

# Язык программирования



Лекция № 4

Владимир Владимирович Руцкий  
[rutsky.vladimir@gmail.com](mailto:rutsky.vladimir@gmail.com)



# План занятия

- Использование сторонних библиотек в Python
- Pygame (SDL)
- Практика

# Постановка задачи

- Задача:
  - Программа пишется на Python
  - В программе на Python вызывается код из других библиотек, написанных на других языках
- Есть другая задача (НЕ рассматривается в этой лекции) — встраивание Python-консоли в приложение:
  - Программа написана на C/C++/Java/C#
  - В программе на C/C++/Java/C# вызываются скрипты на Python

# Зачем?

- Многие алгоритмы и драйвера уже реализованы в разных библиотеках
  - Переписать библиотеку на другой язык (Python) — долго, сложно, трудно поддерживать
- Некоторые задачи слишком ресурсотребовательны для решения чисто на Python
  - Обработка массивов данных, изображений
- Разные части проекта могут быть на разных языках, но использовать общие алгоритмы
  - Подготовка данных и вывод 3D графики — на C/C++, тестирование и логика — на Python

# Решение

- Python можно расширить модулями, реализованными внутри динамически подгружаемой библиотеки (DLL)
- Пишется DLL на C/C++ или других языках, позволяющих создать DLL с Си-интерфейсом
- В DLL реализуется Си-интерфейс для Python-модулей
  - таблица экспортируемых Python имён
  - Си-функции, реализующие Python-функции
  - есть интерфейс для передачи объектов из Python в Си и наоборот
- См. <http://docs.python.org/3/extending/extending.html>
- Полученная библиотека на Python — binding (обёртка) для Си-библиотеки

# Генерация обёрток

- Оборачивать каждую функцию библиотеки — рутинно
- Используются генераторы
  - Boost.Python
  - SWIG
  - shiboken
- см. <https://wiki.python.org/moin/IntegratingPythonWithOtherLanguages>

# Boost.Python

- Генератор обёрток на C++

```
#include <boost/python.hpp>

char const * greet()
{
    return "hello, world";
}

BOOST_PYTHON_MODULE(hello_ext)
{
    using namespace boost::python;
    def("greet", greet);
}
```

```
>>> import hello_ext
>>> print hello_ext.greet()
hello, world
```

# SDL

- SDL (Simple DirectMedia Layer)
  - кроссплатформенная библиотека, предоставляющая низкоуровневый доступ к аудио, вводу (клавиатура, мышь, джойстик, мультитач), таймерам, I/O, потокам, и выводу на экран с помощью OpenGL и Direct3D
  - Автор Sam Lantinga (работал в Blizzard, сейчас в Valve)
  - <http://www.libsdl.org/>
- Кроссплатформенная:
  - Windows (Win32 API, Direct3D), Linux (X11, OpenGL), Android (JNI, OpenGL ES), Mac OS X (Cocoa, OpenGL), iOS (UIKit, OpenGL ES)
  - А также приставки и другие мобильные системы
- Написана на Си
  - как и большинство распространённых библиотек
- Распространяется под свободной лицензией
  - zlib license
- Используется разработчиками игр (Valve, Id Software) для кроссплатформенного доступа к периферии



# SDL API

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

int main( int argc, char * args[] )
{
    SDL_Surface * screen = NULL;

    // Инициализация библиотеки SDL
    SDL_Init(SDL_INIT_EVERYTHING);

    // Создаём окно для рисования (или полный экран)
    screen = SDL_SetVideoMode(640, 480, 32, SDL_SWSURFACE);

    // Пауза 4 секунды
    SDL_Delay(4000);

    // Деинициализация SDL
    SDL_Quit();

    return 0;
}
```

# Pygame

- Pygame — Python *биндинг* для библиотеки SDL
- <http://www.pygame.org/>

```
import pygame

def main():
    # Инициализация библиотеки SDL
    pygame.init()

    # Создаём окно для рисования (или полный экран)
    screen = pygame.display.set_mode([640, 480])

    # Пауза 4 секунды
    pygame.time.wait(4000)

    # Деинициализация SDL
    pygame.quit()

if __name__ == "__main__":
    main()
```

# Установка Pygame

- Скачайте последнюю версию для используемой вами версии Python
  - <https://bitbucket.org/pygame/pygame/downloads>
  - pygame-1.9.2a0-hg\_56e0eadfc267.win32-py3.3.msi
- Установите в используемый дистрибутив Python (по умолчанию)
  - C:\Python33
- Документация будет установлена в
  - C:\Python33\Lib\site-packages\pygame\docs\index.html

# Примеры Pygame

- `pygame01_base_template.py`
- `pygame02_simple_graphics_demo.py`
- `pygame03_move_keyboard.py`
- `pygame04_move_joystick.py`
- `pygame05_move_mouse.py`
- `pygame06_bitmapped_graphics.py`

# Практика