

**Дисциплина: «Основы алгоритмизации и
программирования»**

Лекция 1: Введение в язык программирования PYTHON

**Преподаватель специальных дисциплин:
Кузембаева А.Ж.**

План лекции:

Введение;

1. История;
 2. Особенности языка;
 3. Отличия Python 2 от Python 3;
 4. Структура программы на языке Python;
 5. Преимущества и недостатки;
- Заключение.

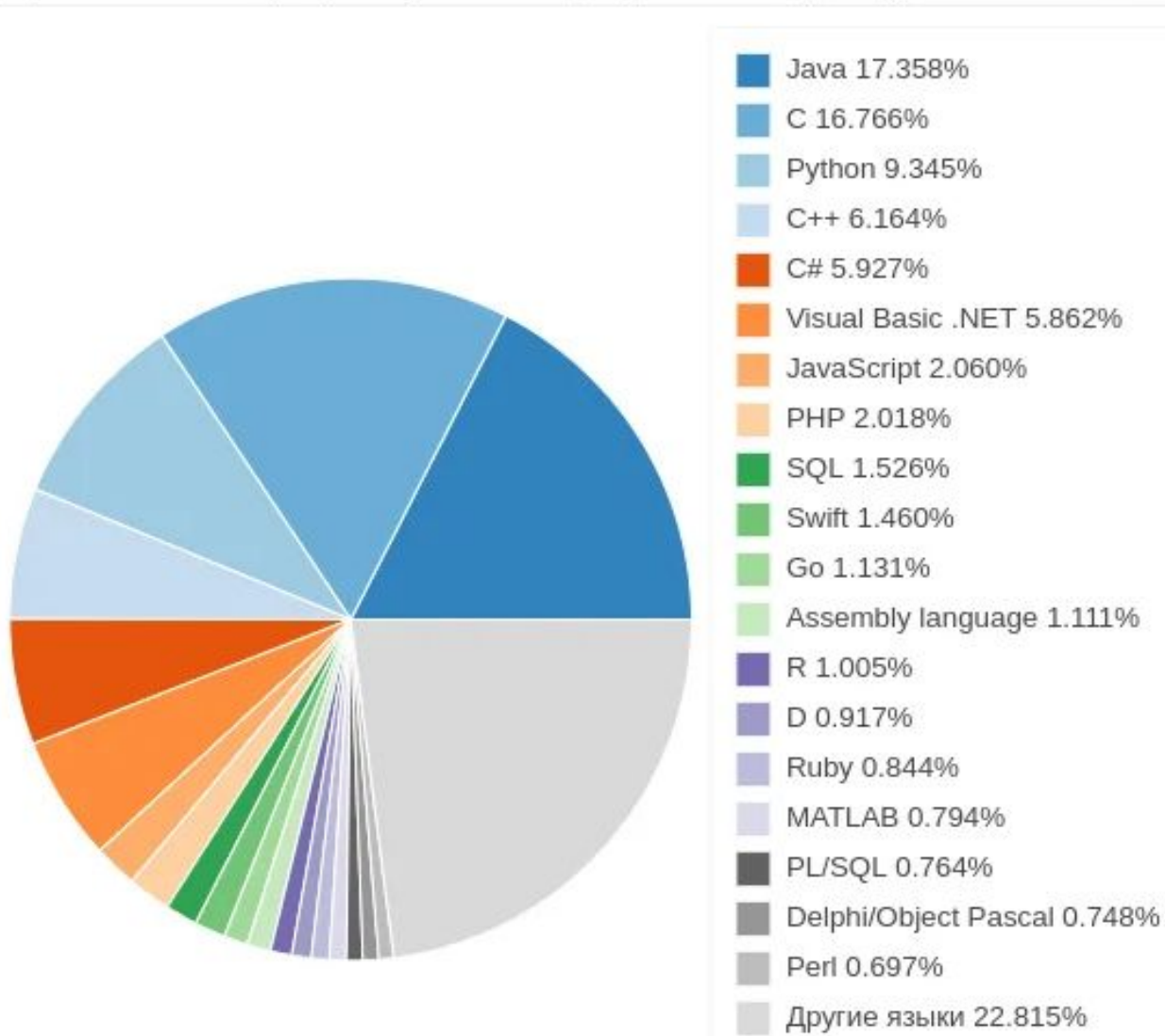
История:



- Язык программирования Python был создан примерно в 1991 году голландцем Гвидо ван Россумом.
- Свое имя – Пайтон (или Питон) – получил от названия телесериала, а не пресмыкающегося.
- После того, как Россум разработал язык, он выложил его в Интернет, где уже целое сообщество программистов присоединилось к его улучшению. Python активно совершенствуется и в настоящее время. Часто выходят его новые версии. Официальный сайт <http://python.org>.

Рейтинг языков программирования:

Популярность языков программирования в феврале 2020 года по данным TIOBE



Особенности языка:



1. Простой
2. Лёгкий в освоении
3. Свободный и открытый
4. Язык высокого уровня
5. Портируемый
6. Интерпретируемый
7. Объектно-ориентированный
8. Расширяемый
9. Встраиваемый
10. Обширные библиотеки

Особенности Python:

- **Python** — высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода
- Python и подавляющее большинство библиотек к нему бесплатны и поставляются в исходных кодах. Более того, в отличие от многих открытых систем, лицензия никак не ограничивает использование Python в коммерческих разработках

Термины:

- **Интерпретируемый язык программирования** — язык программирования, в котором исходный код программы не преобразуется в машинный код для непосредственного выполнения центральным процессором, а исполняется с помощью специальной программы-интерпретатора. Т.е. код программы выполняется «на лету», переводится в машинный язык строка за строкой во время выполнения программы.

Термины:

- **Высокоуровневый язык программирования** — язык программирования, разработанный для скорости и удобства использования программистом.
- **Основная черта высокоуровневых языков** — это абстракция, то есть введение смысловых конструкций, кратко описывающих структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания.

Термины:

- **Модульность** — принцип, согласно которому программа разделяется на отдельные именованные сущности, называемые модулями. Модульность часто является средством упрощения задачи проектирования программы и распределения процесса разработки между группами разработчиков. При разбиении программы на модули для каждого из них указывается реализуемая им функциональность, а также связи с другими модулями

Термины:

- **Модульность** — принцип, согласно которому программа разделяется на отдельные именованные сущности, называемые модулями. Модульность часто является средством упрощения задачи проектирования программы и распределения процесса разработки между группами разработчиков. При разбиении программы на модули для каждого из них указывается реализуемая им функциональность, а также связи с другими модулями

Термины:

- **Синтаксис** — сторона языка программирования, которая описывает структуру программ как набор символов (обычно говорят — безотносительно к содержанию). Синтаксису языка противопоставляется его семантика. Синтаксис языка описывает «чистый» язык, в то время как семантика приписывает значения (действия) различным синтаксическим конструкциям.
- **Семантика** — система правил определения поведения отдельных языковых конструкций. Семантика определяет смысловое значение предложений алгоритмического языка.

Особенности Python:



Если язык компилируемый (C++):

- Программа преобразуется из исходного языка (C++) в язык, понятный компьютеру (бинарный код, 0 и 1) при помощи компилятора. Когда вы запускаете такую программу, компоновщик/загрузчик копирует программу с диска в оперативную память и запускает её.

Особенности Python:

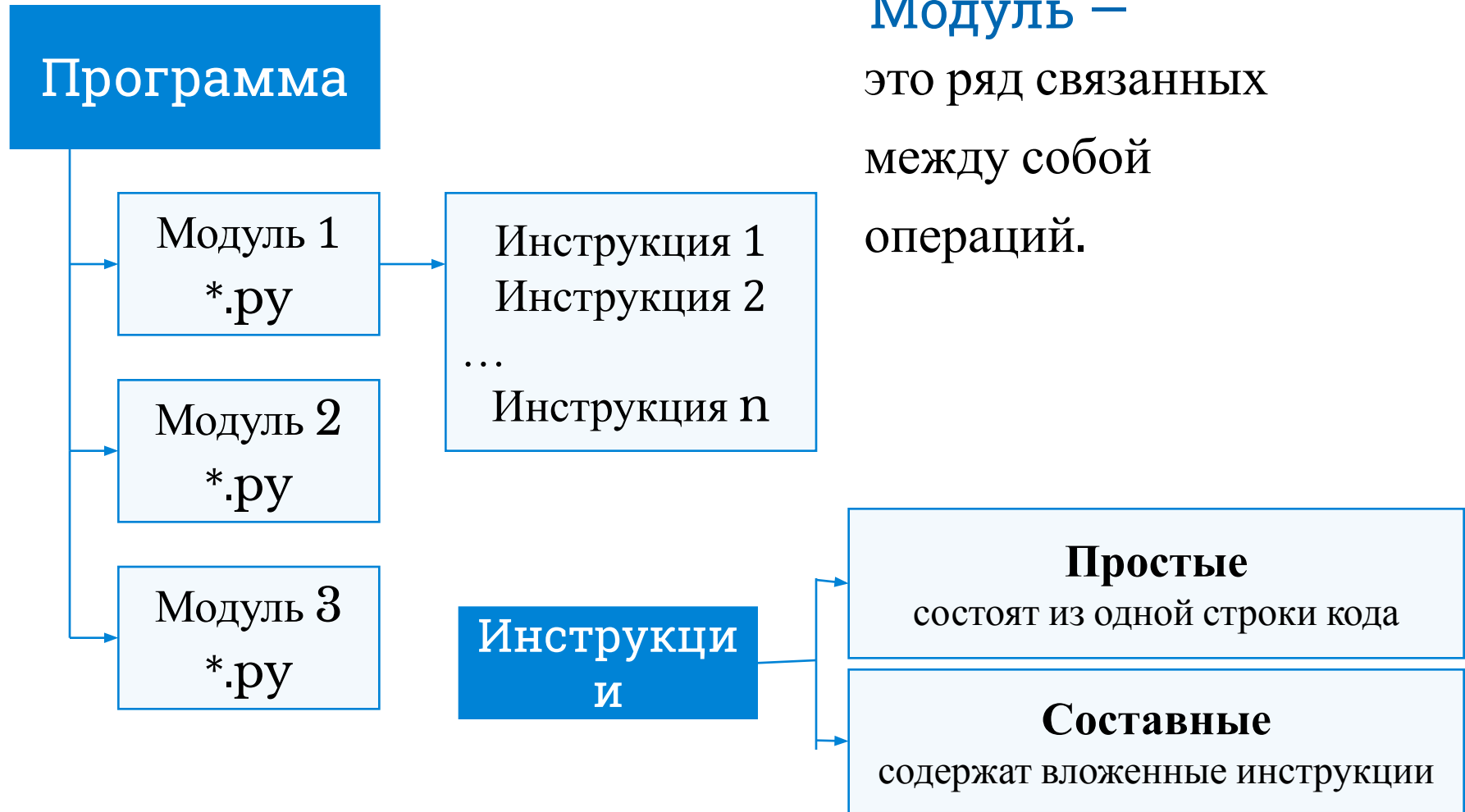
Python – интерпретируемый язык программирования

- Не требует компиляции в бинарный код. Программа выполняется из исходного текста. Он сам преобразует текст в некоторую промежуточную форму, называемую байткодом, а затем переводит его на машинный язык и запускает.
- Программы на Python легко переносимы, их достаточно просто скопировать на другой компьютер, и они работают.

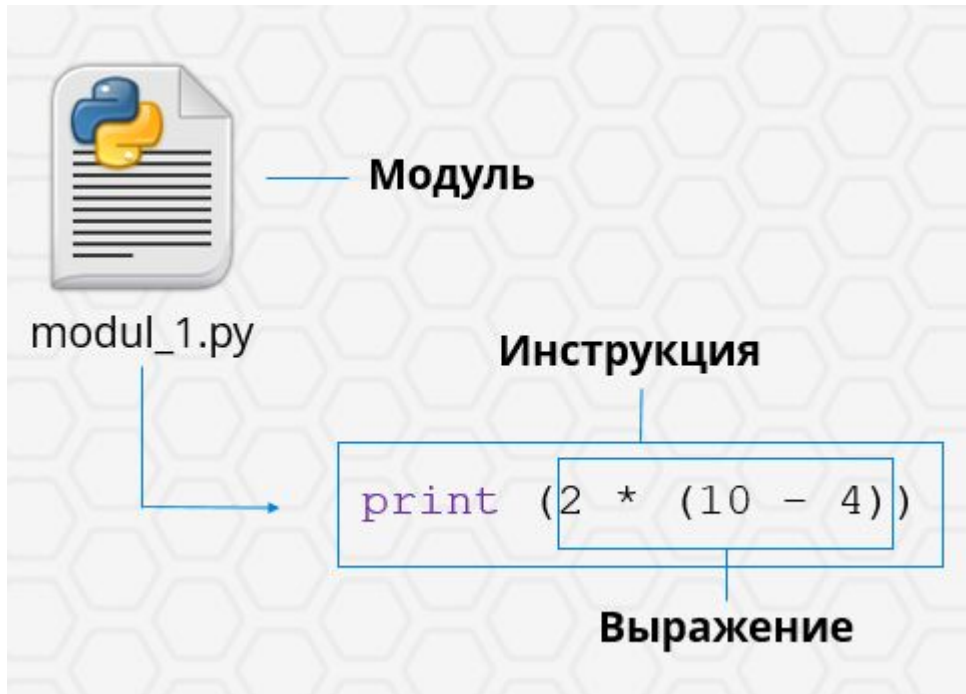
Особенности Python:

- Python поддерживает как процедурно-ориентированное, так и объектно-ориентированное программирование.
- В *процедурно - ориентированных* языках программы строятся на основе процедур или функций, которые представляют собой многократно используемые фрагменты программы.
- В *объектно - ориентированных* языках программирования программы строятся на основе объектов, объединяющих в себе данные и функционал. Python предоставляет простые, но мощные средства для ООП, особенно в сравнении с такими большими языками программирования, как C++ или Java.

Структура программы на языке Python:



Структура программы на языке Python:



Инструкции —

это указания компьютеру, определяющие, какие операции выполнит компьютер над данными.

Выражения в составе инструкций определяют, над какими именно данными будут выполнены действия, описанные в инструкции.

Структура программы на языке Python:

Операции - это любые действия над операндами.

Операнды -

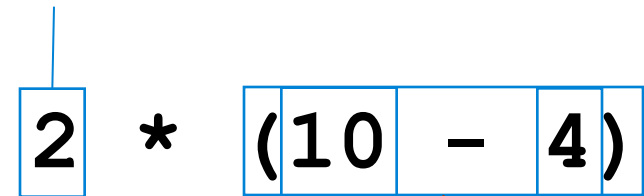
это некоторые данные.

- ✓ литералы;
- ✓ выражения;
- ✓ переменные.

Приоритет выполнения

операций - соответствует принятому в математике.

Литерал



Литералы

Выражение

Структура программы на языке Python:

Переменная —

это именованная область оперативной памяти, в которой хранятся некоторые данные определённого типа.

Переменная:

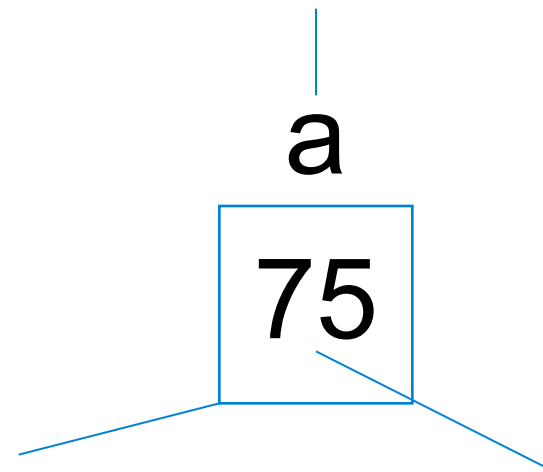
Имя переменной

а

75

Ячейка оперативной
памяти

Значение
переменной



Отличия Python 2 от Python 3

- Python 2.0 была выпущена в 2000 году. Его последняя версия, 2.7, была выпущена в 2010 году.
- Python 3.0 была выпущена в 2008 году. Его новейшая версия 3.6 была выпущена в 2016 году, а версия 3.8 была доступна уже в декабре 2019 года.

Отличия Python 2 от Python 3

- Python 2 - это стандарт, а Python 3 - это будущее
- Python 2 был самой популярной версией более 15 лет, она все еще является актуальной среди некоторых разработчиков приложений. Но, сейчас уже все больше компаний переходят к использованию третьей версии.
- Python 2 и Python 3 имеют разные (иногда несовместимые) библиотеки. Сегодня многие разработчики создают библиотеки исключительно для использования с третьей частью. Точно так же многие старые библиотеки, созданные для Python 2, не совместимы с последующей версией. Возможно, у вас получится портировать библиотеку 2.x на 3.x., но это может быть сложно и потребовать много времени – это определенно не уровень новичков.

Отличия Python 2 от Python 3

- Лучшая поддержка Unicode в Python 3.

В Python 3 текстовые строки по умолчанию являются Unicode. В Python 2 строки хранятся как ASCII по умолчанию – вам нужно добавить "u" если вы захотите сохранить строки в формате Unicode в Python 2.x. Unicode более универсален, чем ASCII. Строки Unicode могут хранить буквы на иностранных языках, латинские буквы и цифры, символы, смайлики и т. д., предлагая пользователю более широкий выбор.

- Версии имеют разный синтаксис

Синтаксис

- Не содержит операторных скобок (`begin..end` в `pascal` или `{..}` в `Си`), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием.
- Однострочные комментарии начинаются со знака фунта «`#`», многострочные — начинаются и заканчиваются тремя двойными кавычками «`"""`».

Синтаксис

- Чтобы присвоить значение переменной используется знак «=», а для сравнения — «==». Для увеличения значения переменной, или добавления к строке используется оператор «+=», а для уменьшения — «-=». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками.

Синтаксис

- На PYTHON

```
print("Hello, World!")
```

- На C++

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

```
void main()
```

```
{
```

```
    cout << "Hello, World!" << endl;
```

```
}
```


Синтаксис

- На PYTHON

```
print("Hello, World!")
```

- На C++

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

```
void main()
```

```
{
```

```
    cout << "Hello, World!" << endl;
```

```
}
```

Синтаксис

Pascal -

```
a := 10;
```

```
b := 1;
```

```
WHILE (a > 0) do
```

```
    Begin
```

```
        b = b * 2;
```

```
        a = a - 1;
```

```
    End;
```

Python-

```
a = 10
```

```
b = 1
```

```
while a > 0:
```

```
    b = b * 2
```

```
    a = a - 1
```

Преимущества Python

- Скорость выполнения программ написанных на Python очень высока. Это связано с тем, что основные библиотеки Python написаны на C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.
- В стандартных библиотеках Python вы можете найти средства для работы с электронной почтой, протоколами Интернета, FTP, HTTP, базами данных, и пр.

Преимущества Python

- Скрипты, написанные при помощи Python выполняются на большинстве современных ОС. Такая переносимость обеспечивает Python применение в самых различных областях.
- Python подходит для любых решений в области программирования, будь то офисные программы, веб-приложения, GUI-приложения и т.д.

Преимущества Python

Интерактивный режим

- В основном интерпретатор выполняет команды построчно: пишешь строку, нажимаешь Enter, интерпретатор выполняет ее, наблюдаешь результат.
- Возможности языка позволяют использовать его как калькулятор, не зная команд программирования.

$2 + 5$

$3 * (5 - 8)$

$2.4 + 3.0 / 2$

Недостатки Python

- Python, как и другие интерпретируемые языки, имеет сравнительно невысокую скорость выполнения программ. Однако, в случае с Python этот недостаток компенсируется уменьшением времени разработки программы. В среднем, программа на Python в 2-4 раза компактнее, чем её аналог на C++ или Java

Заключение

Задачи, решаемые с помощью Python :

- 1. Системное программирование.
- Встроенные в Python интерфейсы доступа к службам операционных систем делают его идеальным инструментом для создания переносимых программ и утилит системного администрирования (иногда они называются инструментами командной оболочки).

Заключение

Python продолжает оставаться лидирующим инструментом в :

□ 2. Графический интерфейс.

- Простота Python и высокая скорость разработки делают его отличным средством разработки графического интерфейса. В состав Python входит стандартный объектно-ориентированный интерфейс к Tk GUI API, который называется tkinter.

Заключение

Python продолжает оставаться лидирующим инструментом в :

□ 2. Графический интерфейс.

- Простота Python и высокая скорость разработки делают его отличным средством разработки графического интерфейса. В состав Python входит стандартный объектно-ориентированный интерфейс к Tk GUI API, который называется tkinter.

Заключение

Python продолжает оставаться лидирующим инструментом в :

□ 3. Веб-сценарии.

Интерпретатор Python поставляется вместе со стандартными интернет-модулями, которые позволяют программам выполнять разнообразные сетевые операции как в режиме клиента, так и в режиме сервера. Сценарии могут производить взаимодействия через сокет, извлекать информацию из форм, отправленных серверным CGI-сценариям; передавать файлы по протоколу FTP; обрабатывать файлы XML; передавать, принимать, создавать и производить разбор писем электронной почты; загружать веб-страницы с указанных адресов URL и многое другое.

Заключение

Python продолжает оставаться лидирующим инструментом в :

□ 4. Интеграция компонентов.

Возможность Python расширяться и встраиваться в системы на языке и C++ делает его удобным и гибким языком для описания поведения других систем и компонентов. Например, интеграция с библиотекой на языке C позволяет Python проверять наличие библиотечных компонентов и запускать их, а встраивание Python в программные продукты позволяет производить настройку программных продуктов без необходимости пересобирать эти продукты.

Заключение

Python продолжает оставаться лидирующим инструментом в :

□5. Приложения баз данных.

В языке Python имеются интерфейсы доступа ко всем основным реляционным базам данных: Sybase, Oracle, Informix, ODBC, MySQL, PostgreSQL, SQLite и многим другим.

□6. Быстрое создание прототипов.

В программах на языке Python компоненты, написанные на Python и на C, выглядят одинаково. Благодаря этому можно сначала создавать прототипы систем на языке Python, а затем переносить выбранные компоненты на компилируемые языки, что существенно экономит время разработ