Основы программирования (КИСиП)

Контрольная работа 21/22 года обучения

К контрольной работе приступать после изучения: **лекции** , **литература** и **Глоссарий (основные определения)**.

Проходной балл 60% (максимально 40% за тестовые вопросы и 60% за программу). Программу проверяют в течении 3 дней.

Функции

Функции

Функции — это такие участки кода, которые изолированы от остальной программы и выполняются только тогда, когда вызываются.

Код функции должен размещаться в начале программы, вернее, до того места, где мы захотим воспользоваться функцией.

Функция определяется с помощью инструкции `def`.

```
def add(x, y):
        return x + y
```

Данная функция производит операцию сложения двух аргументов и возвращает сумму. Первая строчка этого примера является описанием нашей функции. `add` - идентификатор, то есть имя функции. После идентификатора в круглых скобках идет список параметров, которые получает наша функция. Список состоит из перечисленных через запятую идентификаторов параметров. В нашем случае список состоит из двух величин - `x` и `y`. В конце строки ставится двоеточие.

Далее идет тело функции, оформленное в виде блока, то есть с отступом.

Инструкция `return` может встречаться в произвольном месте функции, ее исполнение завершает работу функции и возвращает указанное значение в место вызова. Если функция не возвращает значения, то инструкция return используется без возвращаемого значения. В функциях, которым не нужно возвращать значения, инструкция return может отсутствовать.

Соответственно теперь мы можем вызвать эту функцию в любой части программы и она вернет нам сумму.

Ввод / вывод

Как уже обсуждали раньше для вывода значений на экран есть функция print(). Внутри круглых скобок через запятую мы пишем то, что хотим вывести.

Для ввода данных в программу мы используем функцию `input()`. Она считывает одну строку.

```
print('Как вас зовут?')
name = input()  # считываем строку и кладём её в переменную name
print('Здравствуйте, ' + name + '!')
```

Мы будем писать программы, которые считывают данные, перерабатывают их и выводят какой-то результат. При запуске на компьютере такие программы считывают данные, которые пользователь вводит с клавиатуры, а результат выводят на экран.

Объекты

В Питоне все данные называются объектами. Число 2 представляется объектом «число 2», строка `'hello'` – это объект «строка `'hello'`».

Каждый объект относится к какому-то типу. Строки хранятся в объектах типа `str`, целые числа хранятся в объектах типа `int`, дробные числа (вещественные числа) — в объектах типа `float`. Тип объекта определяет, какие действия можно делать с объектами этого типа. Например, если в переменных `first` и `second` лежат объекты типа `int`, то их можно перемножить, а если в них лежат объекты типа `str`, то их перемножить нельзя:

```
first = 5
second = 7
print(first * second)
first = '5'
second = '7'
print(first * second)
```

Чтобы преобразовать строку из цифр в целое число, воспользуемся функцией `int()`. Например, `int('23')` вернет число 23.

Встроенные функции, выполняющие преобразование типов

bool(x) - преобразование к типу bool, использующая стандартную процедуру проверки истинности. Если x является ложным или опущен, возвращает значение False, в противном случае она возвращает True.

bytearray([источник [, кодировка [ошибки]]]) - преобразование к bytearray. Bytearray - изменяемая последовательность целых чисел в диапазоне 0≤X<256. Вызванная без аргументов, возвращает пустой массив байт.

bytes([источник [, кодировка [ошибки]]]) - возвращает объект типа bytes, который является неизменяемой последовательностью целых чисел в диапазоне 0≤X<256. Аргументы конструктора интерпретируются как для bytearray().

complex([real[, imag]]) - преобразование к комплексному числу.

dict([object]) - преобразование к словарю.

float([X]) - преобразование к числу с плавающей точкой. Если аргумент не указан, возвращается 0.0.

frozenset([последовательность]) - возвращает неизменяемое множество.

int([object], [основание системы счисления]) - преобразование к целому числу.

list([object]) - создает список.

memoryview([object]) - создает объект memoryview.

object() - возвращает безликий объект, являющийся базовым для всех объектов.

range([start=0], stop, [step=1]) - арифметическая прогрессия от start до stop с шагом step.

set([object]) - создает множество.

slice([start=0], stop, [step=1]) - объект среза от start до stop с шагом step.

str([object], [кодировка], [ошибки]) - строковое представление объекта. Использует метод __str__.

tuple(obj) - преобразование к кортежу.

Другие встроенные функции

abs(x) - Возвращает абсолютную величину (модуль числа).

all(последовательность) - Возвращает True, если все элементы истинные (или, если последовательность пуста).

any(последовательность) - Возвращает True, если хотя бы один элемент - истина. Для пустой последовательности возвращает False.

ascii(object) - Как repr(), возвращает строку, содержащую представление объекта, но заменяет не-ASCII символы на экранированные последовательности.

bin(x) - Преобразование целого числа в двоичную строку.

callable(x) - Возвращает True для объекта, поддерживающего вызов (как функции).

chr(x) - Возвращает односимвольную строку, код символа которой равен x.

classmethod(x) - Представляет указанную функцию методом класса.

compile(source, filename, mode, flags=0, dont_inherit=False) - Компиляция в программный код, который впоследствии может выполниться функцией eval или exec. Строка не должна содержать символов возврата каретки или нулевые байты.

delattr(object, name) - Удаляет атрибут с именем 'name'.

dir([object]) - Список имен объекта, а если объект не указан, список имен в текущей локальной области видимости.

divmod(a, b) - Возвращает частное и остаток от деления a на b.

enumerate(iterable, start=0) - Возвращает итератор, при каждом проходе предоставляющем кортеж из номера и соответствующего члена последовательности.

eval(expression, globals=None, locals=None) - Выполняет строку программного кода.

exec(object[, globals[, locals]]) - Выполняет программный код на Python.

filter(function, iterable) - Возвращает итератор из тех элементов, для которых function возвращает истину.

format(value[,format_spec]) - Форматирование (обычно форматирование строки).

getattr(object, name ,[default]) - извлекает атрибут объекта или default.

globals() - Словарь глобальных имен.

hasattr(object, name) - Имеет ли объект атрибут с именем 'name'.

hash(x) - Возвращает хеш указанного объекта.

help([object]) - Вызов встроенной справочной системы.

hex(x) - Преобразование целого числа в шестнадцатеричную строку.

id(object) - Возвращает "адрес" объекта. Это целое число, которое гарантированно будет уникальным и постоянным для данного объекта в течение срока его существования.

input([prompt]) - Возвращает введенную пользователем строку. Prompt - подсказка пользователю.

isinstance(object, ClassInfo) - Истина, если объект является экземпляром ClassInfo или его подклассом. Если объект не является объектом данного типа, функция всегда возвращает ложь.

issubclass(класс, ClassInfo) - Истина, если класс является подклассом ClassInfo. Класс считается подклассом себя.

iter(x) - Возвращает объект итератора.

len(x) - Возвращает число элементов в указанном объекте.

locals() - Словарь локальных имен.

map(function, iterator) - Итератор, получившийся после применения к каждому элементу последовательности функции function.

max(iter, [args ...] * [, key]) - Максимальный элемент последовательности.

min(iter, [args ...] * [, key]) - Минимальный элемент последовательности.

next(x) - Возвращает следующий элемент итератора.

oct(x) - Преобразование целого числа в восьмеричную строку.

open(file, mode='r', buffering=None, encoding=None, errors=None, newline=None, closefd=True) - Открывает файл и возвращает соответствующий поток.

ord(c) - Код символа.

pow(x, y[, r]) - (x ** y) % r.

reversed(object) - Итератор из развернутого объекта.

repr(obj) - Представление объекта.

print([object, ...], *, sep=" ", end='\n', file=sys.stdout) - Печать.

property(fget=None, fset=None, fdel=None, doc=None)

round(X [, N]) - Округление до N знаков после запятой.

setattr(объект, имя, значение) - Устанавливает атрибут объекта.

sorted(iterable[, key][, reverse]) - Отсортированный список.

staticmethod(function) - Статический метод для функции.

sum(iter, start=0) - Сумма членов последовательности.

super([тип [, объект или тип]]) - Доступ к родительскому классу.

type(object) - Возвращает тип объекта.

type(name, bases, dict) - Возвращает новый экземпляр класса name.

vars([object]) - Словарь из атрибутов объекта. По умолчанию - словарь локальных имен.

zip(*iters) - Итератор, возвращающий кортежи, состоящие из соответствующих элементов аргументов-последовательностей.

Вопросы

◄ Проверочное тестирование по лекциям для самокотроля

Перейти на...

Итоговый тест(доступен на сессии) ►