

*Грузин Никита Андреевич, студент, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан*

*Голубничий Артем Александрович, научный руководитель, старший преподаватель кафедры ПОВТуАС, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан*

## **ОБЗОР И СРАВНЕНИЕ БИБЛИОТЕК ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ: GTK, QT, WXWIDGETS**

**Аннотация:** в статье содержится краткий обзор библиотек построения пользовательских интерфейсов. Рассматриваются 3 кроссплатформенные библиотеки GUI: GTK+, Qt и wxWidgets. Представлены небольшие примеры программ на каждой из них.

**Ключевые слова:** библиотека пользовательского интерфейса, GUI, C++, GTK+, Qt, wxWidgets

**Abstract:** The article provides a brief overview of user interface building libraries. Three cross-platform GUI libraries are considered: GTK +, Qt, and wxWidgets. Small examples of programs on each of them are presented.

**Keywords:** user interface library, GUI, C ++, GTK +, Qt, wxWidgets

Одно из решений, которое мы должны принять при создании пользовательского интерфейса программ для различных устройств, это какие графические библиотеки можно использовать. В Open source пространстве есть несколько вариантов из которых можно выбрать [1]. Есть много факторов, на основе которых мы выбираем правильную графическую библиотеку для приложения. Вот некоторые из них:

- Должно ли приложение работать на Windows, macOS, Linux, Android или быть кроссплатформенным?
- Нужны ли приложению динамические эффекты и анимации, такие как в iOS или Android?
- Приложение должно работать на CPU или GPU?
- Требуется ли приложению поддержка интернационализации?
- Нужно ли приложению выглядеть и ощущаться интуитивно понятным?
- Какой язык будет использоваться?

В этой статье мы рассмотрим три кроссплатформенных инструментария построения GUI.

**GTK+.** GTK+ (GIMP Toolkit) – это кроссплатформенный набор инструментов для создания графических пользовательских интерфейсов, в основном используется в таких средах рабочего стола, как GNOME, Xfce, LXDE, ABIWORD, GNUMERIC, ROX и т.д. Его также можно использовать на Windows, macOS и мобильных устройствах [2].

GTK + является одним из самых популярных наборов инструментов для X Window System. Первоначально он был создан для программы обработки изображений GNU (GIMP), редактора растровой графики, в 1997 году Спенсером Кимбаллом и Питером Маттисом, членами eXperimental Computing Facility (XCF) в Калифорнийском университете в Беркли. Он используется в качестве стандартного инструментария для сред рабочего стола GNOME, LXDE и Xfce [2].

GTK+ был создан с использованием языка C, но также позволяет программировать на таких языках, как C#, Java, Ruby, Perl, PHP или Python через привязки.

В следующем коде представлен пример программы «Hello, world» на языке C. В этой программе есть окно с заголовком «Hello» и кнопка с похожим текстом. В самом начале программы мы добавляем библиотеку GTK к проекту (рисунок 1).

```
#include <gtk/gtk.h>
```

Рисунок 1 – Добавление GTK к программе

Далее мы описываем структуру формы приложения. Сначала иницилируем само окно и кнопку. Потом мы задаём название и размер окна приложения, а также добавляем кнопку на форму и задаём её параметры. После этого делаем все объекты видимыми (рисунок 2).

```
static void activate(GtkApplication *app, gpointer user_data)
{
    GtkWidget *window;
    GtkWidget *button;
    GtkWidget *button_box;

    window = gtk_application_window_new(app);
    gtk_window_set_title(GTK_WINDOW (window), "Window");
    gtk_window_set_default_size(GTK_WINDOW (window), 200, 200);

    button_box = gtk_button_box_new(GTK_ORIENTATION_HORIZONTAL);
    gtk_container_add(GTK_CONTAINER window), button_box);

    button = gtk_button_new_with_label ("Hello World");
    g_signal_connect(button, "clicked", G_CALLBACK(print_hello), NULL);
    g_signal_connect_swapped(button, "clicked", G_CALLBACK(gtk_widget_destroy), window);
    gtk_container_add(GTK_CONTAINER(button_box), button);

    gtk_widget_show_all(window);
}
```

Рисунок 2 – Описание объектов формы

В main уже инициализируется само приложение, что бы оно запустилось (рисунок 3). На рисунке 4 показано окно, которое получилось в результате сборки и компиляции.

```
int main(int argc, char **argv)
{
    GtkApplication *app;
    int status;

    app = gtk_application_new("org.gtk.example", G_APPLICATION_FLAGS_NONE);
    g_signal_connect(app, "activate", G_CALLBACK (activate), NULL);
    status = g_application_run(G_APPLICATION (app), argc, argv);
    g_object_unref(app);

    return status;
}
```

Рисунок 3 – Основной код программы

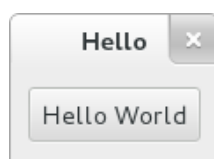


Рисунок 4 – Hello world на GTK

**Qt.** Qt – это мультиплатформенная библиотека для разработки графических пользовательских интерфейсов, а также консольных приложений [3]. Qt, созданный на языке программирования C++, может дополнительно использоваться в других языках программирования через привязки: Python, C#.NET, Ruby, Java, Ada, PHP и т.д.

Он работает на всех основных платформах и пользуется широкой поддержкой как компаний, так и исследовательского сообщества. Qt не только позволяет создавать графический интерфейс, но также предлагает API библиотеки, с методами доступа к базам данных через SQL, а также с использованием XML, управлением потоками, поддержкой сети, манипулированием файлами и т.д.

В коде ниже представлен пример программы «Hello, world» на языке C++. В этой программе есть окно с заголовком «Hello» и label с похожим текстом. В header файле объявляется класс формы и создаётся виджет, в который можно добавлять другие элементы интерфейса (рисунок 5).

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = 0);
    ~MainWindow();
};

#endif // MAINWINDOW_H
```

## Рисунок 5 – header файл

Далее мы объявляем header файл и некоторые модули для приложения (рисунок 6). В main мы инициализируем основное окно, добавляем label с текстом «Hello, World!» и делаем всё видимым (рисунок 7). В результате получится окно как на рисунке 8.

```
#include "mainwindow.h"
#include <QApplication>
#include <QLabel>
```

Рисунок 6 – Объявление header файла и остальных модулей

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;

    QLabel *label = new QLabel("Hello, World!");
    label->show();

    w.show();

    return a.exec();
}
```

Рисунок 7 – Основной код программы

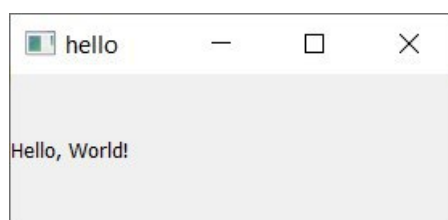


Рисунок 8 – Hello world на Qt

**wxWidgets.** wxWidgets – это библиотека C++ с открытым исходным кодом, которая позволяет разработчикам создавать приложения для 32 и 64-разрядных архитектур Windows, macOS, Linux и UNIX, а также для нескольких мобильных платформ.

Он имеет языковые привязки для Python, Perl, Ruby и многих других языков. В отличие от других многоплатформенных наборов инструментов, wxWidgets придает приложениям естественный вид, поскольку использует собственный API платформы вместо эмуляции графического интерфейса пользователя.

В коде ниже представлен пример программы «Hello, world» на языке C++. В header файле объявляется библиотека wx и создается основной класс (рисунок 9).

```
#ifndef HELLOAPP_H
#define HELLOAPP_H

#include <wx/wx.h>

class HelloApp : public wxApp {
public:
    HelloApp();
    virtual ~HelloApp();
    virtual bool OnInit();
};

DECLARE_APP(HelloApp)
#endif
```

Рисунок 9 – header файл

В основном теле программы мы добавляем текст на форму и «приклеиваем» его в верхней части программы и делаем его видимым (рисунок 10). На рисунке 11 показано окно программы.

```

#include "HelloApp.h"
#include "gui/guifrmMain.h"

IMPLEMENT_APP(HelloApp)

HelloApp::HelloApp() {}
HelloApp::~HelloApp() {}

bool HelloApp::OnInit() {
    guifrmMain* frame = new guifrmMain((wxWindow*)NULL);
    frame->Show();
    SetTopWindow(frame);
    return true;
}

```

Рисунок 10 – Основной код программы

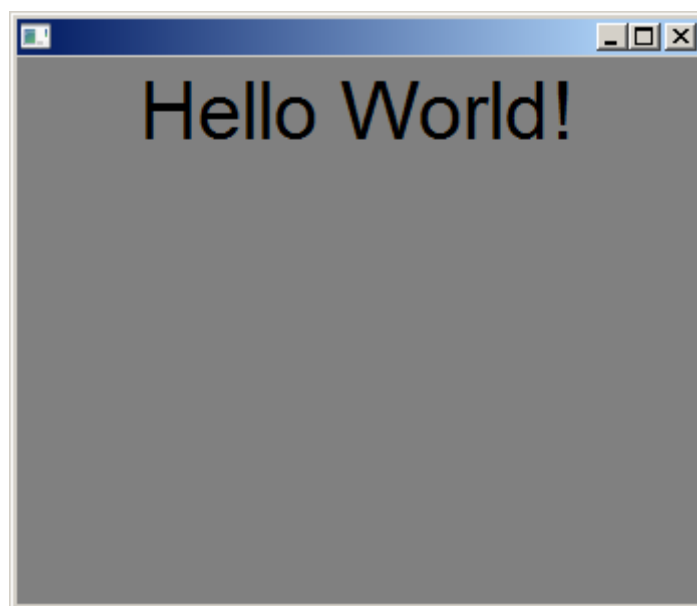


Рисунок 11 – Hello world на wxWidgets

В данной статье были рассмотрены различные графические библиотеки для разработки приложений. Которые являются кроссплатформенными и имеют возможность писать на почти любом удобном языке программирования.

Так же были рассмотрены небольшие примеры программ на всех библиотеках.

### **Библиографический список:**

1. Qt Vs GTK+ Vs WxWidgets - A Comparative Study | System on Module Blog [Электронный ресурс] URL: <https://www.e-consystems.com/blog/system-on-module-SOM/qt-vs-gtk-vs-wxwidgets-a-comparative-study/> (дата обращения: 29.01.2020).
2. Getting Started with GTK+: GTK +3 Reference Manual [Электронный ресурс] URL: <https://developer.gnome.org/gtk3/stable/gtk-getting-started.html> (дата обращения: 29.01.2020).
3. Qt Documentation [Электронный ресурс] URL: <https://doc.qt.io> (дата обращения: 29.01.2020).
4. wxWidgets: Cross-Platform GUI Library [Электронный ресурс] URL: <https://www.wxwidgets.org> (дата обращения: 29.01.2020).