

Колледж космического машиностроения и технологий

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей

программного обеспечения для компьютерных систем

специальность 09.02.03 Программирование в компьютерных системах

Выполнил студент:

Роголев В.А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Гусятинер Л.Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2020

Содержание отчёта

Содержание отчёта

4, Техника работы с линейными и разветвляющимися программами

Приложения: input.py, print.py, std1.py, std2.py, Date.py, Format.py

задание 1.py задание 2.py

Листинг 1. input.py:

'''

Задание. Разработать программы по темам

- input

Функция input() в Python, ввод данных с клавиатура.

https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-input/

'''

# test.py

def str\_to\_num(line):

"""функция конвертирует строку в число"""

line = line.strip()

# если в строке только цифры

if line.isdigit():

return int(line)

# если строка содержит точку или запятую

elif '.' in line or ',' in line:

# если из строки убрать точку или запятую

# и при этом в строке останутся только цифры

if any(line.replace(x, '').isdigit() for x in ['.', ',']):

return float(line.replace(',', '.'))

else:

# ошибка

print('Это не число!\n')

return None

print('\nДля выхода из программы введите Ctrl+C')

print('Для окончания ввода цифр нажмите Enter\n')

nums = []

while True:

inpt = input('Ожидается ввод числа или Enter:')

if inpt == '':

# Закончить ввод чисел

break

n = str\_to\_num(inpt)

if n is not None:

nums.append(n)

if nums:

if len(nums) == 1:

print('Вы ввели одну цифру: ', nums[0])

else:

print('\nВыберите действие:')

print(' сложить цифры введите 1;')

print(' умножить цифры введите 2.\n')

rez = None

while True:

inpt = input('Введите 1 или 2:')

inpt = inpt.strip()

if inpt == '1':

rez = sum(nums)

print('Сумма введенных чисел:', rez)

elif inpt == '2':

rez = 1

for i in nums:

rez \*= i

print('Произведение введенных чисел:', rez)

else:

print('Неправильное действие.\n')

if rez is not None:

break

else:

print('Вы ничего не ввели.')

Листинг 2. print.py:

'''

- print

Функция print() в Python, печатает объект.

https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-print/

'''

print('Hello')

# Hello

print('Hello', 'how are you?')

# Hello how are you?

print('Hello', 'how are you?', sep='---')

# Hello---how are you?

print('Раз', end='=>')

print('Два')

# Раз=>Раз

print(11, 12, 13, 14, sep=';')

# 11;12;13;14

Листинг 3. std1.py:

'''

- stdin, stdout, stderr модуля sys

Объекты stdin, stdout, stderr модуля sys в Python.

https://docs-python.ru/standart-library/modul-sys-python/obekty-stdin-stdout-stderr-modulja-sys/

'''

import sys

stdout = sys.stdout

try:

sys.stdout = open('file.txt', 'w')

print('blah')

finally:

# Закрываем file.txt

sys.stdout.close()

sys.stdout = sys.\_\_stdout\_\_

Листинг 4. std2.py:

'''

- stdin, stdout, stderr модуля sys

Объекты stdin, stdout, stderr модуля sys в Python.

https://docs-python.ru/standart-library/modul-sys-python/obekty-stdin-stdout-stderr-modulja-sys/

'''

import sys, time

def teleprint(\*args, delay=0.05, str\_join=' '):

text = str\_join.join(str(x) for x in args)

n = len(text)

for i, char in enumerate(text, 1):

if i == n:

char = f'{char}\n'

sys.stdout.write(char)

sys.stdout.flush()

time.sleep(delay)

# Строка будет печататься с задержкой, как в телетексте...

teleprint('Печать с задержкой!', 10, 12.5, 'Super!!!', delay=0.07)

# Печать с задержкой!, 10, 12.5, Super!!

Листинг 5. Date.py:

import re

t = '06.10.1990'

r = re.findall(r'(\d{2})\.(\d{2})\.(\d{4})', t)[0]

d, m, y = tuple(map(int, r))

print(d, m, y)

d\_ls = (

'первое', 'второе', 'третье', 'четвёртое', 'пятое',

'шестое', 'седьмое', 'восьмое', 'девятое', 'десятое',

'одиннадцатое', 'двенадцатое', 'тринадцатое', 'четырнадцатое', 'пятнадцатое',

'шестьнадцатое', 'семьнадцатое', 'восемьнадцатое', 'девятнадцатое', 'двадцатое')

# assert 0 < d < 31

if d == 30:

print('тридцатое', end=' ')

else:

if d > 30:

print('тридцать', end=' ')

d %= 30

elif d > 20:

print('двадцать', end=' ')

d %= 20

print(d\_ls[d-1], end=' ')

Листинг 6. Format.py:

'''

Задание. Разработать программу

- форматная строка и метод формат

'''

coord = (3, 5)

print('X: {0[0]}; Y: {0[1]}'.format(coord))

print('X: 3; Y: 5')

print("repr() shows quotes: {!r}; str() doesn't: {!s}".format('test1', 'test2'))

print("repr() shows quotes: 'test1'; str() doesn't: test2")

print('{:<30}'.format('left aligned'))

print('left aligned ')

print('{:>30}'.format('right aligned'))

print(' right aligned')

print('{:^30}'.format('centered'))

print(' centered ')

print('{:\*^30}'.format('centered'))

print('\*\*\*\*\*\*\*\*\*\*\*centered\*\*\*\*\*\*\*\*\*\*\*')

print('{:+f}; {:+f}'.format(3.141592, -3.141592))

print('+3.141592; -3.141592')

print('{: f}; {: f}'.format(3.141592, -3.141592))

print(' 3.141592; -3.141592')

print('{:-f}; {:-f}'.format(3.141592, -3.141592))

print('3.141592; -3.141592')

print("int: {0:d}; hex: {0:x}; oct: {0:o}; bin: {0:b}".format(42))

print('int: 42; hex: 2a; oct: 52; bin: 101010')

print("int: {0:d}; hex: {0:#x}; oct: {0:#o}; bin: {0:#b}".format(42))

print('int: 42; hex: 0x2a; oct: 0o52; bin: 0b101010')

points = 19.5

total = 22

print('Correct answers: {:.2%}'.format(points/total))

print('Correct answers: 88.64%')

Листинг 7. Задание 1.py:

'''

- Задание 1. Разработать программу для печати даты прописью

Пример ввода: 15.12.1983

Пример вывода: Пятнадцатое декабря одна тысяча девятсот восемьдесят третьего года

'''

def get\_date(date):

days = ['первое', 'второе', 'третье', 'четвёртое',

'пятое', 'шестое', 'седьмое', 'восьмое',

'девятое', 'десятое', 'одиннадцатое', 'двенадцатое',

'тринадцатое', 'четырнадцатое', 'пятнадцатое', 'шестнадцатое',

'семнадцатое', 'восемнадцатое', 'девятнадцатое', 'двадцатое',

'двадцать первое', 'двадцать второе', 'двадцать третье',

'двадацать четвёртое', 'двадцать пятое', 'двадцать шестое',

'двадцать седьмое', 'двадцать восьмое', 'двадцать девятое',

'тридцатое', 'тридцать первое']

months = ['января', 'февраля', 'марта', 'апреля', 'мая', 'июня',

'июля', 'августа', 'сентября', 'октября', 'ноября', 'декабря']

date = date.split('.')

return (days[int(date[0]) - 1] + ' ' + months[int(date[1]) - 1] + ' ' + date[2] + ' года')

date = input()

print(get\_date(date))

Листинг 8. Задание 2.py:

'''

Задание 2. Разработать программу с меню для демонстрации работы с типами данных:

список(list), словарь(dict), множество(set)

Меню -> выбор типа данных -> выбор метода -> краткая справка

'''

options = int(input("Choose from Menu:\n"

"list: 1.\n"

"dict: 2.\n"

"set: 3.\n"))

listMethods = ['list.append(x) Добавляет элемент в конец списка\n',

'list.extend(L) Расширяет список list, добавляя в конец все элементы списка L\n',

'list.insert(i, x) Вставляет на i-ый элемент значение x\n',

'list.remove(x) Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует\n'

'list.pop([i]) Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент\n'

'list.index(x, [start [, end]]) Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end)\n'

'list.count(x) Возвращает количество элементов со значением x\n'

'list.sort([key=функция]) Сортирует список на основе функции\n'

'list.reverse() Разворачивает список\n'

'list.copy() Поверхностная копия списка\n'

'list.clear() Очищает список\n']

dictMethods = ['dict.clear() - очищает словарь.\n',

'dict.copy() - возвращает копию словаря.\n',

'classmethod dict.fromkeys(seq[, value]) - создает словарь с ключами из seq и значением value (по умолчанию None).\n',

'dict.get(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а возвращает default (по умолчанию None).\n',

'dict.items() - возвращает пары (ключ, значение).\n',

'dict.keys() - возвращает ключи в словаре.\n',

'dict.pop(key[, default]) - удаляет ключ и возвращает значение. Если ключа нет, возвращает default (по умолчанию бросает исключение).\n',

'dict.popitem() - удаляет и возвращает пару (ключ, значение). Если словарь пуст, бросает исключение KeyError. Помните, что словари неупорядочены.\n',

'dict.setdefault(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а создает ключ с значением default (по умолчанию None).\n',

'dict.update([other]) - обновляет словарь, добавляя пары (ключ, значение) из other. Существующие ключи перезаписываются. Возвращает None (не новый словарь!).\n',

'dict.values() - возвращает значения в словаре.\n']

setMethods = ['len(s) - число элементов в множестве (размер множества).\n',

'x in s - принадлежит ли x множеству s.\n',

'set.isdisjoint(other) - истина, если set и other не имеют общих элементов.\n',

'set == other - все элементы set принадлежат other, все элементы other принадлежат set.\n',

'set.issubset(other) или set <= other - все элементы set принадлежат other.\n',

'set.issuperset(other) или set >= other - аналогично.\n',

'set.union(other, ...) или set | other | ... - объединение нескольких множеств.\n',

'set.intersection(other, ...) или set & other & ... - пересечение.\n',

'set.difference(other, ...) или set - other - ... - множество из всех элементов set, не принадлежащие ни одному из other.\n',

'set.symmetric\_difference(other); set ^ other - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.\n',

'set.copy() - копия множества.\n',

'И операции, непосредственно изменяющие множество:\n',

'set.update(other, ...); set |= other | ... - объединение.\n',

'set.intersection\_update(other, ...); set &= other & ... - пересечение.\n',

'set.difference\_update(other, ...); set -= other | ... - вычитание.\n',

'set.symmetric\_difference\_update(other); set ^= other - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.\n',

'set.add(elem) - добавляет элемент в множество.\n',

'set.remove(elem) - удаляет элемент из множества. KeyError, если такого элемента не существует.\n',

'set.discard(elem) - удаляет элемент, если он находится в множестве.\n',

'set.pop() - удаляет первый элемент из множества. Так как множества не упорядочены, нельзя точно сказать, какой элемент будет первым.\n',

'set.clear() - очистка множества.\n']

menuList = [listMethods, dictMethods, setMethods]

print(\*menuList[options - 1])