

Лабораторная работа № 2

Python: условия

Содержание

1 Цель работы.....	2
2 Краткие теоретические сведения.....	2
3 Оборудование рабочего места.....	2
4 Задание.....	2
5 Содержание отчета.....	2
6 Порядок выполнения работы.....	3
6.1 Использование условий.....	3
6.2 Самостоятельная работа.....	6

1 Цель работы

Изучить порядок составления и использования условий на языке программирования Python.

2 Краткие теоретические сведения

Программирование — это процесс проектирования, написания и отладки программ. Что такое программа? Программа — это последовательность инструкций, предназначенная для исполнения устройством управления вычислительной машины [1]. Есть, также, официальные трактовки термина «Программа»:

Программа — данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определённого алгоритма. [2]

Программа — представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения. [3]

3 Оборудование рабочего места

Персональный компьютер под управлением ОС Linux или Windows с установленной средой программирования IDLE и языком программирования Python версии 3.4 и выше.

4 Задание

- Ознакомиться с формальными определениями, требующими использования условной конструкции.
- Выполнить примеры, приведенные в разделе «Порядок выполнения работы».
- Решить примеры для самостоятельной работы.

5 Содержание отчета

Отчет представляется в виде текстового документа, содержащего скриншоты окна редактора, содержащего код программы и командного окна с результатом выполнения программы для всех самостоятельных заданий.

6 Порядок выполнения работы

6.1 Использование условий

В работе большого количества автоматических систем используется проверка условий, которые можно записать формально в виде фразы если [формулировка условия], то [действия, если условие выполнилось] иначе [действие, если условие не выполнилось (или бездействие)]. Например:

- двери лифта на этаже открываются, если лифт подъехал к этажу,
- если обнаружено задымление, срабатывает пожарная сигнализация.

Ещё больше проверок условий и реакции на них выполняются с участием человека:

- водитель останавливается на перекрестке, если на светофоре горит красный свет,
- дорожки посыпают песком, если они покрылись льдом,
- если идет дождь, мы берем зонт или одеваем дождевик,
- и т. д.

Во время выполнения программ также необходим выбор дальнейших действий в зависимости от выполнения (или невыполнения) того или иного условия.

В языке программирования Python для проверки условия используется конструкция `if`. Особенностью языка Python является отсутствие операторных скобок. Для того, чтобы показать объединение блоков, используются отступы (примерно так, как в других языках программирования они используются для повышения читаемости кода).

Вот как выглядит конструкция `if` в общем виде:

```
if условие1:
    блок1
elif условие2:
    блок2
else:
    блок3
```

Обратите внимание! Любой блок может состоять более чем из одной команды. Для того, чтобы показать, что команда относится к блоку, необходимо каждую такую команду вводить после отступа из 4-х пробелов!

Рассмотрим простую задачу, для решения которой необходима условная конструкция.

Пользователь вводит с клавиатуры два числа. Необходимо сравнить числа и вывести на экран сообщение «Первое число больше.», «Второе число больше.» или «Числа равны».

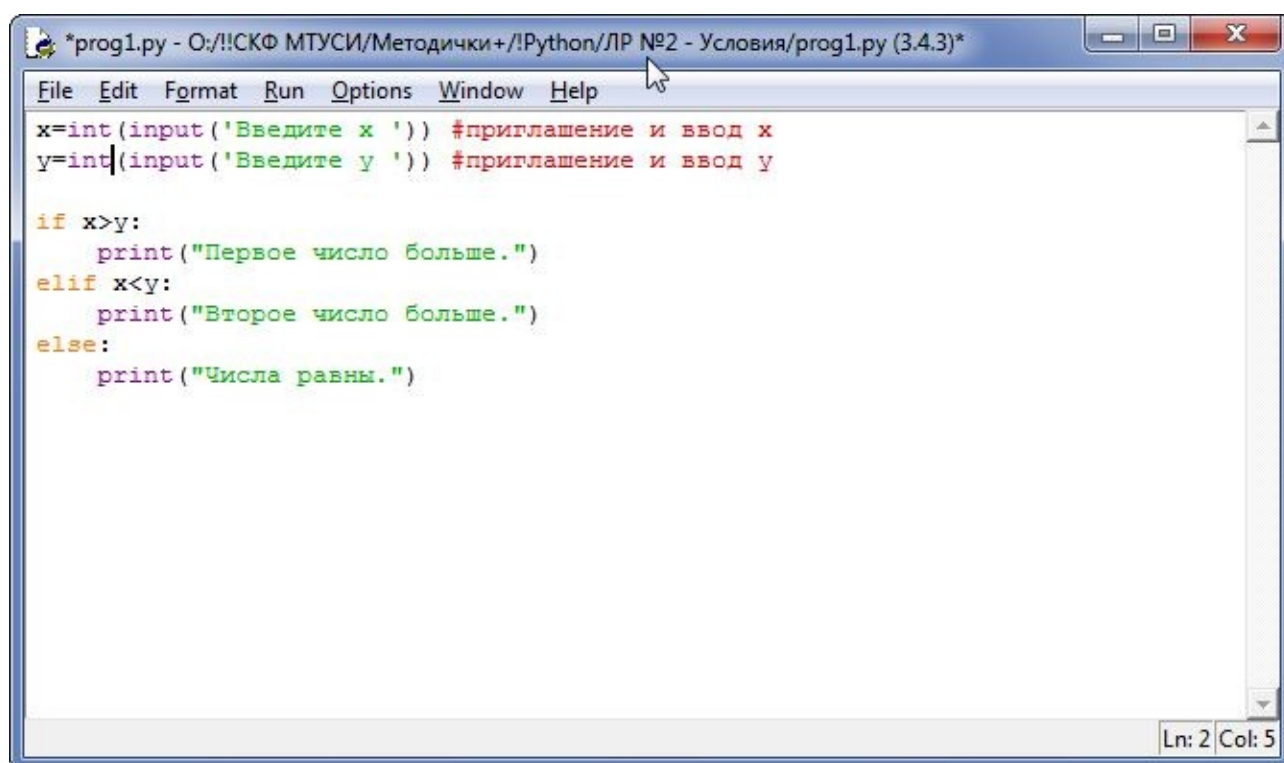
Итак, нам надо обеспечить ввод чисел с клавиатуры:

```
x=int(input('Введите x ')) #приглашение и ввод x
y=int(input('Введите y ')) #приглашение и ввод y
```

Дальше необходимо сравнить два числа. Для сравнения используются операторы сравнения, уже знакомые нам по первой работе. Коротко перечислим их:

оператор	описание
==	равно
!=	не равно
<>	не равно
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно

Записать наше условие можно по-разному, т. к. один раз необходимо будет применить второе условие с помощью строки `elif` (сокращение от «иначе если»). Один из вариантов будет выглядеть так:

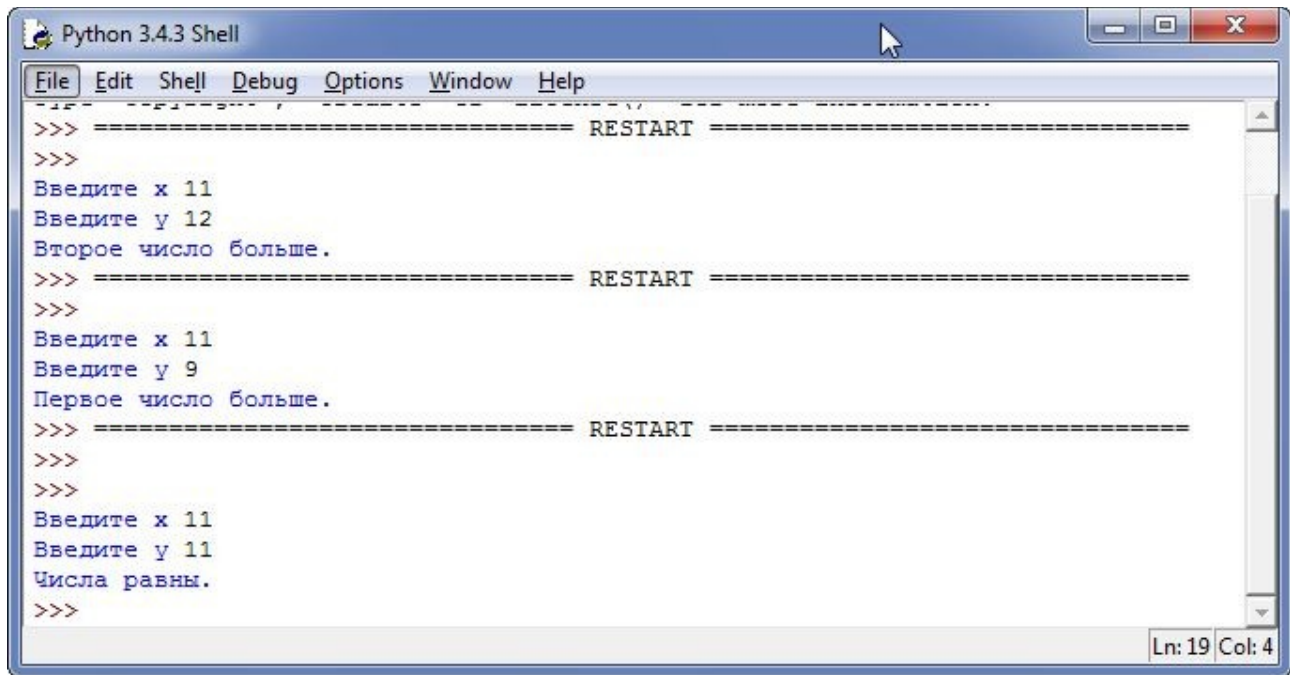


```
*prog1.py - O:/!!СКФ МТУСИ/Методички+!/Python/ЛР №2 - Условия/prog1.py (3.4.3)*
File Edit Format Run Options Window Help
x=int(input('Введите x ')) #приглашение и ввод x
y=int(input('Введите y ')) #приглашение и ввод y

if x>y:
    print("Первое число больше.")
elif x<y:
    print("Второе число больше.")
else:
    print("Числа равны.")

Ln: 2 Col: 5
```

Проверим программу — запустим её несколько раз с разными данными:



The screenshot shows a Python 3.4.3 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a status bar (Ln: 19 Col: 4). The program code is as follows:

```
>>> ===== RESTART =====  
>>>  
Введите x 11  
Введите y 12  
Второе число больше.  
>>> ===== RESTART =====  
>>>  
Введите x 11  
Введите y 9  
Первое число больше.  
>>> ===== RESTART =====  
>>>  
>>>  
Введите x 11  
Введите y 11  
Числа равны.  
>>>
```

Программа работает корректно.

6.2 Самостоятельная работа

1. Используя функцию `random.randint(0,3)` сгенерировать случайное число t от 0 до 3. Сделать скриншот запуска программы.

2. Написать на языке Python программы для решения задачи. Предусмотреть ввод и вывод данных.

Необходимо решить 5 примеров. Номера вариантов заданий n выбираются по правилу:

$$n_i = 4 \cdot (i-1) + t,$$

где t – результат работы генератора в предыдущем задании,

i – порядковый номер выбираемого задания (1,2,3,4,5).

Т.е. выполняется условие $n \bmod 4 = t$.

В отчет по каждой программе включить:

- текст программы
- скриншот окна редактора с готовой программой,
- скриншот окна сразу после выполнения программы.

Варианты заданий:

0. С клавиатуры вводится целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.

1. С клавиатуры вводится целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число.

2. С клавиатуры вводится целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число.

3. С клавиатуры вводятся три целых числа. Найти количество положительных чисел в исходном наборе.

4. С клавиатуры вводятся три целых числа. Найти количество положительных и количество отрицательных чисел в исходном наборе.

5. С клавиатуры вводятся два числа. Вывести большее из них.

6. С клавиатуры вводятся два числа. Вывести порядковый номер меньшего из них.

7. С клавиатуры вводятся два числа. Вывести вначале большее, а затем меньшее из них.

8. С клавиатуры вводятся значения двух переменных с плавающей точкой: A , B . Перераспределить значения данных переменных так, чтобы в A оказалось меньшее из значений, а в B — большее. Вывести новые значения переменных A и B .

9. С клавиатуры вводятся значения двух переменных целого типа: A и B . Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B .

10. С клавиатуры вводятся значения двух переменных целого типа: A и B . Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B .

11. С клавиатуры вводятся три числа. Найти наименьшее из них.

12. С клавиатуры вводятся три числа. Найти среднее из них (то есть число, расположен-

ное между наименьшим и наибольшим).

13. С клавиатуры вводятся три числа. Вывести вначале наименьшее, а затем наибольшее из данных чисел.

14. С клавиатуры вводятся три числа. Найти сумму двух наибольших из них.

15. С клавиатуры вводятся значения трех переменных с плавающей точкой: А, В, С. Если их значения упорядочены по возрастанию, то удвоить их; в противном случае заменить значение каждой переменной на его модуль. Вывести новые значения переменных А, В, С.

16. С клавиатуры вводятся значения трех переменных с плавающей точкой: А, В, С. Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных А, В, С.

17. С клавиатуры вводятся три целых числа, одно из которых отлично от двух других, равных между собой. Определить порядковый номер числа, отличного от остальных.

18. С клавиатуры вводятся четыре целых числа, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных.

19. На числовой оси расположены три точки: А, В, С (целые числа, вводятся с клавиатуры). Определить, какая из двух последних точек (В или С) расположена ближе к А, и вывести эту точку и ее расстояние от точки А.