Торайғыров университетінің
ҒЫЛЫМИ ЖУРНАЛ

НАУЧНЫЙ ЖУРНАЛ
Торайгыров университета

ТОРАЙҒЫРОВ УНИВЕРСИТЕТІНІҢ ХАБАРШЫСЫ

Физика, математика және компьютерлік ғылымдар сериясы
1997 жылдан бастап шығады

TORAIGHYROV UNIVERSITY

ВЕСТНИК ТОРАЙГЫРОВ УНИВЕРСИТЕТА

Серия: Физика, математика и компьютерные науки
Издается с 1997 года

ISSN 2959-068X

№ 4 (2023)
Павлодар

НАУЧНЫЙ ЖУРНАЛ
ТОРАЙГЫРОВ УНИВЕРСИТЕТА

Серия: Физика, математика и компьютерные науки
выходит 4 раза в год

СВИДЕТЕЛЬСТВО
о постановке на переучет периодического печатного издания,
информационного агентства и сетевого издания
№ KZ91VPY00046988
выдано
Министерством информации и общественного развития
Республики Казахстан

Тематическая направленность
публикация материалов в области физики, математики,
механики и информатики

Подписной индекс – 76208

https://doi.org/10.48081/EKIW3862

МРНТИ 50.05.09

https://doi.org/10.48081/JEDU8308

С. Н. Талипов
Торайгыров университет, Республика Казахстан, г. Павлодар
*e-mail: talipovsn@gmail.com

ВЫБОР ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ И ФРЕЙМВОРКА ДЛЯ СОЗДАНИЯ ДЕСКТОПНЫХ КРОССПЛАТФОРМЕННЫХ ПРОГРАММ

Цель в данной статье исследовать и определить, какой из современных языков программирования и графических фреймворков лучше всего подходит для создания десктопных кроссплатформенных программ, т. к. в мире широко используются компьютеры не только с Windows, но и с Linux, macOS. В начале было проанализировано, на чем написаны современные программы под Windows. Для анализа были взяты распространённые программы на стандартном рабочем месте разработчика программного обеспечения (ПО). Сигнатурный анализ показал, что 90 % программ написаны на C++ в Visual Studio с помощью компилятора Microsoft, остальные 10 % делят поровну между собой языки C# и Delphi. Анализ использования кроссплатформенных графических фреймворков показал, что только 15 % программ их используют, и это говорит о том, что многие разработчики ПО не заинтересованы в использовании их программ в других операционных системах, кроме Windows. Три выявленных кроссплатформенных графических фреймворков имеют равный процент использования, т. е. нет явного лидера среди них. Использование фреймворка Qt платно для коммерческого использования и очень дорого, т. к. нужно платить за лицензии как всем разработчикам ПО, так и за использование в каждом устройстве для встраиваемого ПО. Фреймворк Electron при своей бесплатности и кроссплатформенности имеет существенные недостатки: ресурсоемкость, размер приложений, ограниченный доступ к системным ресурсам и ограниченная поддержка для нативных функций. Кроссплатформенный фреймворк wxWidgets лишен всех вышеперечисленных недостатков, дает нативный интерфейс приложениям, которые также получаются нативными под конкретную операционную систему. Таким образом, в результате выявлено, что наилучшим выбором для разработки современного

кроссплатформенного ПО является язык программирования C++ и графический кроссплатформенный фреймворк wxWidgets.

Ключевые слова: программное обеспечение, язык программирования, программный код, фреймворк, графический интерфейс, консольная программа.

Введение

В настоящее время в мире широко используются компьютеры не только с Windows, но и с Linux, macOS. Поэтому современный разработчик программного обеспечения должен делать программы, рассчитанные на все основные операционные системы (ОС), чтобы охватить весь рынок пользователей.

Поэтому разработчику изначально очень важно выбрать правильный язык разработки программного обеспечения и дополнительные фреймворки для создания графического интерфейса у программ.

Материалы и методы

Для начала проанализируем с помощью анализатора программного кода Detect-It-Easy [4] на чем написаны современные программы под Windows. Для конкретного анализа возьмем 20 распространённых программ на стандартном рабочем месте разработчика программного обеспечения (ПО).

Результат проведенного сигнатурного анализа программ приведен в таблице 1:

Таблица1 – Результат сигнатурного анализа программ

№	Название ПО	Основной язык разработки			Фреймворк		
		C++	C#	Delphi	Qt	wxWidgets	Electron
	Acrobat Reader	VS2019					
	Far Manager	VS2019					
	GetScreen	VS2022					
	Java	VS2022					
	Macrium Reflect	VS2017					
	Microsoft Office 365	VS2022					
	Microsoft OneDrive	VS2022					
	Visual Studio 2022	VS2022					
	MySQL Server	VS2019					

	MySQL Workbench		.NET4				
	Python	VS2022					
	qBittorrent	VS2022			+		
	TeamViewer	VS2019					
	WinDjView	VS2013					
	Windows Media	VS2022					
	WinRAR	VS2022					
	FastStone Image Viewer			Delphi 6			
	Jutoh	VS2010				+	
	Kaspersky Antivirus	VS2019					
	Microsoft Edge	VC++16					+
	Итого:	18	1	1	1	1	1

Сигнатурный анализ ПО показал, что 18 программ из 20 написаны на C++ [2], а это 90 %. Остальные 10 % делят поровну между собой языки C# [1] и Delphi [3]. Это говорит о том, что практически все приложения разрабатываются на C++ в среде разработки Visual Studio [8] с помощью компилятора Microsoft.

Анализ использования кроссплатформенных графических фреймворков показал, что только три программы их используют, а это всего 15%. Это говорит о том, что многие производители ПО не заинтересованы в использовании их программ в других ОС, кроме Windows. Причем три фреймворка имеют равный процент использования, т.е. нет явного лидера среди них.

Использование фреймворка Qt [7] платно для коммерческого использования и очень дорого, т.к. нужно платить за лицензии как всем разработчикам ПО, так и за использование в каждом устройстве для встраиваемого ПО. Поэтому данный фреймворк не подходит для небольших компаний и индивидуальных разработчиков.

Фреймворк Electron [6] при своей бесплатности и кроссплатформенности имеет существенные недостатки: ресурсоемкость, размер приложений, ограниченный доступ к системным ресурсам и ограниченная поддержка для нативных функций. Поэтому его нецелесообразно использовать.

Кроссплатформенный фреймворк wxWidgets [10] лишен всех вышеперечисленных недостатков, дает нативный интерфейс приложениям, которые также получаются нативными под конкретную операционную

систему. Фреймворк wxWidgets позволяет разрабатывать программы не только на C++, но и на wxPython [9].

Результаты и обсуждение

Таким образом, анализ показал, что наилучшим выбором для разработки современного кроссплатформенного ПО является язык C++ и графический фреймворк wxWidgets.

Рассмотрим, как правильно нужно делать программу на C++ для кроссплатформенного использования с wxWidgets на примере консольной программы [11].

Простейший код, работающий для Windows:

```
#include <clocale>
#include <string>
#include <iostream>
using namespace std;
int main() {
  setlocale(LC_ALL, “Russian”);
  string fio;
  cout << «Введите строку: «;
  cin >> fio;
  cout << «Вы ввели: « << fio << endl;
}
```

Данный код будет непригоден для использования интернациональных символов UTF-8 [13], например, китайских иероглифов, вместо них будут выводиться знаки вопроса. Зато под Linux данный код будет работать нормально.

Чтобы вышеприведенная программа стала работоспособна и под Windows и под Linux, нужно использовать теги условной компиляции, широкие строки для хранения данных и соответствующие функции для работы с ними. А также не забывать переключать консоль Windows в режим поддержки кодировки UTF-8.

Универсальный правильный код будет такой:

```
#include <stdlib.h>
#include <iostream>

#ifdef _WIN32
#include <io.h>
#include <fcntl.h>
```

```
#endif

int main(int argc, char** argv) {
  setlocale(LC_ALL, “ru_RU.UTF-8”);

#ifdef _WIN32
  _setmode(_fileno(stdout), _O_U16TEXT);
  _setmode(_fileno(stdin), _O_U16TEXT);
  _setmode(_fileno(stderr), _O_U16TEXT);
#endif

  std::wcout << L»Введите строку: «;
  std::wstring fio;
  std::wcin >> fio;
  std::wcout << L”Вы ввели: “ << fio << std::endl;
}
```

Если мы захотим сделать эту программу с локализацией под разные языки и страны, то нужно использовать фреймворк wxWidgets, т. к. в нем уже есть встроенные механизмы локализации и интернационализации.

Интернационализированный кроссплатформенный программный код будет таким:

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

#ifdef _WIN32
#include <io.h>
#include <fcntl.h>
#endif

int main(int argc, char** argv) {
  wxApp myApp;
  wxInitializer initializer(argc, argv);
  wxLocale m_locale;
  setlocale(LC_ALL, “ru_RU.UTF-8”);
  m_locale.AddCatalogLookupPathPrefix(wxT(“locale”));
  m_locale.AddCatalog(wxT(“en”));
  m_locale.AddCatalog(wxT(“de”));
  m_locale.AddCatalog(wxT(“kk”));
  m_locale.Init(wxLANGUAGE_RUSSIAN);
```

```
#ifdef _WIN32
   _setmode(_fileno(stdout), _O_U16TEXT);
   _setmode(_fileno(stdin), _O_U16TEXT);
   _setmode(_fileno(stderr), _O_U16TEXT);
#endif

   wxPrintf(wxGetTranslation(L"Введите имя: "));
   std::wstring fio;
   std::getline(std::wcin, fio);
   wxPuts(wxGetTranslation(L"Вы ввели: ") + fio);
   myApp.Exit();
}
```

Для разработки современных кроссплатформенных программ с графическим интерфейсом [12] на языке C++ и wxWidgets проще всего использовать интегрированную среду разработки (IDE) DialogBlocks [5] совместно с Visual Studio 2022. В DialogBlocks создается интерфейс программы и программный код, который синхронно можно редактировать в Visual Studio 2022. Использование Visual Studio 2022 позволяет использовать ее компилятор C++, анализаторы и помощники кода. В самой программе нужно максимально задействовать встроенные в wxWidgets типы данных, классы, функции и методы для кроссплатформенности и независимости от конкретной ОС.

Рассмотрим простейшую визуальную кроссплатформенную программу деления двух цифр. Ее интерфейс в DialogBlocks и элементы графического интерфейса приведены на рисунке 1.

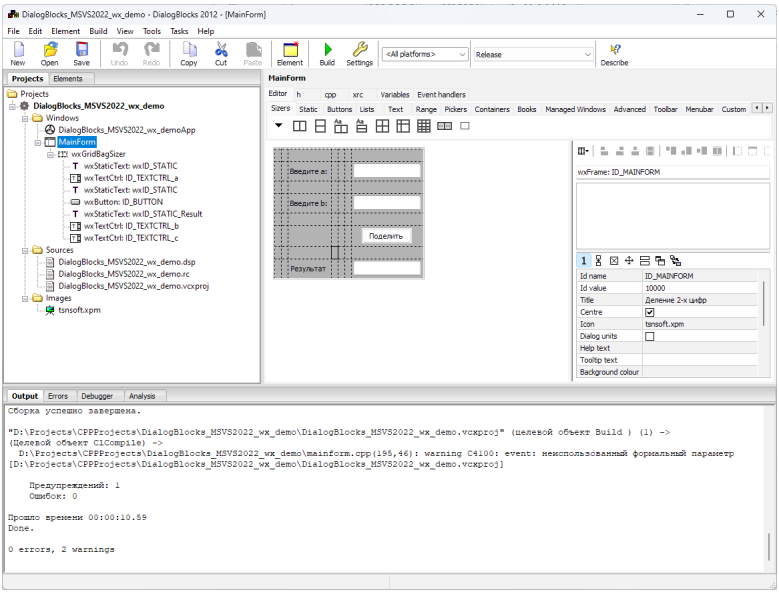

Рисунок 1 – Интерфейс программы в DialogBlocks

Если программу скомпилировать и запустить в Window 11, то она примет вид, показанный на рисунке 2:

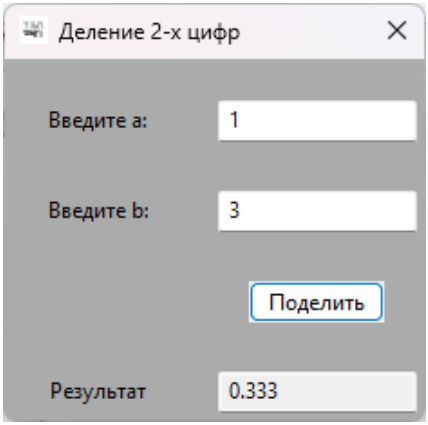

Рисунок 2 – Вид программы в Windows 11

Если программу скомпилировать и запустить в Linux Ubuntu 22, то она примет вид, показанный на рисунке 3:

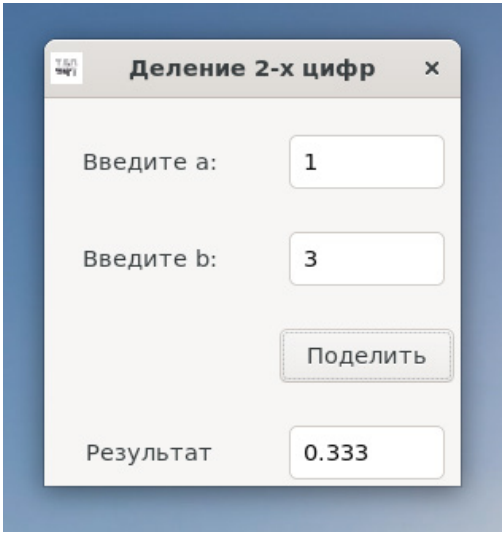

Рисунок 3 – Вид программы в Ubuntu 22

Как видно из рисунка 2 и рисунка 3, программа в каждой ОС выглядит по-своему, нативно для этой среды.

Фрагмент кода, отвечающий за сам алгоритм вычисления, такой:

```
void MainForm::OnButtonClick(wxCommandEvent& event) {
  double a, b;
  wxTextCtrl* itemTextCtrl1 = (wxTextCtrl*)FindWindow(ID_
TEXTCTRL_a);
  wxTextCtrl* itemTextCtrl2 = (wxTextCtrl*)FindWindow(ID_
TEXTCTRL_b);
  wxTextCtrl* itemTextCtrl3 = (wxTextCtrl*)FindWindow(ID_
TEXTCTRL_c);
  if (!itemTextCtrl1->GetValue().ToDouble(&a)) {
    wxMessageBox(wxT("a не число!")); return;
  }
  if (!itemTextCtrl2->GetValue().ToDouble(&b)) {
    wxMessageBox(wxT("b не число!")); return;
  }
```

```
    double c = a / b;
    if (isnan(c) || isinf(c)) {
        wxMessageBox(wxT(«результат не число!»)); return;
    }
    itemTextCtrl3->SetValue(wxString::Format(wxT(“%0.3f”), c));
}
```

Выводы

В заключении хочется отметить, что кроме языка C++ и wxWidgets в настоящее время больше нет способов создавать нативные бесплатные кроссплатформенные десктопные программы. Языки Java, C# и wxPython не делают нативных программ, Delphi и Qt делают, но они очень дороги.

Поэтому необходимо язык C++ и фреймворк wxWidgets активно изучать и использовать в разработке современного ПО.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 C Sharp [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C_Sharp

2 C++ [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C++

3 Delphi 12 [Электронный ресурс] – Режим доступа: https://www.embarcadero.com/ru/products/delphi

4 Detect-It-Easy [Электронный ресурс] – Режим доступа: https://github.com/horsicq/Detect-It-Easy

5 DialogBlocks [Электронный ресурс] – Режим доступа: http://www.anthemion.co.uk/dialogblocks/

6 Electron [Электронный ресурс] – Режим доступа: https://www.electronjs.org/

7 Qt [Электронный ресурс] – Режим доступа: https://www.qt.io/

8 Visual Studio 2022 [Электронный ресурс] – Режим доступа: https://visualstudio.microsoft.com

9 wxPython [Электронный ресурс] – Режим доступа: https://wxpython.org/

10 wxWidgets [Электронный ресурс] – Режим доступа: https://www.wxwidgets.org/

11 Консольные приложения [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Console_application

12 Оконный интерфейс [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Оконный_интерфейс

13 Юникод [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Юникод

Принято к изданию 15.12.23

REFERENCES

1 C Sharp [Electronic resource]. – Access mode: https://ru.wikipedia.org/wiki/C_Sharp

2 C++ [Electronic resource]. – Access mode: https://ru.wikipedia.org/wiki/C++

3 Delphi 12 [Electronic resource]. – Access mode: https://www.embarcadero.com/ru/products/delphi

4 Detect-It-Easy [Electronic resource]. – Access mode: https://github.com/horsicq/Detect-It-Easy

5 DialogBlocks [Electronic resource]. – Access mode: http://www.anthemion.co.uk/dialogblocks/

6 Electron [Electronic resource]. – Access mode: https://www.electronjs.org/

7 Qt [Electronic resource]. – Access mode: https://www.qt.io/

8 Visual Studio 2022 [Electronic resource]. – Access mode: https://visualstudio.microsoft.com

9 wxPython [Electronic resource]. – Access mode: https://wxpython.org/

10 wxWidgets [Electronic resource]. – Access mode: https://www.wxwidgets.org/

11 Console applications [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Console_application

12 Graphical user interface [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Graphical_user_interface

13 Unicode [Electronic resource]. – Access mode: https://en.wikipedia.org/wiki/Unicode

С. Н. Талипов
Торайғыров университеті, Қазақстан Республикасы, Павлодар қ.
Басып шығаруға 15.12.23 қабылданды.

ЖҰМЫС ҮСТЕЛІ КРОСС-ПЛАТФОРМАСЫ БАҒДАРЛАМАЛАРЫН ЖАСАУҒА АРНАЛҒАН ҚҰРАЛДАР МЕН НЕГІЗДЕРДІ ТАҢДАУ

Бұл мақаланың мақсаты қазіргі заманғы бағдарламалау тілдері мен графикалық негіздердің қайсысы жұмыс үстелінің

кросс-платформалық бағдарламаларын жасауға ең қолайлы екенін зерттеу және анықтау болып табылады, себебі компьютерлер әлемде Windows жүйесімен ғана емес, Linux және macOS-пен де кеңінен қолданылады. Басында Windows үшін қандай заманауи бағдарламалар жазылғанын талдадық. Талдау үшін біз стандартты бағдарламалық жасақтама әзірлеушінің жұмыс станциясында ортақ бағдарламаларды қабылдадық. Қолтаңба талдауы көрсеткендей, бағдарламалардың 90 %- ы Visual Studio-да Microsoft компиляторын пайдаланып C++ жазылса, қалған 10 %- ы C# және Delphi арасында тең бөлінген. Кросс-платформалық графикалық қаңқаларды пайдалануды талдау оларды бағдарламалардың тек 15 %-ы ғана пайдаланатынын көрсетті, бұл көптеген бағдарламалық жасақтама әзірлеушілердің windows-тан басқа операциялық жүйелерде өз бағдарламаларын пайдалануға мүдделі емес екенін көрсетеді. Анықталған үш кросс-платформалық графикалық негіздің пайдалану пайызы тең, яғни олардың арасында нақты көшбасшы жоқ. Qt шеңберін пайдалану коммерциялық пайдалану үшін төленеді және өте қымбат, себебі барлық бағдарламалық жасақтаманы әзірлеушілер үшін де, ендірілген бағдарламалық жасақтама үшін әр құрылғыда пайдалану үшін де лицензиялар үшін ақы төлеу қажет. Электрон шеңбері өзінің еркін және кросс-платформалық сипатына қарамастан елеулі кемшіліктерге ие: ресурстардың қарқындылығы, қолдану мөлшері, жүйе ресурстарына шектеулі қолжетімділік, сондай-ақ туған функцияларды қолдаудың шектеулілігі. WxWidgets кросс-платформасының шеңбері жоғарыда көрсетілген барлық кемшіліктерден ауытқыған, ол сондай-ақ белгілі бір операциялық жүйе үшін туған болып шығатын қосымшаларға туған интерфейс береді. Сөйтіп, осы заманғы кросс-платформалық бағдарламалық қамтамасыз етуді дамыту үшін ең жақсы таңдау C++ бағдарламалау тілі және wxWidgets графикалық кросс-платформа шеңбері екені анықталды.

Кілтті сөздер: бағдарламалық жасақтама, бағдарламалау тілі, программа коды, шеңбер, графикалық интерфейс, консольдық бағдарлама.

S. N. Talipov
Toraigyrov University, Republic of Kazakhstan, Pavlodar
Accepted for publication 15.12.23

SELECTION OF TOOLS AND FRAMEWORK TO CREATE DESKTOP CROSS-PLATFORM PROGRAMS

The purpose of this article is to investigate and determine which of the modern programming languages and graphical frameworks is best suited for creating desktop cross-platform programs, because computers are widely used in the world not only with Windows, but also with Linux, macOS. At the beginning, it was analyzed what modern programs for Windows are written on. For the analysis, common programs were taken at the standard workplace of a software developer (software). Signature analysis showed that 90 % of programs are written in C++ in Visual Studio using the Microsoft compiler, the remaining 10 % are divided equally between the C# and Delphi languages. An analysis of the use of cross-platform graphics frameworks showed that only 15 % of programs use them, and this suggests that many software developers are not interested in using their programs in other operating systems other than Windows. The three identified cross-platform graphics frameworks have an equal percentage of usage, i.e. there is no clear leader among them. The use of the Qt framework is paid for commercial use and is very expensive, because you need to pay for licenses both for all software developers and for use in each device for embedded software. The Electron framework, despite its free and cross-platform nature, has significant drawbacks: resource intensity, application size, limited access to system resources and limited support for native functions. The wxWidgets cross-platform framework is devoid of all the above disadvantages, it gives a native interface to applications that are also native for a specific operating system. Thus, as a result, it was revealed that the best choice for the development of modern cross-platform software is the C++ programming language and the wxWidgets graphical cross-platform framework.

Keywords: software, programming language, program code, framework, graphical interface, console program.

Теруге 15.12.2023 ж. жіберілді. Басуға 29.12.2023 ж. қол қойылды.
Электрондық баспа
7,50 Mb RAM
Шартты баспа табағы 10,01. Таралымы 300 дана. Бағасы келісім бойынша.
Компьютерде беттеген: Е. Е. Калихан
Корректор: А. Р. Омарова
Тапсырыс № 4178

Сдано в набор 15.12.2023 г. Подписано в печать 29.12.2023 г.
Электронное издание
7,50 Mb RAM
Усл.печ.л. 10,01. Тираж 300 экз. Цена договорная.
Компьютерная верстка Е. Е. Калихан
Корректор: А. Р. Омарова
Заказ № 4178

«Toraighyrov University» баспасынан басылып шығарылған
«Торайғыров университеті» КЕ АҚ
140008, Павлодар қ., Ломов к., 64, 137 каб.

«Toraighyrov University» баспасы
«Торайғыров университеті» КЕ АҚ
140008, Павлодар қ., Ломов к., 64, 137 каб.
+7(718)267-36-69

e-mail: kereku@tou.edu.kz
www.vestnik.tou.edu.kz
https://vestnik-pm.tou.edu.kz/