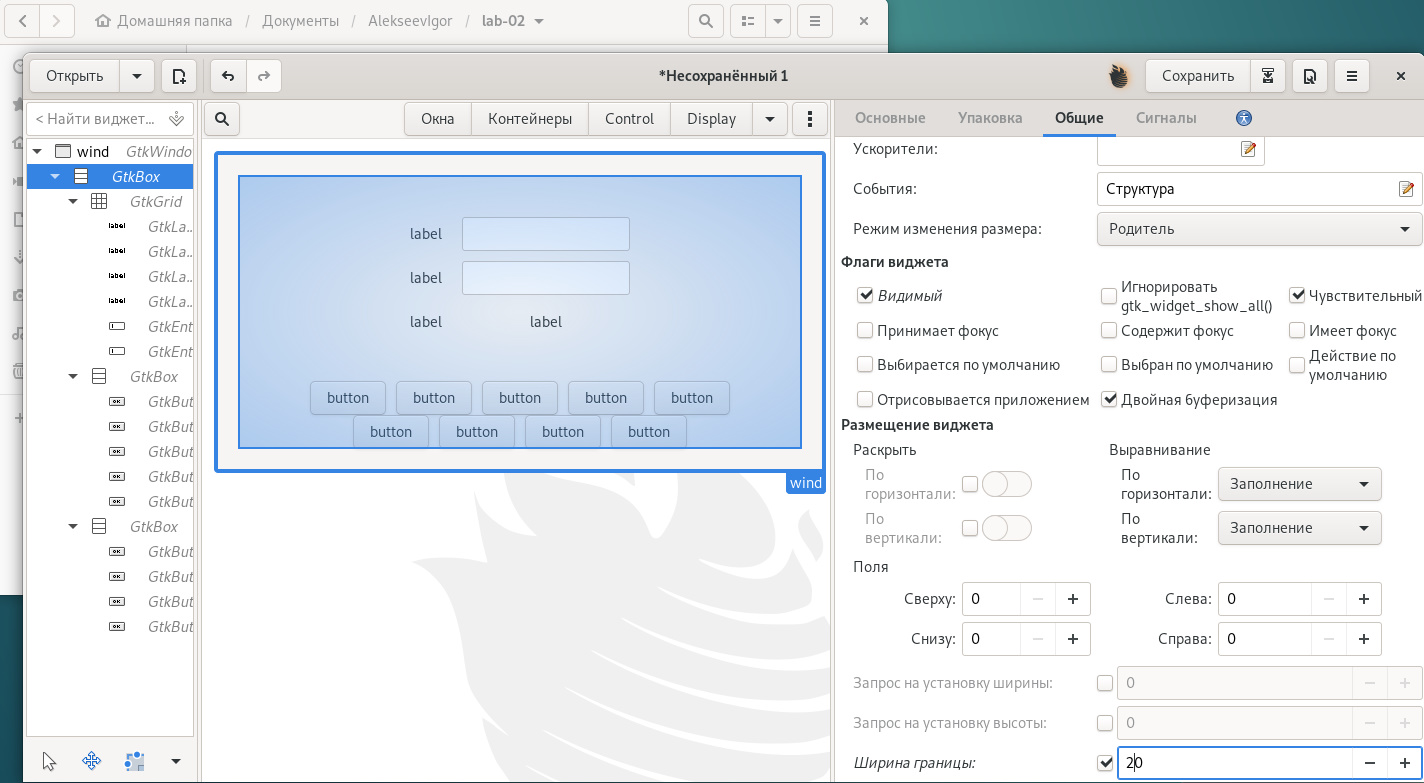


Рисунок 7 – Установление отступов

1. Установите иконки на кнопки пользуясь свойством Изображение. Выберите изображение из списка затем нажмите ОК или создайте новое кнопкой Создать.

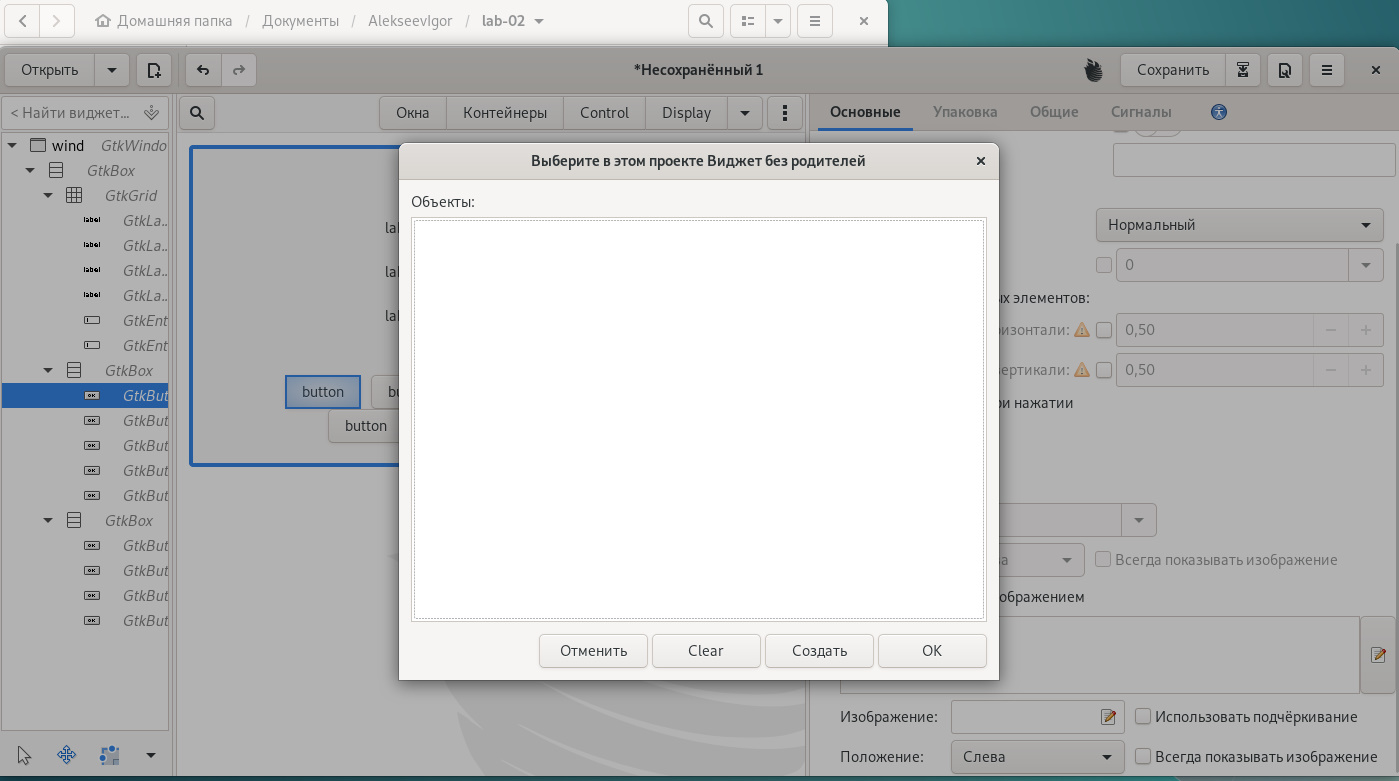


Рисунок 8 – Запуск программы

1. Выберите созданное изображение и настройте его, используя свойства Идентификатор (ID) встроенного значка или Имя файла. Включите принудительное отображение иконок на кнопках используя свойство Всегда показывать изображение.

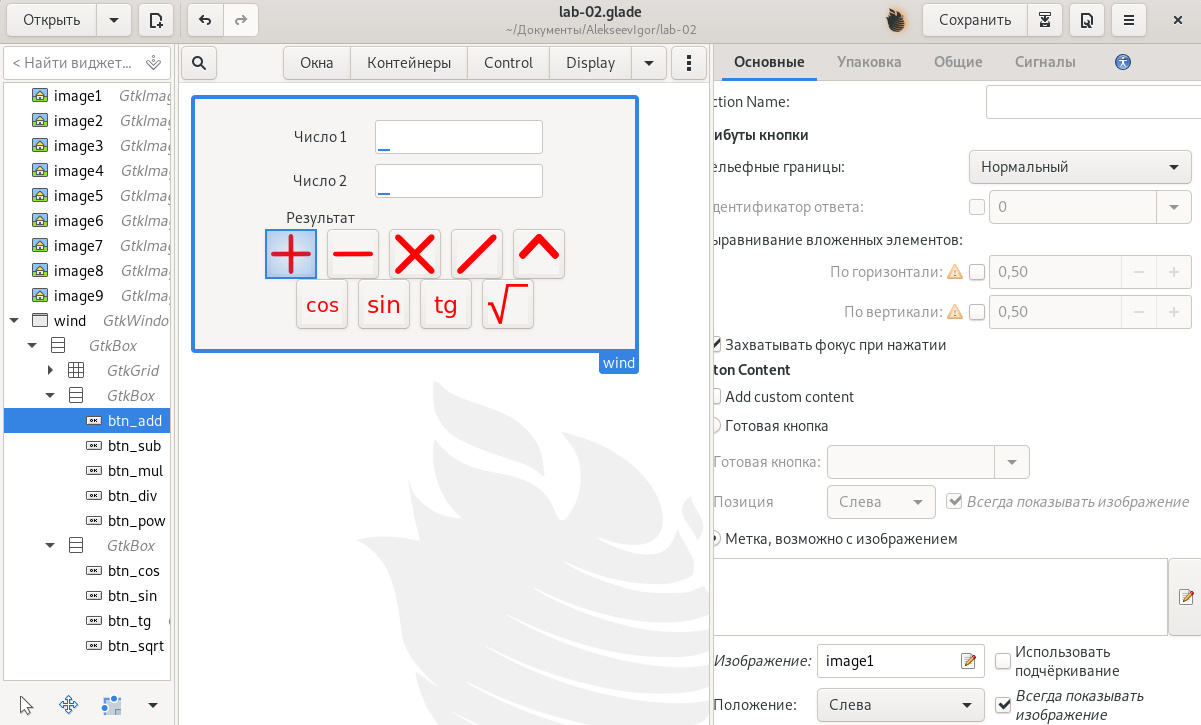


Рисунок 9 – Картинки на кнопках

1. Задайте окну, кнопкам, текстовым полям ввода и метке с результатом идентификаторы в свойстве ID.

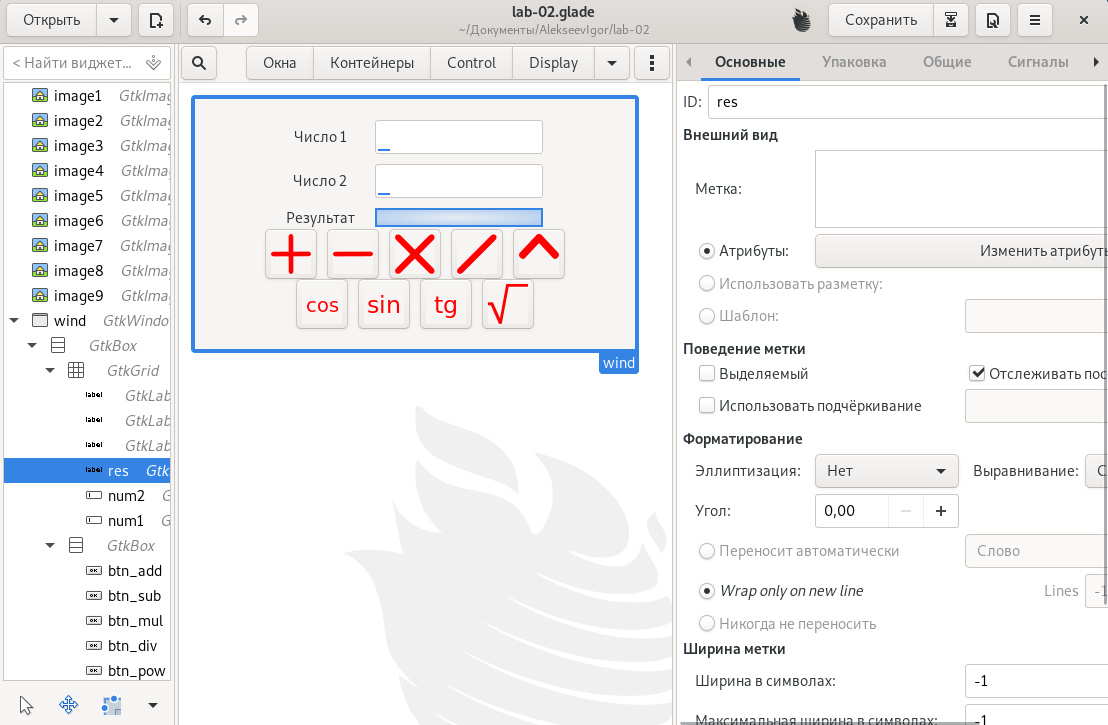


Рисунок 10 – Указание ID элементу

1. Создайте новый файл lab-02.lua и напишите код ниже.

*lgi* *=* require 'lgi'

*gtk* *=* *lgi*.*Gtk*

*gtk*.*init*()

*bld* *=* *gtk*.*Builder*()

bld:*add\_from\_file*('lab-02.glade')

*ui* *=* *bld*.*objects*

*ui*.*wind*.*title* *=* 'lab-02\_Алексеев\_Игорь'

*ui*.*wind*.*on\_destroy* *=* *gtk*.*main\_quit*

*ui*.wind:*show\_all*()

*function* *ui*.btn\_add:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*b* *=* tonumber(*ui*.*num2*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *and* type(tonumber(*b*)) *==* 'number' *then*

*ui*.*res*.*label* *=* *a* *+* *b*

*else*

*ui*.*res*.*label* *=* 'Заполните первое и второе поле'

*end*

*end*

*function* *ui*.btn\_sub:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*b* *=* tonumber(*ui*.*num2*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *and* type(tonumber(*b*)) *==* 'number' *then*

*ui*.*res*.*label* *=* *a* *-* *b*

*else*

*ui*.*res*.*label* *=* 'Заполните первое и второе поле'

*end*

*end*

*function* *ui*.btn\_mul:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*b* *=* tonumber(*ui*.*num2*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *and* type(tonumber(*b*)) *==* 'number' *then*

*ui*.*res*.*label* *=* *a* *\** *b*

*else*

*ui*.*res*.*label* *=* 'Заполните первое и второе поле'

*end*

*end*

*function* *ui*.btn\_div:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*b* *=* tonumber(*ui*.*num2*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *and* type(tonumber(*b*)) *==* 'number' *then*

*if* *b* *==* 0 *then*

*ui*.*res*.*label* *=* 'Нельзя делить на ноль'

*else*

*ui*.*res*.*label* *=* *a* */* *b*

*end*

*else*

*ui*.*res*.*label* *=* 'Заполните первое и второе поле'

*end*

*end*

*function* *ui*.btn\_pow:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*b* *=* tonumber(*ui*.*num2*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *and* type(tonumber(*b*)) *==* 'number' *then*

*ui*.*res*.*label* *=* *a* *^* *b*

*else*

*ui*.*res*.*label* *=* 'Заполните первое и второе поле'

*end*

*end*

*function* *ui*.btn\_cos:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *then*

*ui*.*res*.*label* *=* math.cos(math.rad(*a*))

*else*

*ui*.*res*.*label* *=* 'Заполните первое поле'

*end*

*end*

*function* *ui*.btn\_sin:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *then*

*ui*.*res*.*label* *=* math.sin(math.rad(*a*))

*else*

*ui*.*res*.*label* *=* 'Заполните первое поле'

*end*

*end*

*function* *ui*.btn\_tg:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *then*

*ui*.*res*.*label* *=* math.tan(math.rad(*a*))

*else*

*ui*.*res*.*label* *=* 'Заполните первое поле'

*end*

*end*

*function* *ui*.btn\_sqrt:*on\_clicked*()

*a* *=* tonumber(*ui*.*num1*.*text*)

*if* type(tonumber(*a*)) *==* 'number' *then*

*if* *a* *<* 0 *then*

*ui*.*res*.*label* *=* 'Число не должно быть отрицательным'

*else*

*ui*.*res*.*label* *=* math.sqrt(*a*)

*end*

*else*

*ui*.*res*.*label* *=* 'Заполните первое поле'

*end*

*end*

*gtk*.*main*()

1. Результаты:

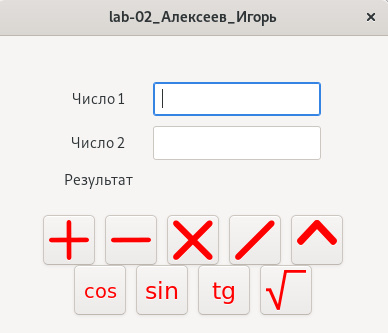


Рисунок 11 – Запущенная программа

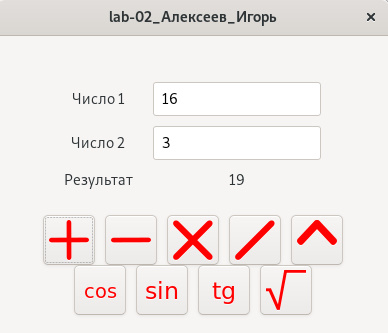


Рисунок 12 – Сложение

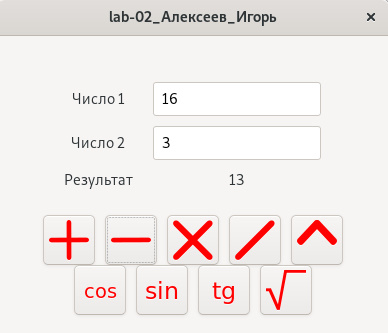


Рисунок 13 – Вычитание

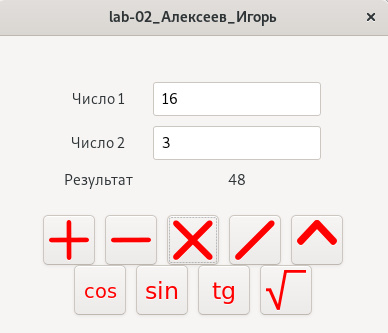


Рисунок 14 – Умножение

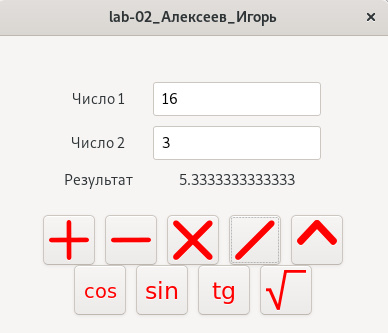


Рисунок 15 – Деление

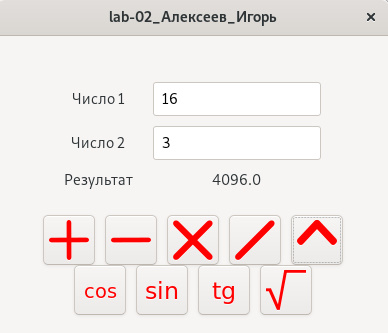


Рисунок 16 – Степень



Рисунок 17 – Косинус



Рисунок 18 – Тангенс



Рисунок 19 – Синус

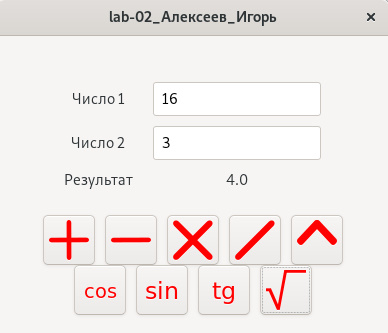
**

Рисунок 20 – Корень квадратный

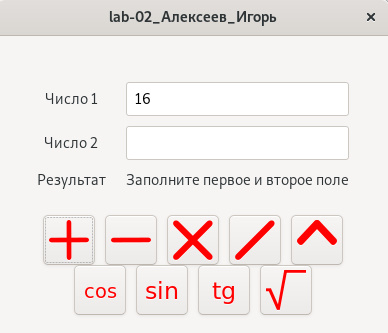
**

Рисунок 21 – Ошибка при нажатии верхних кнопок

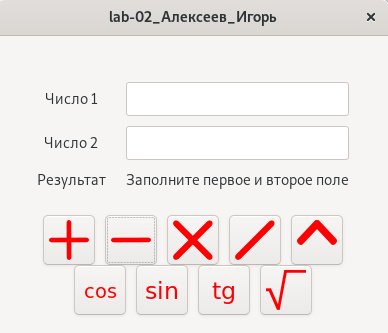
**

Рисунок 22 – Ошибка при нажатии верхних кнопок

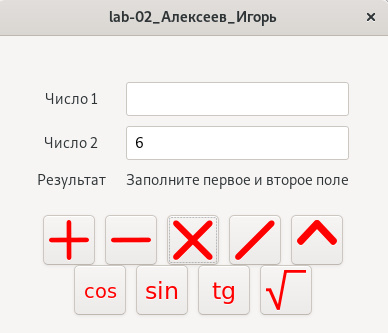
**

Рисунок 23 – Ошибка при нажатии верхних кнопок

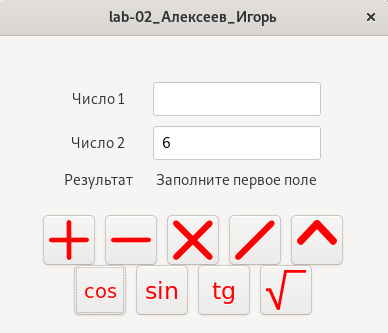


Рисунок 24 – Ошибка при нажатии нижних кнопок

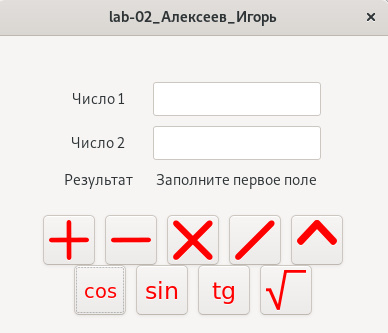


Рисунок 25 – Ошибка при нажатии нижних кнопок

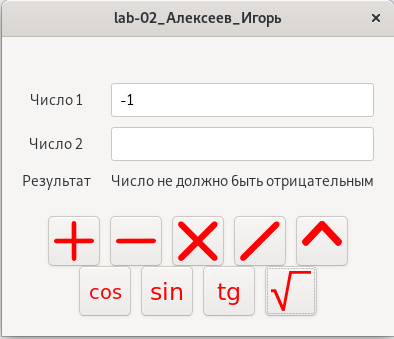
**

Рисунок 26 – Ошибка при извлечения квадратного корня

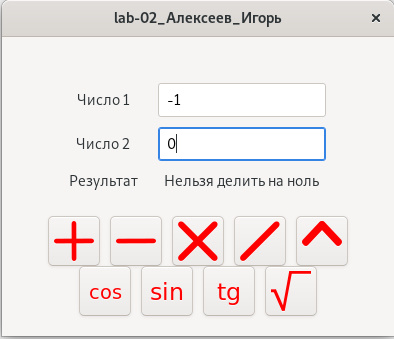


Рисунок 27 – Ошибка при делении

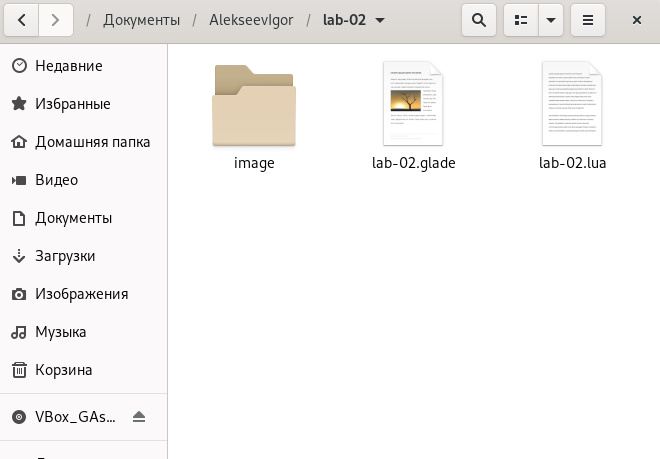


Рисунок 28 – Файлы программы

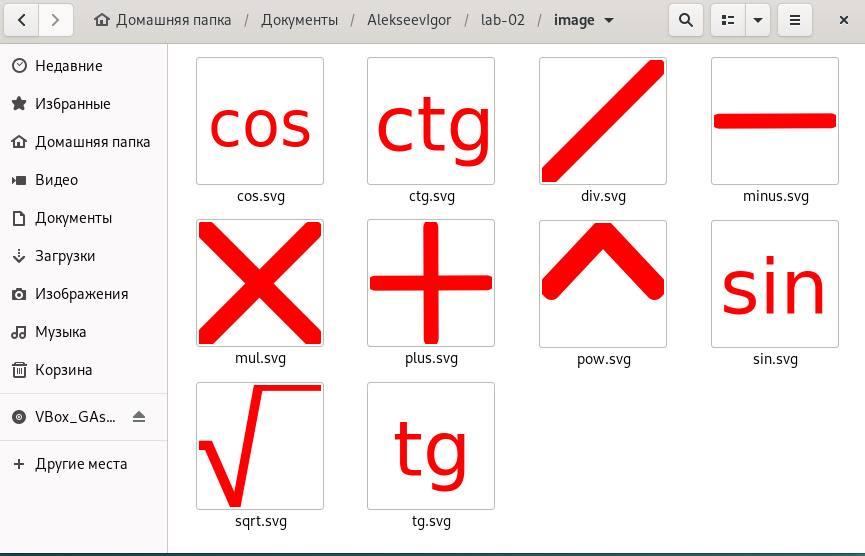


Рисунок 29 – Картинки

Ссылка на Github: