

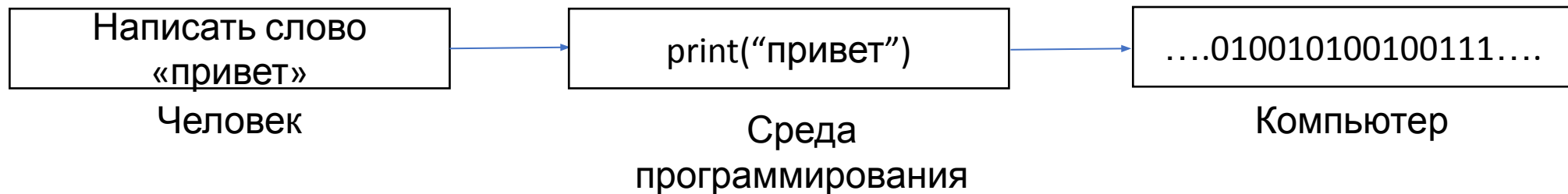
python

Занятие 0. Что такое программирование?

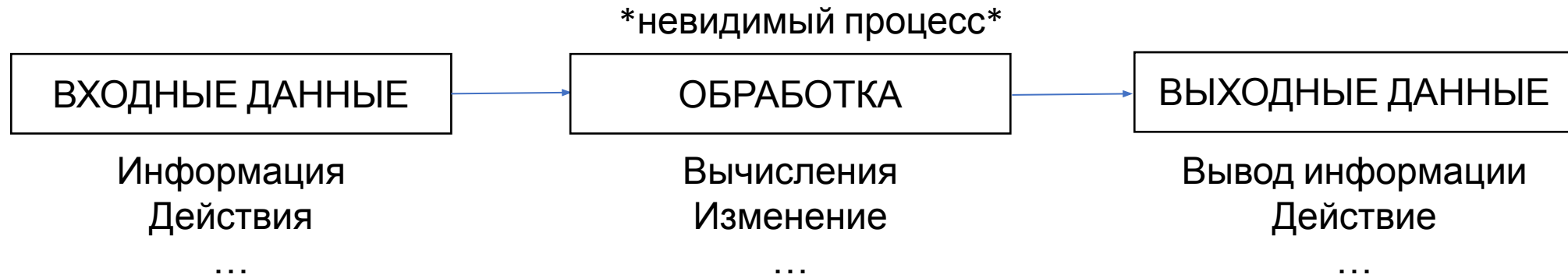
Таран Данила Сергеевич

Программирование

- Программирование – процесс создания программ
- Язык программирования – формальный язык, предназначенный для записи программ



Программирование | алгоритм



Пример:

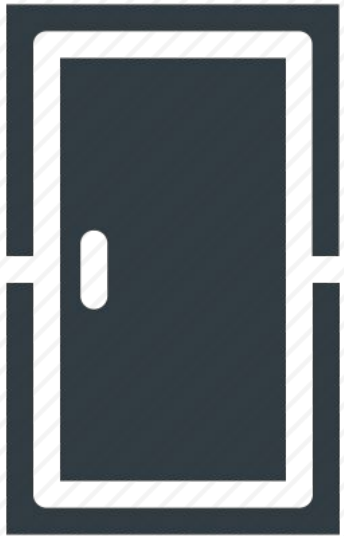
```
=====
Введите два числа:
5
10
Сумма равна: 15
```



Программирование | алгоритм

«Мысли как компьютер!»

Преврати простое действие в четкую последовательность шагов



1. Подойти к двери
2. Толкнуть дверь
3. Если дверь открылась, **войти**
4. Если не открылась, **стучать** пока не откроют, **войти**



Программирование | алгоритм

- Если язык программирования не умеет выполнять некоторые действия сам, придется научить:

Подойти:

1. Пока впереди не препятствие, **делать шаг**

Делать шаг:

1. ...
2. ...
- ...



Программирование | данные

- Переменная – область памяти в оперативной памяти компьютера, к которой можно удобно обращаться по имени, «читать» и изменять информацию

«Не пытайся засунуть холодильник в кошелек!»

- Переменные языка Python (как и многих других) хранят информацию **определенного** типа. В переменную, определенную для хранения числа не получится записать **ТЕКСТ**. *(если, конечно, не преобразовать ее в другой тип, но об этом позднее)*



Типы данных в Python

В Python имеется множество встроенных типов данных. Наиболее важные из них:

- ▶ **Логический**
- ▶ **Числа:** целые, с плавающей точкой, дробные и комплексные
- ▶ **Строки** — последовательности символов Юникода
- ▶ **Байты и массивы байтов**
- ▶ **Списки** — упорядоченные последовательности значений
- ▶ **Кортежи** — упорядоченные неизменяемые последовательности значений
- ▶ **Множества** — неупорядоченные наборы значений
- ▶ **Словари** — неупорядоченные наборы пар вида ключ-значение



Программирование | компиляция и интерпретация

- **Компиляция** – перевод всего текста программы, написанной на языке программирования (*как правило, высокоуровневом*) в машинный код (*или на язык более низкого уровня*) и последующая сборка приложения, которое нужно потом запустить. *Напр, C++, C#*
- **Интерпретация** – построчный анализ, обработка и выполнение кода программы без предварительной сборки. *Напр, Python, JS*
- **Компилятор** – программа, выполняющая компиляцию
- **Интерпретатор** – программа, выполняющая интерпретацию...



Программирование | компиляция и интерпретация

Что насчет Python?

Тут не все так просто. Принято считать, что Python – **интерпретируемый язык программирования**.

На самом деле код программы, написанной на Python сначала компилируется в байт-код, который далее интерпретируется.

А еще Pythonов много.

Так что вопрос неоднозначный, все зависит от разных факторов.

Но мы будем говорить, что Python интерпретируемый.



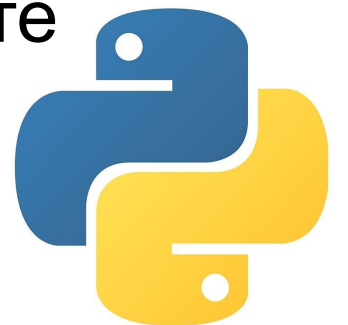
Программирование | компиляция и интерпретация

Так что же все-таки лучше?

И на этот вопрос ответить нельзя точно. Интерпретируемый код чуть медленнее (*почему?*), но менее чувствителен к платформе.

А еще интерпретируемый код легче найти и прочитать, тогда как скомпилированный может быть не доступен.

Выбор должен зависеть от программы, которую вы хотите написать.



Hello, World!

```
>>> print ('Hello, World!')  
'Hello, World!'
```



Калькулятор

```
• print("Ноль в качестве знака операции"
"\nзавершит работу программы")
while True:
s = input("Знак (+,-,*,/): ")
if s == '0':
break
if s in ('+', '-', '*', '/'):
x = float(input("x="))
y = float(input("y="))
if s == '+':
print("%.2f" % (x+y))
elif s == '-':
print("%.2f" % (x-y))
elif s == '*':
print("%.2f" % (x*y))
elif s == '/':
if y != 0:
print("%.2f" % (x/y))
else:
print("Деление на ноль!")
else:
print("Неверный знак операции!")
```



Для чего используется Python

Python может использоваться для многих целей, например:

- ▶ Разработка прикладного ПО
- ▶ Разработка мобильных приложений
- ▶ Разработка web-приложений
- ▶ В качестве встраиваемого скриптового языка во многих играх, и программах
- ▶ В научных расчетах



Python



Dropbox

