

ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦИЯ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

ЛЕКЦИЯ №1

# Типы и операции языка программирования Python.

СОСТАВИТЕЛЬ:            КАНД. ТЕХН. НАУК    БЫКАДОР В.С.

# Общая характеристика языка программирования Python

- 1) Поддерживает основные парадигмы программирования.
- 2) Высокая степень кроссплатформенности.
- 3) Независимость от вендора и бесплатность.
- 4) Динамическая типизация.
- 5) Автоматическое управление памятью.
- 6) Модульная организация программ.
- 7) Встроенные типы объектов.
- 8) Широкая собственная библиотека классов.
- 9) Ещё более широкая библиотека классов и фреймворков сторонних разработчиков.

# Преимущества языка Python

- 1) Высокая скорость разработки (синтаксис динамическая типизация, много готовых библиотек).
- 2) Переносимость программного кода (виртуальная машина).
- 3) Большое количество библиотек.
- 4) Интеграция компонентов (вызов кода на C, последовательный порт).
- 5) Автоматическое управление памятью.

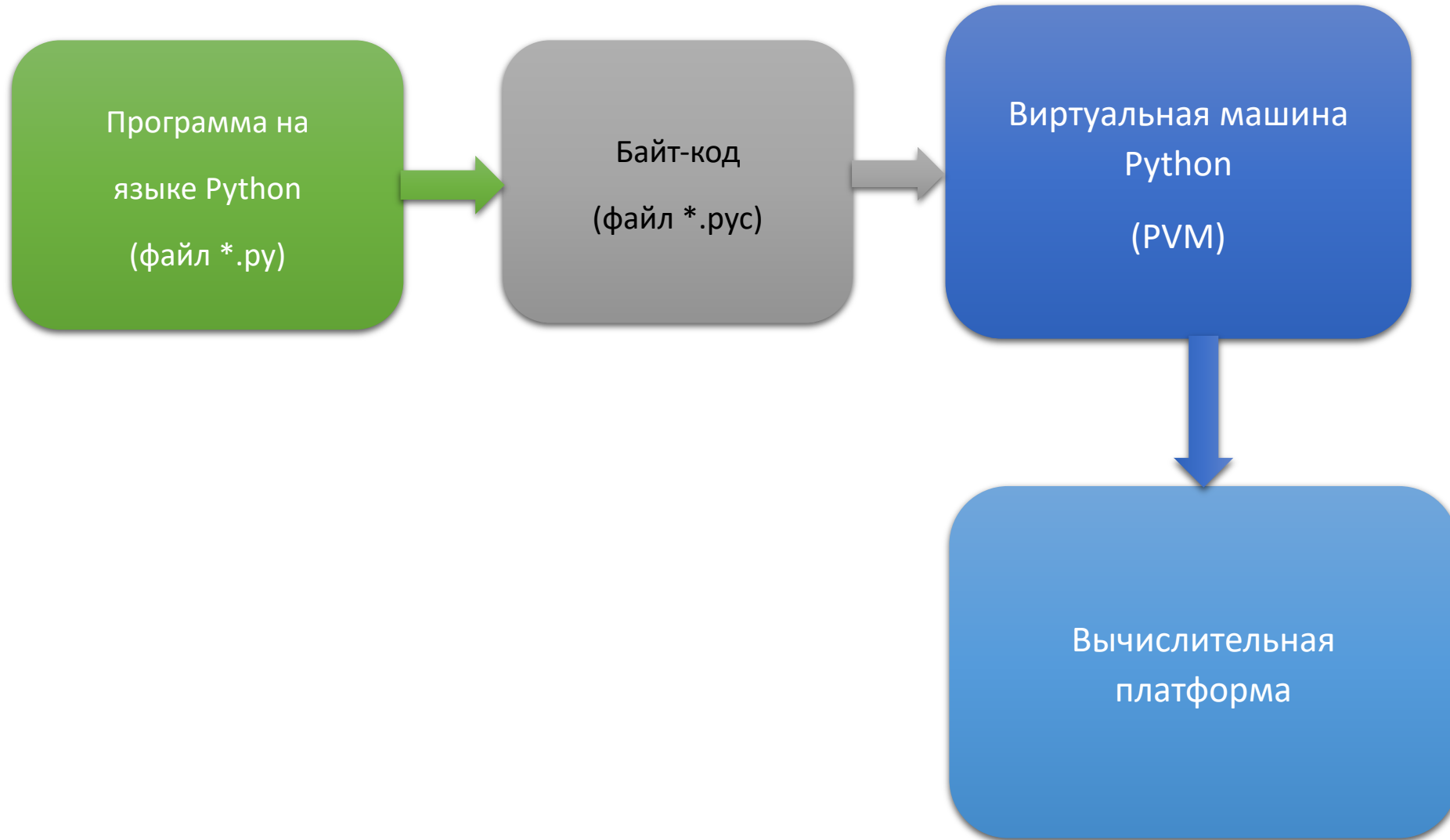
# Недостатки языка Python

- 1) Низкая скорость выполнения кода.
- 2) Потребление большого количества памяти.
- 3) Динамическая типизация (для управления проектов).
- 4) Можно программировать различные типы проектов, но все.

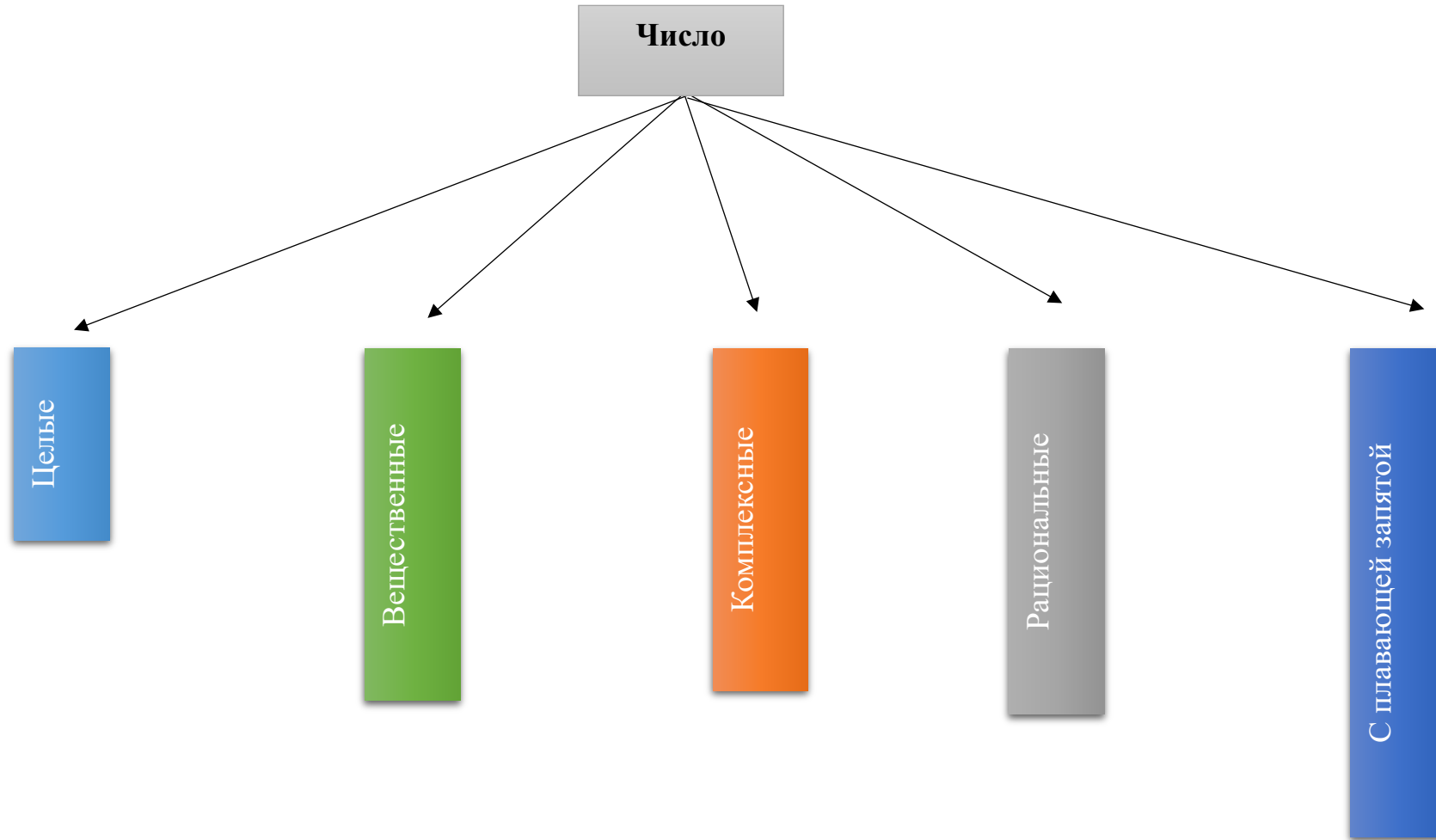
# Типы проектов, в которых используется язык Python

- 1) Научные и инженерные вычисления (математическое моделирование, расчёты).
- 2) Общий анализ данных.
- 3) Машинное обучение.
- 4) Нейросети.
- 5) Веб-разработка (backend-часть).
- 6) Скрайпинг (скрейпинг).
- 7) Веб-роботы (чат-боты).
- 8) Автоматизация (тестирование, администрирование, любое другое).
- 9) Десктопные приложения с графическим интерфейсом.
- 10) Мобильная разработка.
- 11) Микроконтроллеры.

# Понятие об интерпретаторе Python



# Основные типы объектов Python



Целые,  
вещественные  
и  
комплексные  
числа

```
Python 3.10.4  
AMD64) ] on win  
Type "help", "  
>>>  
>>> 123  
123  
>>>  
>>> 3.14  
3.14  
>>>  
>>>  
>>> 4+2j|  
(4+2j)
```



## Рациональные числа

```
1  from fractions import Fraction
2
3
4  f = Fraction(2, 5)
5  print(f)
6
7
8
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```
PS C:\Users\bykvi\YandexDisk\В работе\Лекции по Python3.10.exe "c:/Users/bykvi/YandexDisk/В работе/Лекции по Python3.10.exe"
2/5
```

```
PS C:\Users\bykvi\YandexDisk\В работе\Лекции по Python3.10.exe
```

# Работа с вещественными и рациональными числами

```
1  from fractions import Fraction
2  from decimal import Decimal
3
4  d = Decimal('1.1')
5  f = Fraction(d)
6  print(f)
7
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

PS C:\Users\bykvi\YandexDisk\В работе\Лекции по Рут  
n3.10.exe "c:/Users/bykvi/YandexDisk/В работе/Лекции

11/10

PS C:\Users\bykvi\YandexDisk\В работе\Лекции по Рут

# Тип объекта – Строка

- 1) Строки представляют собой последовательности, то есть упорядоченные коллекции других объектов – символов.
- 2) Индексирование символов в строке начинается с нуля.
- 3) Индексация обычно выполняется положительными индексами, но можно и отрицательными.
- 4) Так как строка - последовательность символов, то можно вычислить количество составляющих её символов, так называемую «длину» строки.
- 5) Можно выполнять «срезы» над строками.
- 6) Имеется ряд специфичных для типа методов, например,
  - 1) **find**,
  - 2) **replace**,
  - 3) **split**,
  - 4) **isalpha**,
  - 5) **isdigit**,
  - 6) **upper**,
  - 7) **format** и ряд других.

**Важно! Строка является неизменяемым объектом в памяти.**

# Тип объекта – Списки ( list() )

1) Списки являются обобщенным видом упорядоченных последовательностей, которые могут содержать любое количество любых других объектов при этом в пределах одного списка.

2) Так же список может содержать другие списки:

```
>>> lst = [1, 3+4j, 'spam', [1, 2, 'other']]
>>> lst
[1, (3+4j), 'spam', [1, 2, 'other']]
>>> |
```

3) Так же, как и у строк, у списков есть ряд таких же методов, например, индексация, срезы, длина списка.

4) Но для списков есть и свои специфичные методы, часто используемыми из них являются **append**, **pop**, **sort**.

**Важно!** При присвоении одного списка другому происходит копирование ссылки на исходный список.

# Тип объекта – Словари ( dict() )

Словари представляют собой отображения, то есть коллекции обращения к элементам которой происходит по ключу. Словари являются классом изменяемых объектов.

На рисунке ниже показаны типовые действия над словарём:

- 1) Объявление и инициализация переменной типа «словарь».
- 2) Получение значения из словаря по ключу 'lastname'.
- 3) Метод keys() возвращает все ключи словаря.
- 4) Метод values() возвращает все значения словаря.
- 5) Далее выполняется добавление новой пары «ключ – значение» = 'university' – 'dstu', которой ранее не было в словаре. И получение значения по только что добавленному ключу 'university'.

# Тип объекта – Словари ( dict() )

```
>>>
>>> d = {'name': 'Лиза', 'lastname': 'Иванова', 'age': 26}
>>> d
{'name': 'Лиза', 'lastname': 'Иванова', 'age': 26}
>>>
>>> d['lastname']
'Иванова'
>>>
>>> d.keys()
dict_keys(['name', 'lastname', 'age'])
>>> d.values()
dict_values(['Лиза', 'Иванова', 26])
>>>
>>> d['university'] = 'dstu'
>>>
>>> d
{'name': 'Лиза', 'lastname': 'Иванова', 'age': 26, 'university': 'dstu'}
>>>
>>> d['university']
'dstu'
>>>
```