

Обзор компонентов (приложений, библиотек) языка приложения Python для создания конечного приложения

Кизянов Антон Олегович

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Лучанинов Дмитрий Васильевич

*Приамурский государственный университет имени Шолом-Алейхема
старший преподаватель кафедры информационных систем, математики и
методик преподавания*

Аннотация

В данной статье проведено сравнение компонентов для компиляции конечного приложения на языке программирования Python.

Ключевые слова: Python, Pyinstaller, py2exe, nuitka, cx_Freeze

Analysis of the software implementing the Python programming language

Kizyanov Anton Olegovich,

*Sholom-Aleichem Priamursky State University
student*

Luchaninov Dmitry Vasilyevich

*Sholom-Aleichem Priamursky State University
Senior lecturer of the Department of Information Systems, Mathematics and
teaching methods*

Abstract

In this article components to compile the final application in the Python programming language are compared.

Keywords: Python, Pyinstaller, py2exe, nuitka, cx_Freeze

Эта статья поможет разобраться с методами скрытия исходных кодов в языке программирования Python.

Целью данной статьи является описание и оценка основных компиляторов языка программирования Python.

Для решения этой проблемы есть несколько решений: Pyinstaller, py2exe, nuitka, cx_Freeze.

Изучением работы с языком программирования Python занимались С.А.Кудрявцев показал на его примере тестирование в python, объектно-ориентированным и процедурным подходом[6]. В.А.Машков, В.И.Литвиненко рассказали о применении языка программирования python

для решения задач самодиагностики на системном уровне [7]. Э.А.Усеинов показал объектно-ориентированное программирование в рамках дисциплины «язык программирования python» [8]. Г.С.Сейдаметов продемонстрировал особенности использования языка программирования python в подготовке будущих инженеров-программистов [9].

Самый популярный компилятор для Python это Pyinstaller. Дальнейшая установка предусматривает что на компьютер установлен интерпретатор Python. Устанавливается командой в консоли **pip install Pyinstaller**. После нужно перейти в папку где установлен Python, зайти в папку **\Scripts**, там есть файл **pyinstaller.exe**. Перетаскиваем скрипт написанный на языке Python на этот файл и через секунды в рабочем каталоге будет папка со всеми библиотеками и исполняемым файлом. Главное преимущество в его универсальности, «ест» все библиотеки и запускается везде где есть Windows. Для полноценной работы нужно носить все файлы которые появились вместе с исполняемым файлом с собой.

Есть ещё py2exe, второй по популярности компилятор для Python. Устанавливается следующей командой в консоли **pip install py2exe**. Для компиляции нужно создать файл такого содержания

```
from distutils.core import setup
import py2exe

setup(console=['hello.py'])
```

Самое главное это файл hello.py (это имя скрипта для перевода в .exe). Остается только сохранить и запустить. В папке с файлом появится скомпилированный файл hello.exe, для полноценной работы нужно все созданные файлы таскать с исполняемым hello.exe.

Следующий способ очень похож на py2exe, называется он cx_Freeze.

Для компиляции нужен файл настроек. Вот пример содержимого файла настроек.

```
import sys
from cx_Freeze import setup, Executable
build_exe_options = {"packages": ["os"], "excludes": ["tkinter"]}
base = None
if sys.platform == "win32":
    base = "Win32GUI"

setup( name = "guifoo",
       version = "0.1",
       description = "My GUI application!",
       options = {"build_exe": build_exe_options},
       executables = [Executable("guifoo.py", base=base)])
```

Здесь скрипт выступает как `guifoo.py`, его и предполагается перевести в `exe`. После запуска в текущем каталоге появляется файл `.exe` и его библиотеки. Для транспортировки нужно нести все библиотеки с собой. Существенный недостаток, что нужно в переменной «`build_exe_options`» указывать все используемые библиотеки.

Следующая `nuitka`, задумка этого способа в переводе скриптов на языке программирования `python` на язык программирования `C++` с последующей компиляцией. Разработчики обещают прирост производительности на 258 процентов и меньший размер готового `.exe`. Для установки нужно набрать в консоли **`pip install nuitka`** все. Чтобы компилировать файл нужно в консоли указать **`nuitka <путь до файла>`**, компилирует долго (около 15 минут) положит файл в рабочую директорию, для работоспособности не нужны никакие библиотеки, переносим только один `.exe`.

Первые 3 способа отличаются от последнего тем, что они копируют главное ядро языка `Python` в `exe` и ваши листинги, поэтому когда запускаем его на сторонней машине, мы носим с собой вшитый в `exe` интерпретатор `Python`, а `nuitka` компилирует все в байт код и запаковывает в `exe`.

Представим сравнение компонент (табл.1).

Таблица 1. Сравнение компонентов

Критерии	Pyinstaller	Py2exe	Cx_Freeze	nuitka
Вес готового <code>exe</code>	~16 мб	~20мб	~19мб	~12 мб
Сам ищет библиотеки	Да	Нет	Нет	Да
Переносимость <code>exe</code> на другие машины	Да	Да	Да	Да

Библиографический список

1. Pyinstaller. [Электронный ресурс]. URL: <http://www.pyinstaller.org/> (дата обращения: 18.08.2016)
2. Py2Exe. [Электронный ресурс]. URL: <http://www.py2exe.org/> (дата обращения: 18.08.2016)
3. Cx_Freeze. [Электронный ресурс]. URL: <http://cx-freeze.sourceforge.net/> (дата обращения: 18.08.2016)
4. Nuitka. [Электронный ресурс]. URL: <http://nuitka.net/> (дата обращения: 16.07.2016)
5. Python. [Электронный ресурс]. URL: <https://www.python.org/> (дата обращения: 18.08.2016)
6. Кудрявцев С.А. Тестирование в `python` - объектно-ориентированный и процедурный подход // В сборнике: RSDN MAGAZINE Издательство: К-Пресс (Москва) С. 67-70

7. Машков В.А., Литвиненко В.И. Использование языка программирования python 3 и системы компьютерной алгебры sympy факультативных занятиях по теории чисел // В сборнике: Электротехнические и компьютерные системы Издательство: Одесский национальный политехнический университет (Одесса) С. 48-54
8. Усеинов Э.А. Объектно-ориентированное программирование в рамках дисциплины «язык программирования python» // В сборнике: Ученые записки крымского инженерно-педагогического университета Издательство: Государственное бюджетное образовательное учреждение высшего образования Республики Крым "Крымский инженерно-педагогический университет" (Симферополь) С. 157-160 [Электронный ресурс].
9. Сейдаметов Г. С. Особенности использования языка программирования python в подготовке будущих инженеров-программистов // В сборнике: INTERNATIONAL SCIENTIFIC REVIEW Издательство: Олимп (Иваново) С. 50-51