

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

Отчет по лабораторной работе №3

По дисциплине основы кроссплатформенного программирования

«Условные операторы и циклы в языке Python»

Выполнила:

студентк группы ИТС-б-о-21-1

Аллаёров Жамшид Хасан угли

(подпись)

Проверил: Доцент, к.т.н, доцент
кафедры

инфокоммуникаций

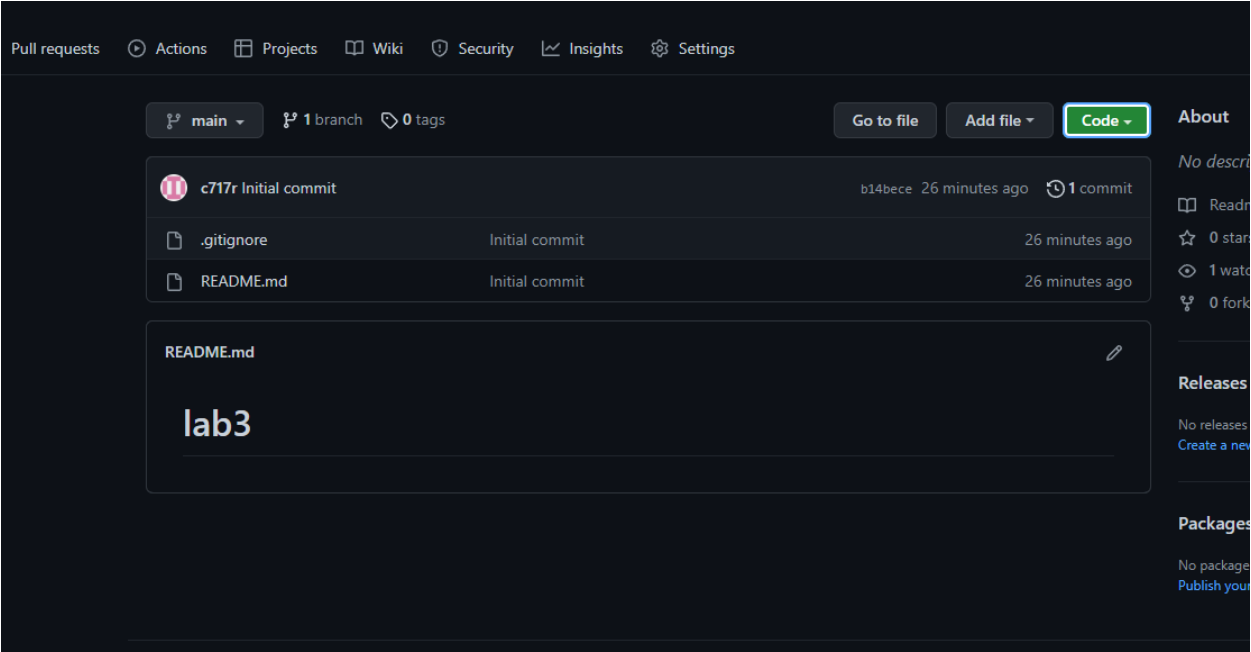
Воронкин Р. А.

Работа защищена с оценкой:

(подпись)

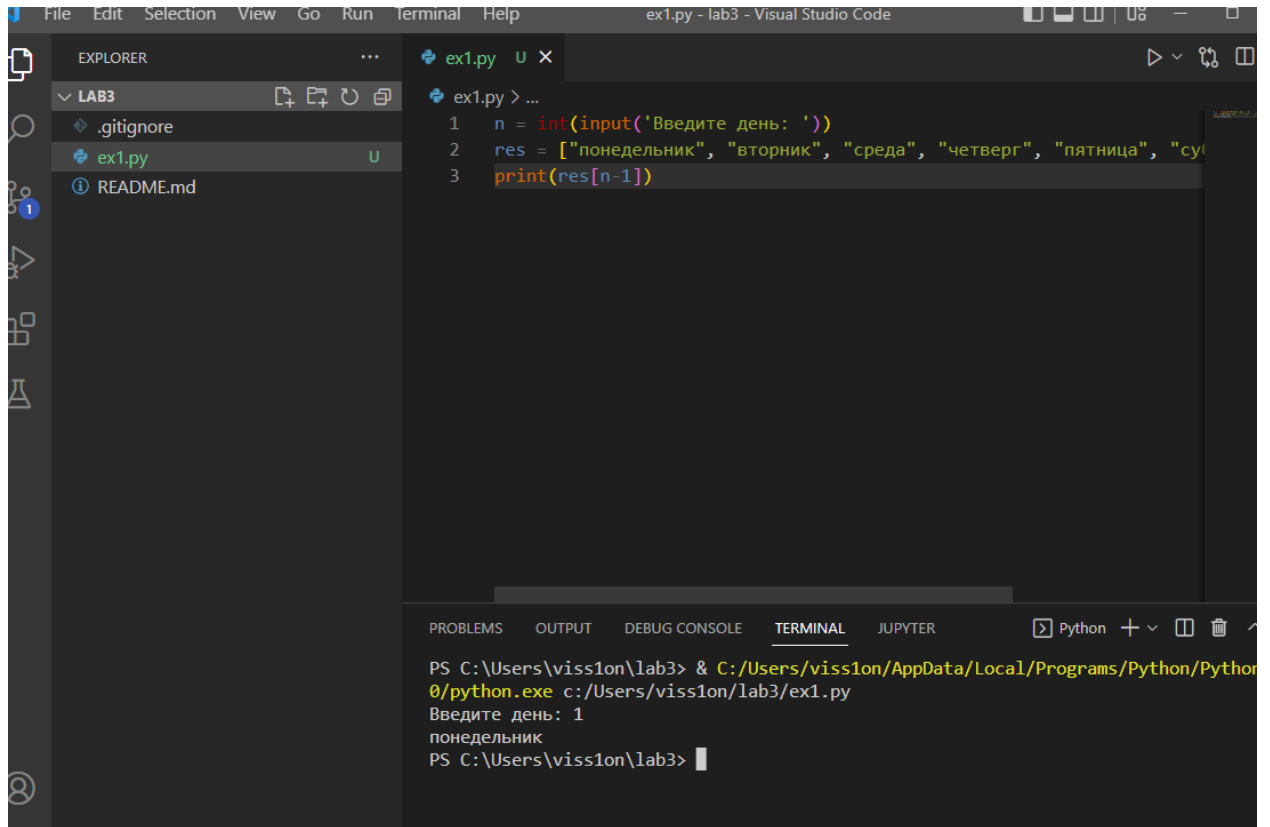
Ставрополь, 2022

. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python.



Дано число (). Вывести на экран название дня недели, который соответствует

этому номеру.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a file named `ex1.py` under a folder named `LAB3`. The main editor window shows the content of `ex1.py`:

```
1 n = int(input('Введите день: '))
2 res = ["понедельник", "вторник", "среда", "четверг", "пятница", "суббота", "воскресенье"]
3 print(res[n-1])
```

The bottom panel shows the TERMINAL output:

```
PS C:\Users\viss1on\lab3> & C:/Users/viss1on/AppData/Local/Programs/Python/Python310/python.exe c:/Users/viss1on/lab3/ex1.py
Введите день: 1
понедельник
PS C:\Users\viss1on\lab3>
```

Начав тренировки, спортсмен пробежал 10 км. Каждый следующий день он увеличивал дневную норму на 10% от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?

The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a folder named 'LAB3' containing files: '.gitignore', 'ex1.py', 'ex3', and 'README.md'. The file 'ex3' is selected. The main editor displays the content of 'ex3', which is a Python script:

```
1 day = 1
2 distance = 20
3 while distance < 20:
4     distance += (distance/100)*10
5     day += 1
6 print(day+1)
```

At the bottom, the TERMINAL pane shows the execution of the script using Python 3. The output is:

```
PS C:\Users\visssion\lab3> & C:/Users/visssion/AppData/Local/Programs/Python/Python39/python.exe c:/Users/visssion/lab3/ex1.py
Введите день: 1
понедельник
PS C:\Users\visssion\lab3> & C:/Users/visssion/AppData/Local/Programs/Python/Python39/python.exe c:/Users/visssion/lab3/ex3
2
PS C:\Users\visssion\lab3>
```

Зафиксируйте сделанные изменения в репозитории

The screenshot shows the Visual Studio Code interface. The Explorer pane shows the same 'LAB3' folder. The main editor displays the content of 'ex3'. The TERMINAL pane shows the execution of git commands to commit and push the changes:

```
2
PS C:\Users\visssion\lab3> git add .
PS C:\Users\visssion\lab3> git commit --m "dne"
[main d84c449] dne
2 files changed, 9 insertions(+)
create mode 100644 ex1.py
create mode 100644 ex3
PS C:\Users\visssion\lab3> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 562 bytes | 281.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/c717r/lab3.git
 b14bece..d84c449 main -> main
PS C:\Users\visssion\lab3>
```

1. Для чего нужны диаграммы деятельности UML?

Диаграмма активности UML позволяет более детально визуализировать конкретный случай использования. Это поведенческая диаграмма, которая иллюстрирует поток деятельности через систему. Диаграммы активности UML также могут быть использованы для отображения потока событий в бизнес-процессе.

2. Что такое состояние действия и состояние деятельности?

Состояния деятельности и действия Состояние деятельности (activity state) - состояние в графе деятельности, которое служит для представления процедурной последовательности действий, требующих определенного времени.

3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?

Переход как элемент диаграммы состояний был рассмотрен в лекции 9. При построении диаграммы деятельности используются только нетриггерные переходы, т. е. такие, которые происходят сразу после завершения деятельности или выполнения соответствующего действия. Такой переход передает управление в последующее состояние сразу, как только закончится действие или *деятельность* в предыдущем состоянии. На диаграмме такой переход изображается сплошной линией со стрелкой.

Если из *состояния действия* выходит единственный переход, то его можно никак не помечать. Если же таких переходов несколько, то при моделировании последовательной деятельности запускается только один из них. В этом случае для каждого из таких переходов должно быть явно записано собственное *сторожевое условие* в прямых скобках (см. лекцию 9). При этом для всех выходящих из некоторого состояния деятельности

переходов должно выполняться требование истинности только одного из них. Подобный случай встречается тогда, когда последовательно выполняемая *деятельность* должна разделиться на *альтернативные* ветви в зависимости от значения промежуточного результата. Такая ситуация получила название ветвления, а для ее обозначения применяется *специальный символ решения*.

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

Разветвляющийся алгоритм – это алгоритм, в котором последовательность выполнения операций зависит от определенных условий. Если в алгоритме присутствует «действие1» и «действие2» (то есть ветвь 1 и ветвь 2), то это разветвляющийся алгоритм с полной альтернативой.

5. Чем отличается разветвляющийся алгоритм от линейного?

Линейный алгоритм - алгоритм, все этапы которого выполняются однократно и строго последовательно.

Разветвляющийся алгоритм - алгоритм, содержащий хотя бы одно условие, в результате проверки которого ЭВМ обеспечивает переход на один из двух возможных шагов.

6. Что такое условный оператор? Какие существуют его формы?

Ветвление в программировании — операция, применяющаяся в случаях, когда выполнение или невыполнение некоторого набора команд должно зависеть от выполнения или невыполнения некоторого условия. Ветвление — одна из трёх (наряду с последовательным исполнением команд и циклом) базовых конструкций структурного программирования.

Основные формы реализации ветвления в императивных языках программирования — *условный оператор* (оператор `if`) и *оператор многозначного выбора* (переключатель, `case`, `switch`). В ранних языках

низкого уровня ветвление реализуется посредством оператора условного перехода.

7. Какие операторы сравнения используются в Python?

В Python есть шесть операций сравнения. Все они имеют одинаковый приоритет, который выше, чем у логических операций.

...

Разрешенные операции сравнения:

- $x < y$ - строго x меньше y ,
- $x \leq y$ - x меньше или равно y ,
- $x > y$ - строго x больше y ,
- $x \geq y$ - x больше или равно y ,
- $x == y$ - x равно y ,
- $x != y$ - x не равно y .

8. Что называется простым условием? Приведите примеры.

Простым условием (отношением) называется выражение, составленное из двух арифметических выражений или двух текстовых величин (иначе их еще называют операндами), связанных одним из знаков: $<$ - меньше, чем...

Например, простыми отношениями являются следующие: $x - y > 10$;

$k \leq \sqrt{c} + \text{abs}(a + b)$; $9 < 11$; 'мама' <> 'папа'.

9. Что такое составное условие? Приведите примеры.

Составные условия — это условия, состоящие из двух или более простых условий, соединенных с помощью логических операций: `and` , `or` , `not` . Простые условия при этом заключаются в скобки.

Примеры составных условий:

($a > 8$)

($x = 0$) or (x

not ($a = 0$) or ($b = 0$)

10. Какие логические операторы допускаются при составлении сложных условий?

Логические операции образуют сложное (составное) условие из нескольких простых (два или более) условий.

...

В C++ существует три логические операции:

Логическая операция И && , нам уже известная;

Логическая операция ИЛИ || ;

Логическая операция НЕ ! или логическое отрицание.