

Технологии и методы программирования

Часть 4

Ст. преподаватель
кафедры ПИВТ
Воронцова И.О.

2020 год

Библиотеки языка C++

- Стандартная библиотека C++
- Библиотека шаблонов STL
- Библиотеки для создания приложений с графическим интерфейсом пользователя (GUI): Win API, MFC, OWL, gtk+, Qt
- Библиотеки компонентов для быстрой разработки (RAD) VCL, CLX, ActiveX
- Библиотеки для создания мультимедийных приложений OpenGL, SDL, DirectX

Классификация библиотек по способу их подключения к приложению

- **Статические**
- Linux: ***.a** (Archive)
- Windows: ***.lib** (Library)
- **Динамические**
- Linux: ***.so** (Shared Object)
- Windows: ***.dll** (Dynamic Linked Library)

Стандартная библиотека языка C++

- Стандарты языка C++:

- ◆ 1998 год,
- ◆ C++11 (11.09.2011),
- ◆ C++ 2014 (14.01.2015),
- ◆ C++ 2017 (ISO/IEC 14882:2017).

- Состав библиотеки:

- ◆ Средства языка C в соответствии со стандартом C89 (ныне действующий стандарт – C11, декабрь 2011),
- ◆ Классы языка C++,
- ◆ Средства языка, унаследованные от библиотеки STL.

- Входит в состав инструментального средства (компилятора, IDE).

- Доступна всегда после подключения соответствующих заголовочных файлов.

Библиотека STL

STL (Standard Template Library) – Стандартная библиотека шаблонов

Разработчики: Александр Степанов, Дэвид Муссер, Менг Ли (Hewlett-Packard, начало 1990-х годов)

В состав библиотеки входят:

- ♦ **контейнеры**: строки, вектора, множества, списки, очереди
- ♦ **алгоритмы**: поиск, сортировка, слияние последовательностей и перестановка элементов
- ♦ **итераторы**: ввода, вывода, однонаправленные, двунаправленные, произвольного доступа

Заголовочные файлы STL

`<vector>` – одномерный массив элементов;

`<ctime>` – дата и время;

`<algorithm>` – основные алгоритмы;

`<cstdlib>` – функции обработки данных (поиск, сортировка, обработка строк в стиле C, генератор случайных чисел);

`<string>` – строка;

`<iostream>` – стандартные потоки ввода/вывода;

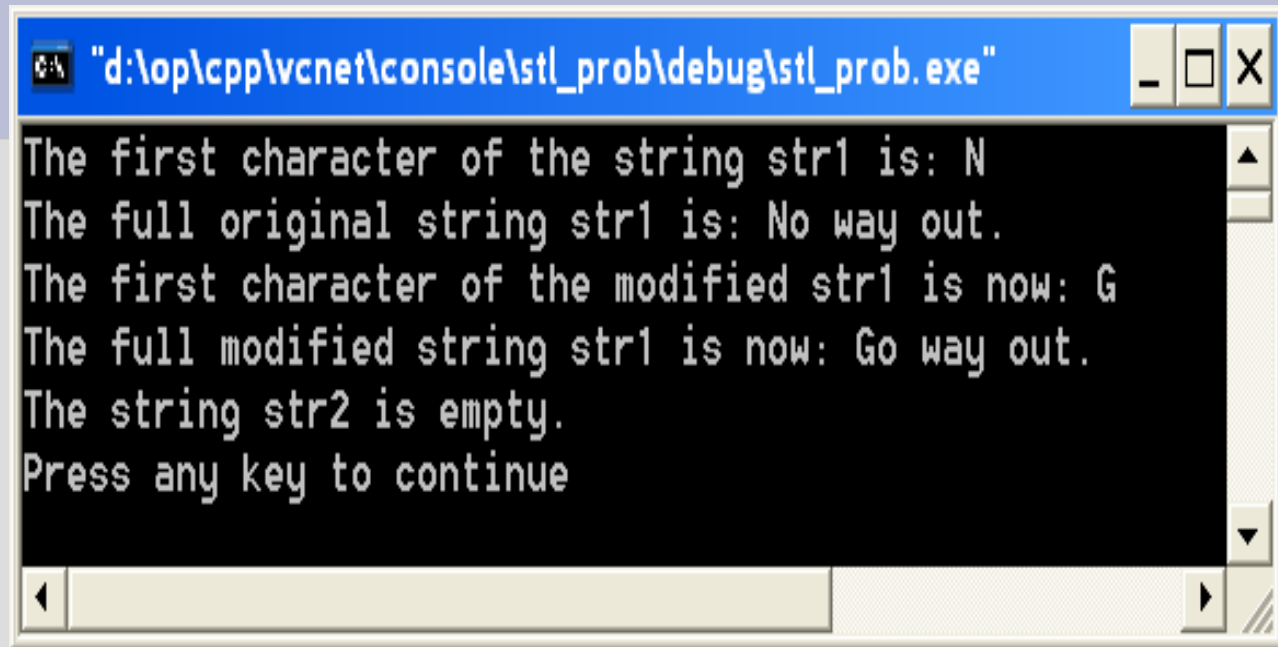
`<complex>` – комплексные числа;

`<c_math>` – общие математические функции;

`<new>` – работа с динамической памятью.

Пример использования библиотеки STL

```
#include "stdafx.h"
#include <string>
#include <iostream>
int _tmain(int argc, _TCHAR* argv[])
{
    using namespace std;
    string str1 ( "No way out." ), str2;
    basic_string <char>::iterator str1_iter;
```



```
"d:\op\cpp\vcnet\console\stl_prob\debug\stl_prob.exe"
The first character of the string str1 is: N
The full original string str1 is: No way out.
The first character of the modified str1 is now: G
The full modified string str1 is now: Go way out.
The string str2 is empty.
Press any key to continue
```

STL - продолжение

```
str1_iter = str1.begin ( );  
cout << "The first character of the string str1 is: " << *str1_iter  
<< endl;  
cout << "The full original string str1 is: " << str1 << endl;  
*str1_iter = 'G';  
cout << "The first character of the modified str1 is now: " <<  
*str1_iter << endl;  
cout << "The full modified string str1 is now: " << str1 << endl;  
  
if ( str2.begin ( ) == str2.end ( ) ) cout << "The string str2 is  
empty." << endl;  
else cout << "The string str2 is not empty." << endl;  
return 0;  
}
```


Пример программы со СПИСКОМ

```
#include <iostream>
#include <list>
#include <iterator>
using namespace std;

int main()
{
    list <int> mylist;
    int i;
    for(i=0;i<10;i++)    mylist.push_back(i*i);
    cout << "number of elements:" << mylist.size();
    if(!mylist.empty())
        cout<<"first:" << *(mylist.begin())<<" last:"<<*(mylist.end())<< endl;
    else    cout<<"list is empty!";
```

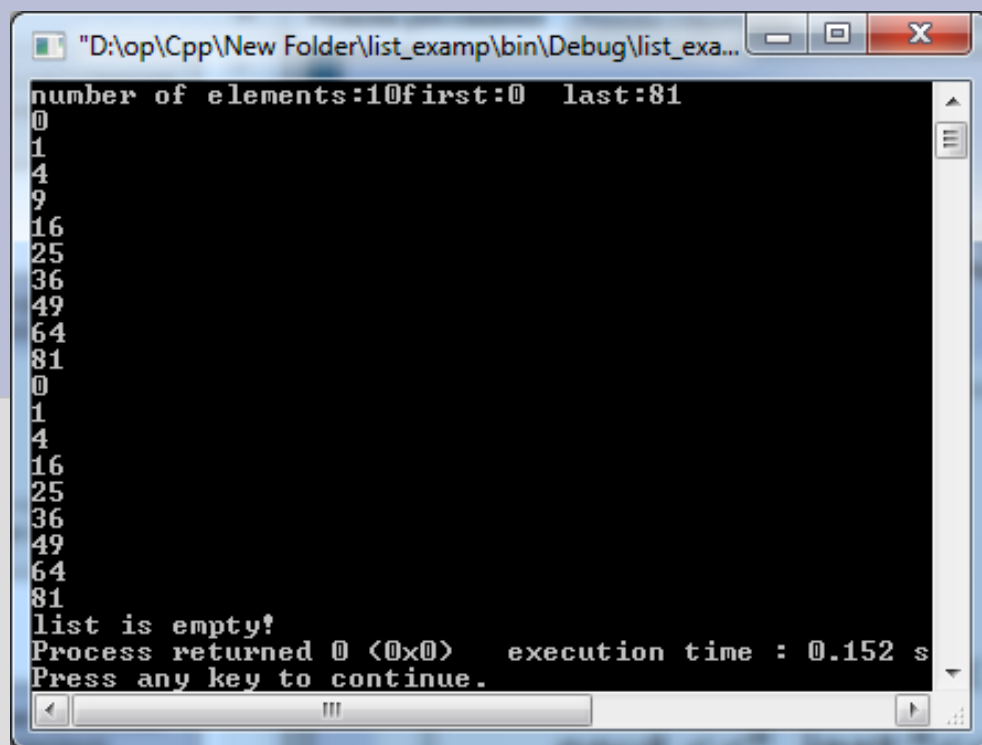
Пример программы со списком (2)

```
list<int>::iterator iter; // объявление итератора
for(iter=mylist.begin(); iter!=mylist.end();iter++)
    cout<<*iter<<endl;
for(iter=mylist.begin(),i=0;i<3;iter++,i++);
mylist.erase(iter); // удаление элемента списка

for(iter=mylist.begin();iter!=mylist.end();++iter)
    cout<<*iter<<endl;
```

Пример программы со списком (3)

```
// очистка списка
mylist.clear();
if(!mylist.empty())
{
    cout <<"first:" << *(mylist.begin());
    cout <<" last:"<<*(mylist.end())<< endl;
}
else    cout<<"list is empty!";
return 0;
}
```



```
"D:\op\Cpp\New Folder\list_examp\bin\Debug\list_exa...
number of elements:10first:0 last:81
0
1
4
9
16
25
36
49
64
81
0
1
4
16
25
36
49
64
81
list is empty!
Process returned 0 (0x0) execution time : 0.152 s
Press any key to continue.
```

Библиотека Win32 API

- Стандарт отсутствует
- Состав библиотеки:
 - Типы данных
 - Константы
 - Функции
- Разработчик: Microsoft
- Область использования: приложения для ОС Windows на языках C или C++ (консольные, оконные)
- Лицензия: коммерческий продукт
- Подключение через заголовочные файлы (windows.h)

Библиотека OWL

- Стандарт отсутствует
- Состав библиотеки:
 - ◆ Классы, представляющие собой надстройку над объектами Win16 и Win32 API, - TApplication, TDialog, TButton, TBrush
 - ◆ Классы общего назначения TRect
 - ◆ Структуры
 - ◆ Макросы
- Разработчик: Borland
- Область использования: оконные приложения для ОС Windows на языках C++, Pascal (Object Pascal)
- Лицензия: коммерческий продукт
- Подключение через заголовочные файлы

Библиотека OWL (продолжение)

- Библиотека предоставляет каркас прикладной программы
- Главная функция:

```
#include <owl\applicat.h>
```

```
int OwlMain (int argc, char** argv)
{
    TApplication app ("Application name");
    app.Run ();
    return 0;
}
```

Библиотека MFC

- Стандарт отсутствует
- Состав библиотеки:
 - Классы, представляющие собой надстройку над объектами Win32 API, - CApplication, CWnd, CDialog, CButton, CBrush
 - Классы общего назначения – CRect
 - Шаблоны
- Разработчик: Microsoft
- Область использования: оконные приложения для ОС Windows на языке C++
- Лицензия: коммерческий продукт
- Подключение через заголовочные файлы

Библиотека gtk+

- Стандарт отсутствует
- Состав библиотеки:
 - ◆ Функции, написанные на языке C
 - ◆ Макросы
 - ◆ Структуры и др. типы данных
- Разработчик: Open Source
- Последние версии – gtk+ 3.24 – 2018 год
- Область использования: оконные приложения для ОС Linux, Windows на языках C или C++
- Основа для рабочего окружения Linux (оболочек) GNOME, Xfce (), приложений Gimp, Inkscape, Chromium, MonoDevelop.
- Подключение через заголовочные файлы
- Требуется установка дистрибутива
- Лицензия: свободная
- Информация и дистрибутивы: www.gtk.org

Библиотека wxWidgets

Стандарт отсутствует.

- Текущая версия 3.1.2 (2018).
- Стабильная версия 3.0.4 (2018).
- Состав библиотеки: классы (wxApp, wxFrame), функции (connect).
- Разработчик: Юлиан Сمارт, Эдинбургский университет, 1992 год.
- Написана на C++.
- ОС: Linux, Windows, Mac OS X.
- Лицензия — бесплатная.
- Программы, созданные на основе: Code::Blocks, wxMaxima.

Библиотека Qt

- Стандарт отсутствует
- Состав библиотеки: классы, макросы (сигналы и слоты)
- Разработчик: Trolltech (Хаавард Норд, Эрик Чамбенг, 1995 год), Nokia, Digia
- Лицензия: коммерческая и свободная
- Версия: 5.10 (на 6.12.2018)
- Сайт: qt.io
- Область использования: оконные приложения для ОС Linux, Mac OS, Windows на языке C++
- Основа для оболочки KDE (Linux)
- Подключение через заголовочные файлы
- Требуется установка дистрибутива
- Используется метаобъектный компилятор (MOC)
- Qt Creator – инструментальная среда разработки

Классы библиотеки Qt

QObject — базовый класс (механизм сигнал/слот)

QApplication — класс приложения

QWidget – базовый класс элементов управления

QMainWindow – класс главного окна

QDialog – класс диалоговой панели

QLabel – надпись (статический текст)

QPushButton - кнопка

QLineEdit – однострочный текстовый редактор

QString – класс строки

Простейшее приложение на Qt

```
#include <QApplication>
#include <QLabel>

int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    QLabel lb1("Hello, World!");
    lb1.show();
    return app.exec();
}
```

Компиляция проекта в командной строке (Linux, MacOS X)

- Перейти в папку, где находятся файлы проекта.
- Выполнить последовательно команды:
 - `qmake –project`
 - `qmake`
 - `make`

Сигналы и слоты



```
int main(int argc, char* argv[])
{
    QApplication app(argc, argv);
    QPushButton quit("Quit");
    quit.resize(75, 30);
    quit.setFont(QFont("Times", 18, QFont::Bold));

    QObject::connect(&quit, SIGNAL(clicked()), &app,
SLOT(quit()));

    quit.show();

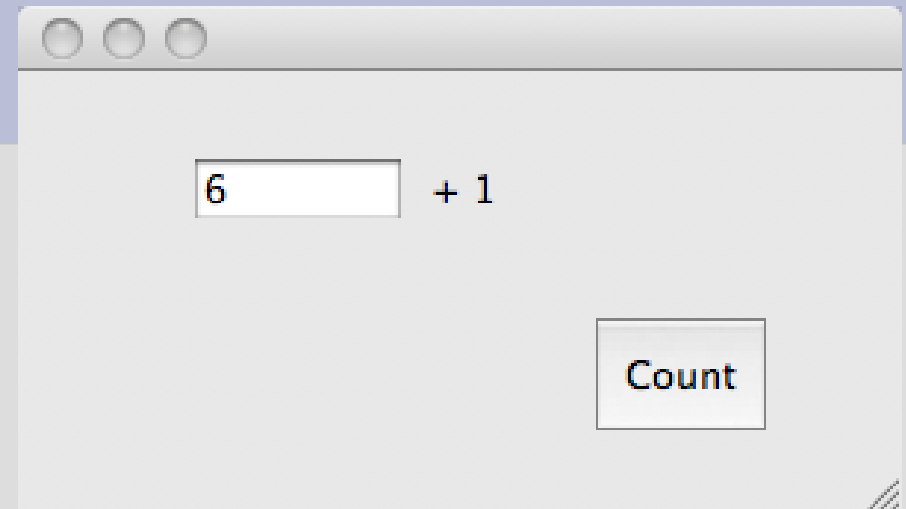
    return app.exec();
}
```

Проект на основе Qt

```
// MyWidget.h
#ifndef MYWIDGET_H
#define MYWIDGET_H

#include <QMainWindow>
#include <QPushButton>
#include <QLineEdit>
#include <QLabel>
#include <QString>

class MyWidget : public QMainWindow
{ Q_OBJECT
public:
    MyWidget();
private:
    QLineEdit* line1;
    QPushButton* bcount;
    QLabel* lcount;
public slots:
    void slotCount();
};
#endif // MYWIDGET_H
```



Разработка в IDE
Code::Blocks

```
#include "..\include\MyWidget.h"
```

MyWidget.cpp

```
MyWidget::MyWidget()
```

```
{
```

```
    this -> resize(300,150);
```

```
    line1 = new QLineEdit(this);
```

```
    bcount = new  
    QPushButton("Count",this);
```

```
    lcount = new QLabel("плюс 1",this);
```

```
    line1 -> move(60,30); line1 ->  
    resize(70,20);
```

```
    lcount -> move(140,30); lcount ->  
    resize(30,20);
```

```
    bcount -> move(190,80); bcount ->  
    resize(50,50);
```

```
    connect (bcount, SIGNAL (clicked()),  
    this, SLOT (slotCount()));
```

```
}
```

```
void MyWidget::slotCount()
```

```
{
```

```
    double d;
```

```
    QString tmp;
```

```
    d=line1->text().toDouble();
```

```
    d=d+1;
```

```
    line1->setText(tmp.setNum(d));
```

```
}
```

```
MyWidget::~~MyWidget()
```

```
{
```

```
    delete line1;
```

```
    delete bcount;
```

```
    delete lcount;
```

```
}
```


main.cpp

```
#include <QApplication>
#include "include\MyWidget.h"

int main(int argc, char* argv[])
{
    QApplication app(argc, argv);
    MyWidget widget;
    widget.show();

    return app.exec();
}
```

Библиотека VCL

- Стандарт отсутствует
- Состав библиотеки VCL (Visual Component Library):
 - Классы, представляющие собой надстройку над объектами Win32 API, - TWindow, TDialog, TButton, TBrush
 - Классы общего назначения TRect, TString
 - Визуальные и не визуальные компоненты (окна, таймеры, меню)
- Разработчик: Borland (Embarcadero Technologies)
- Область использования: оконные приложения для ОС Windows на языках C++, Pascal (Object Pascal)
- Лицензия: коммерческий продукт
- Подключение через заголовочные файлы
- Классы используют простое наследование, базовый класс – TObject (TComponent)

Главный модуль приложения, использующего VCL

```
program Project1;

uses
  Forms,
  Unit1 in 'prob\Unit1.pas' {Form1};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

Библиотека CLX

- Стандарт отсутствует
- Состав библиотеки:
 - ◆ Классы, представляющие собой надстройку над объектами Win32 API, - TWindow, TDialog, TButton, TBrush
 - ◆ Классы общего назначения - TRect
- Разработчик: Borland
- Область использования: оконные приложения для ОС Windows и Unix на языках C++, Pascal (Object Pascal – Delphi, Kylix)
- Подключение через заголовочные файлы
- Классы используют простое наследование, базовый класс – TObject (TComponent)
- Лицензия: коммерческий продукт
- Использует в качестве промежуточного звена библиотеку Qt

Windows Forms

Входит в состав Microsoft .Net Framework.
Содержит средства программирования (API) приложений, в т.ч. графический интерфейс пользователя.

Является надстройкой над Win32 API.
Вытесняет библиотеку MFC.

Предназначена для разработки ПО на языках C#, C++ и др.

Библиотека SDL

- SDL (Simple DirectMedia Layer) – содержит базовый набор средств для создания мультимедийных приложений на языке C:
 - ◆ Установка видеорежима с любой глубиной цвета (от 8 бит и выше)
 - ◆ Прямая запись в линейный графический видеобuffer
 - ◆ Создание поверхностей с атрибутами прозрачности
 - ◆ Воспроизведение 8-ми и 16-ти битных звуков, моно или стерео.
 - ◆ Полный API для управления CD аудио
 - ◆ События для ввода с клавиатуры, ввода мыши, выхода по требованию пользователя
 - ◆ Потoki
 - ◆ Таймеры

Библиотека SDL (продолжение)

- Разработчик: Сэм Лантинга
- Официально поддерживаются операционные системы: Linux, Microsoft Windows, Mac OS, Mac OS X, FreeBSD, OpenBSD, Solaris, IRIX и QNX
- SDL доступна для языков: C, C++, D, Ada, Eiffel, Haskell, Java, Lisp, Lua, Pascal, Perl, PHP, Python и Ruby
- Лицензия: свободная (GNU LGPL)
- Версия: 2.0.4 (3.01.2016)
- Сайт: www.libsdl.org

Библиотека OpenGL

- Стандарт: 1992 год
- Состав:
 - Средства для программирования двух- и трехмерной графики (геометрические и растровые примитивы, текстуры, освещение, атмосферные эффекты)
- Разработчик: Silicon Graphics
- Разработка приложений для ОС Unix, Irix, Linux, Windows, Solaris
- Языки программирования: Java, Perl, Python, Ada, Visual Basic, Pascal, C++ и C#
- Лицензия: различные, в т.ч. Свободная
- Версия: 4.6 (31.06. 2017), поддержка: Chronos Group
- Официальный сайт: www.opengl.org
- Подключение: заголовочные файлы `gl.h`, `glu.h`, `glaux.h`, `glut.h`

DirectX

Состав: набор мультимедийных драйверов высокого уровня, напрямую использующих аппаратные возможности оборудования.

Разработчик: Microsoft.

Версия: DirectX 12 (2015 год).

Возможности: 2D/3D графика, потоковое видео, звук, различные устройства ввода.

Использование: ОС Windows.

Свободно распространяемые с открытым исходным кодом графические библиотеки

OpenCV — библиотека компьютерного зрения.

SFML - мультимедийная библиотека, объектно-ориентированный аналог SDL.

Irrlicht — графический 3D движок.

Lightfeather — 3D движок для разработки игр.

Ogre — объектно-ориентированный графический 3D движок.

Unity — игровой движок со средой разработки.