

1. Для чего нужны диаграммы деятельности UML?

Диаграмма деятельности - это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, однако, по сравнению с последней, у ней есть явные преимущества: поддержка многопоточности и объектно-ориентированного проектирования.

2. Что такое состояние действия и состояние деятельности?

Состояние действия – различные операции и манипуляции наподобие вычисления выражения, выполнение операции над объектом, отправка ему сигнала и т.д. Оно не может быть подвержено декомпозиции, состояние действия атомарно.

Состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, они не атомарны. Можно считать, что это частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции.

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

Линия со стрелкой для переходов, ромб для ветвлений

4. Какой алгоритм является алгоритмом разветвляющейся структуры?

Алгоритм разветвляющейся структуры - это алгоритм, в котором вычислительный процесс осуществляется либо по одной, либо по другой ветви, в зависимости от выполнения некоторого условия. Программа разветвляющейся структуры реализует такой алгоритм. В программе разветвляющейся структуры имеется один или несколько условных операторов. Для программной реализации условия используется логическое выражение. В сложных структурах с большим числом ветвей применяют оператор выбора.

5. Чем отличается разветвляющийся алгоритм от линейного?

Наличием нескольких возможных ветвей осуществления вычислительного процесса

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

Условный оператор позволяет выполнить какое-либо действие если выполняется условие. If, elif, else, while, for

7. Какие операторы сравнения используются в Python?

<, <=, ==, >=, >

8. Что называется простым условием? Приведите примеры.

Простое условие состоит из одного оператора. Например

`X = 4`

`If x = 4`

`Print ("x равен 4")`

9. Что такое составное условие? Приведите примеры.

Составное условие – условие, требующее выполнения сразу нескольких условий. Например, выполнение структуры с операторами `if` и `or` или `and` одновременно:

`X = 4`

`If x >= 2 and x <= 7:`

`Print("x попадает в промежуток")`

10. Какие логические операторы допускаются при составлении сложных условий?

`If`, `else`, `elif`, `range`, `while`, `break`, `continue`, `for`

11. Может ли оператор ветвления содержать внутри себя другие ветвления?

Нет, не может

12. Какой алгоритм является алгоритмом циклической структуры?

Алгоритм циклической структуры - это алгоритм, в котором происходит многократное повторение одного и того же участка программы. Такие повторяемые участки вычислительного процесса называются циклами.

13. Типы циклов в языке Python.

`While`, `for`

14. Назовите назначение и способы применения функции `range` .

Составление последовательности чисел с определенным шагом и без, возрастающие и убывающие.

15. Как с помощью функции `range` организовать перебор значений от 15 до 0 с шагом 2?

`Range (15, 0, 2)`

16. Могут ли быть циклы вложенными?

Могут

17. Как образуется бесконечный цикл и как выйти из него?

Бесконечный цикл в Python получить легко - надо задать условие, при котором всегда будет возвращаться True. Завершить работу программы можно будет, только принудительно закрыв программу или с помощью ключевого слова break.

18. Для чего нужен оператор break ?

Для досрочного прерывания работы цикла while

19. Где употребляется оператор continue и для чего он используется?

Данный оператор запускает цикл заново, при этом код, расположенный после данного оператора, не выполняется

20. Для чего нужны стандартные потоки stdout и stderr?

Файловые объекты sys.stdout и sys.stderr используются интерпретатором для стандартного ввода, вывода и ошибок:

sys.stdout - используется для вывода оператором print() и выражений, которые возвращают значение, а также для подсказок input()

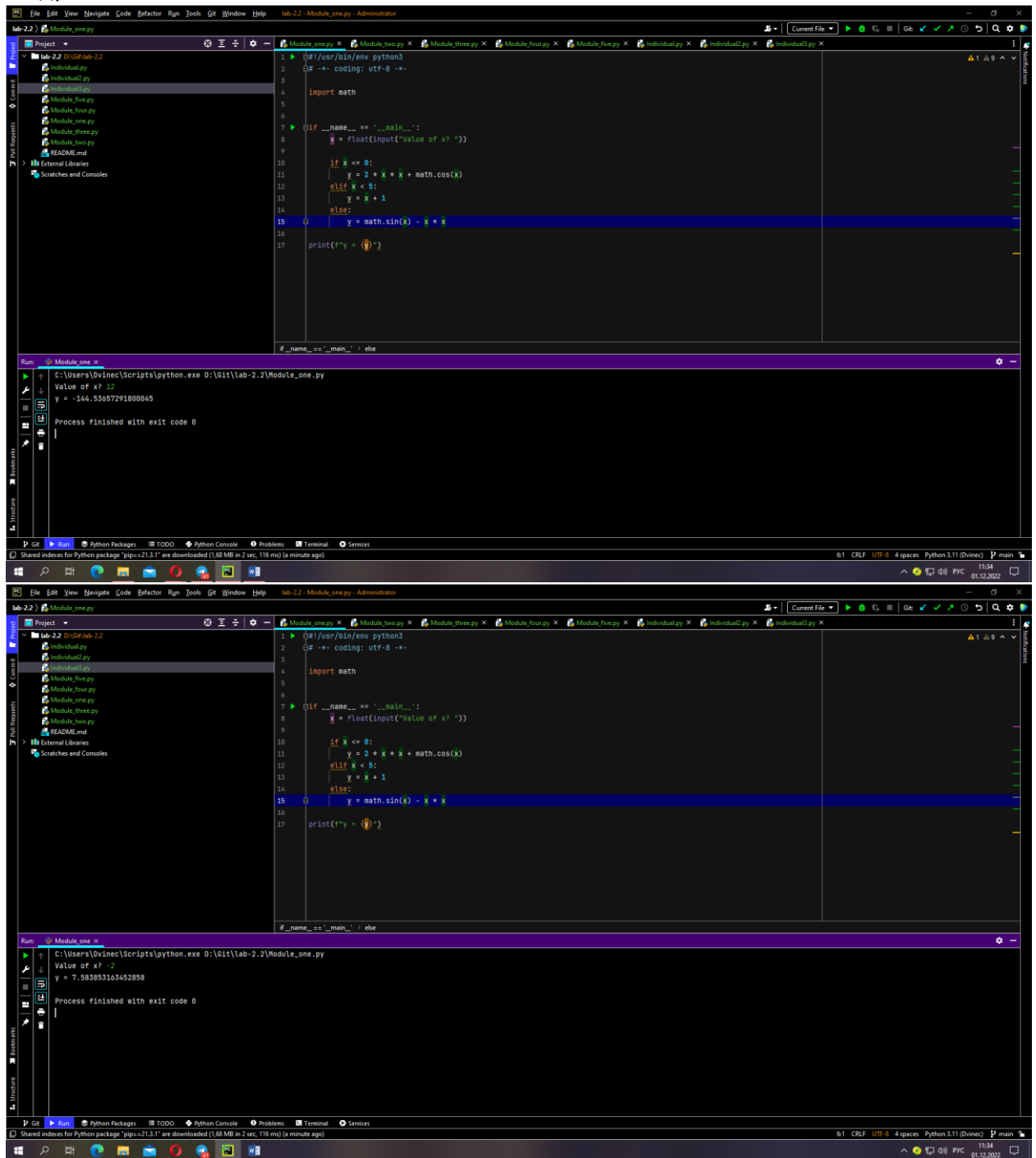
sys.stderr - сообщения об ошибках и собственные запросы переводчика

21. Как в Python организовать вывод в стандартный поток stderr?

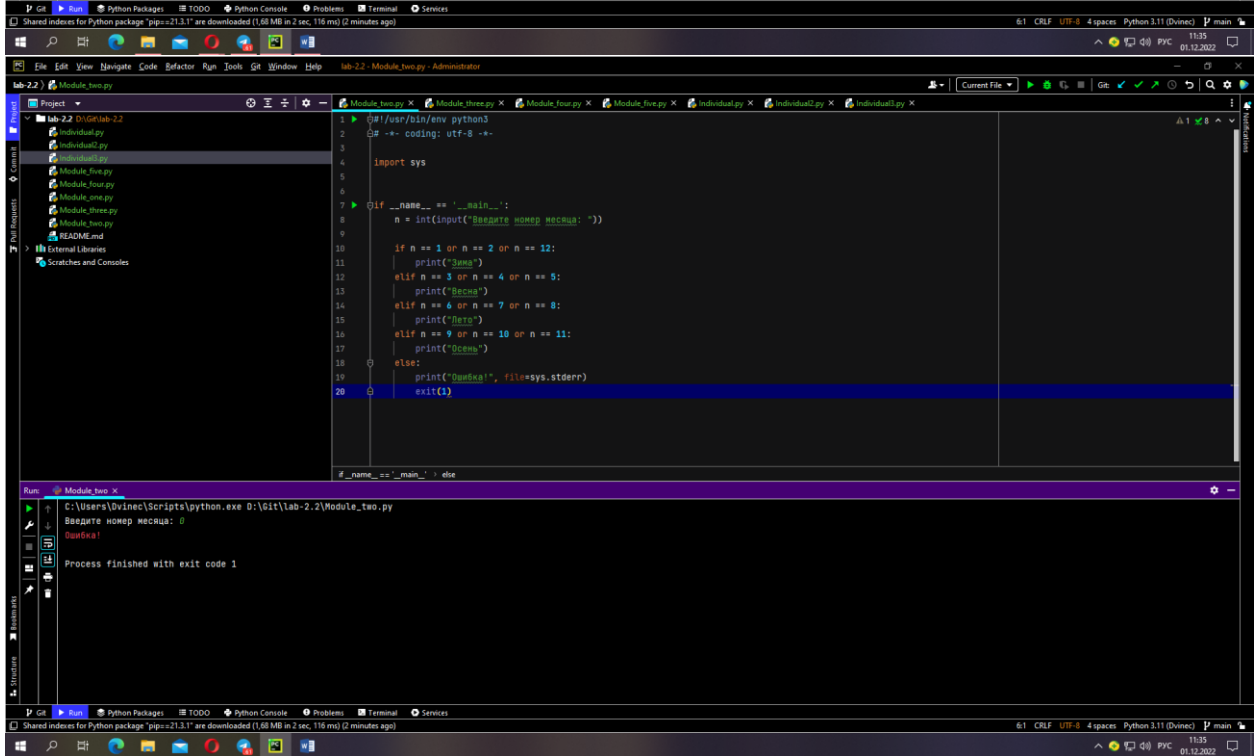
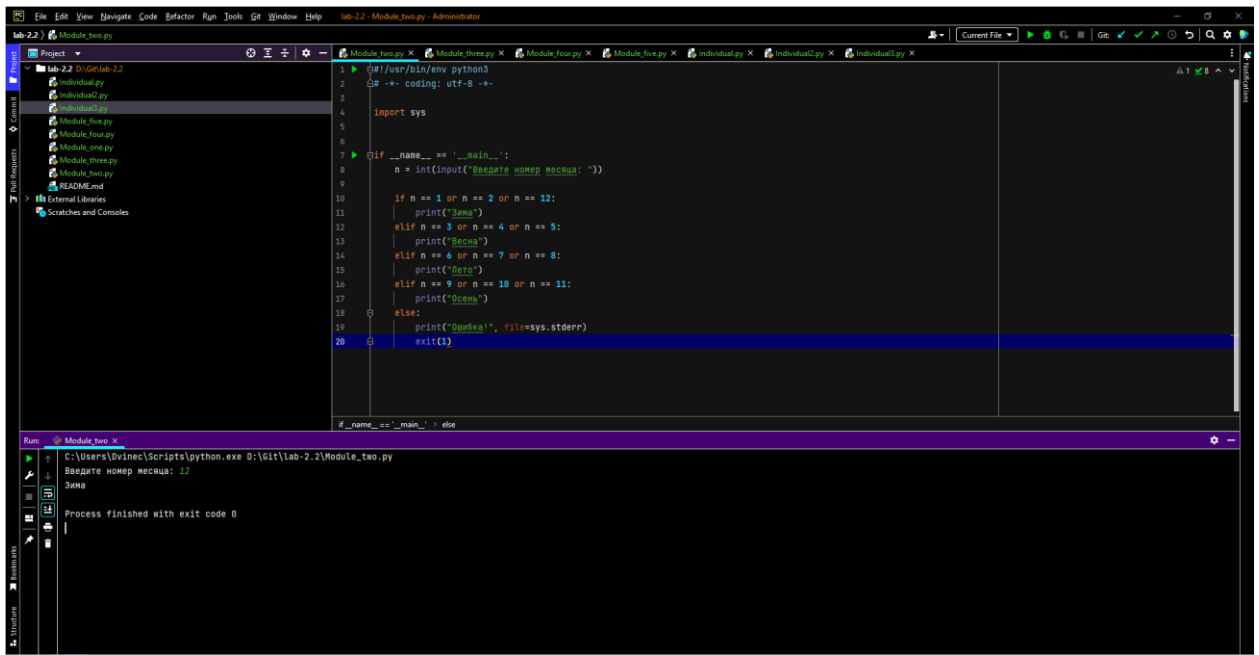
22. Каково назначение функции exit ?

Функция exit() модуля sys - это быстрый способ выйти из программы при возникновении ошибки. Эта функция реализуется путем вызова исключения SystemExit, и, следовательно, выполняются действия по очистке, указанные в предложениях finally операторов try. Так же можно перехватить попытку выхода на внешнем уровне.

Модуль 1



Модуль 2



Модуль 3

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5
6
7 if __name__ == '__main__':
8     n = int(input("value of n? "))
9     x = float(input("value of x? "))
10
11     S = 0.0
12     for k in range(1, n + 1):
13         a = math.log(k * x) / (k * k)
14         S += a
15
16     print(f"S = {S}")
```

Run: Module three X

C:\Users\Divine\Scripts\python.exe D:\git\lab-2.2\Module_three.py

Value of n? 105

Value of x? 23

S = 6.032579057533379

Process finished with exit code 0

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5
6
7 if __name__ == '__main__':
8     n = int(input("value of n? "))
9     x = float(input("value of x? "))
10
11     S = 0.0
12     for k in range(1, n + 1):
13         a = math.log(k * x) / (k * k)
14         S += a
15
16     print(f"S = {S}")
```

Run: Module three X

C:\Users\Divine\Scripts\python.exe D:\git\lab-2.2\Module_three.py

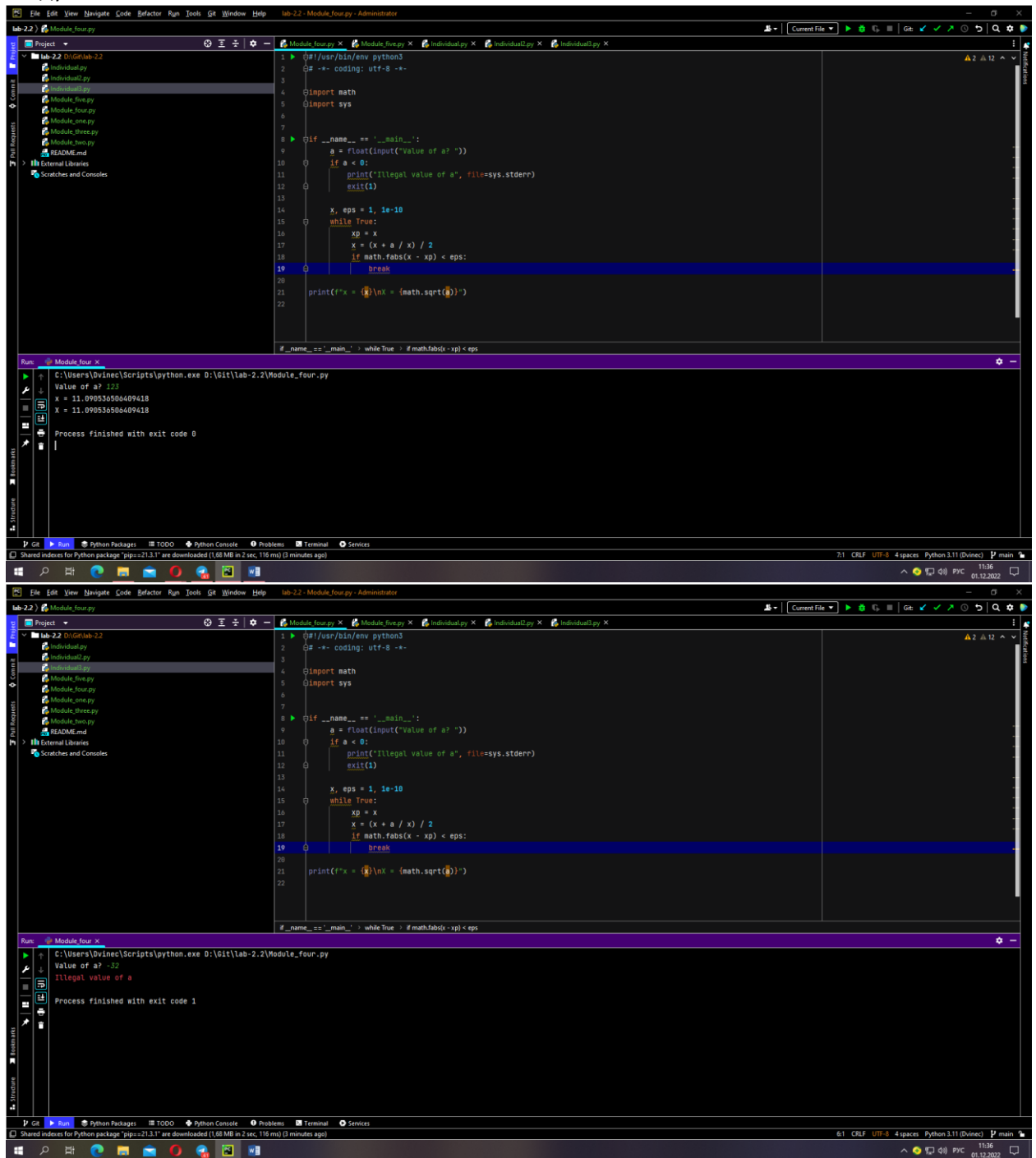
Value of n? 102

Value of x? 21

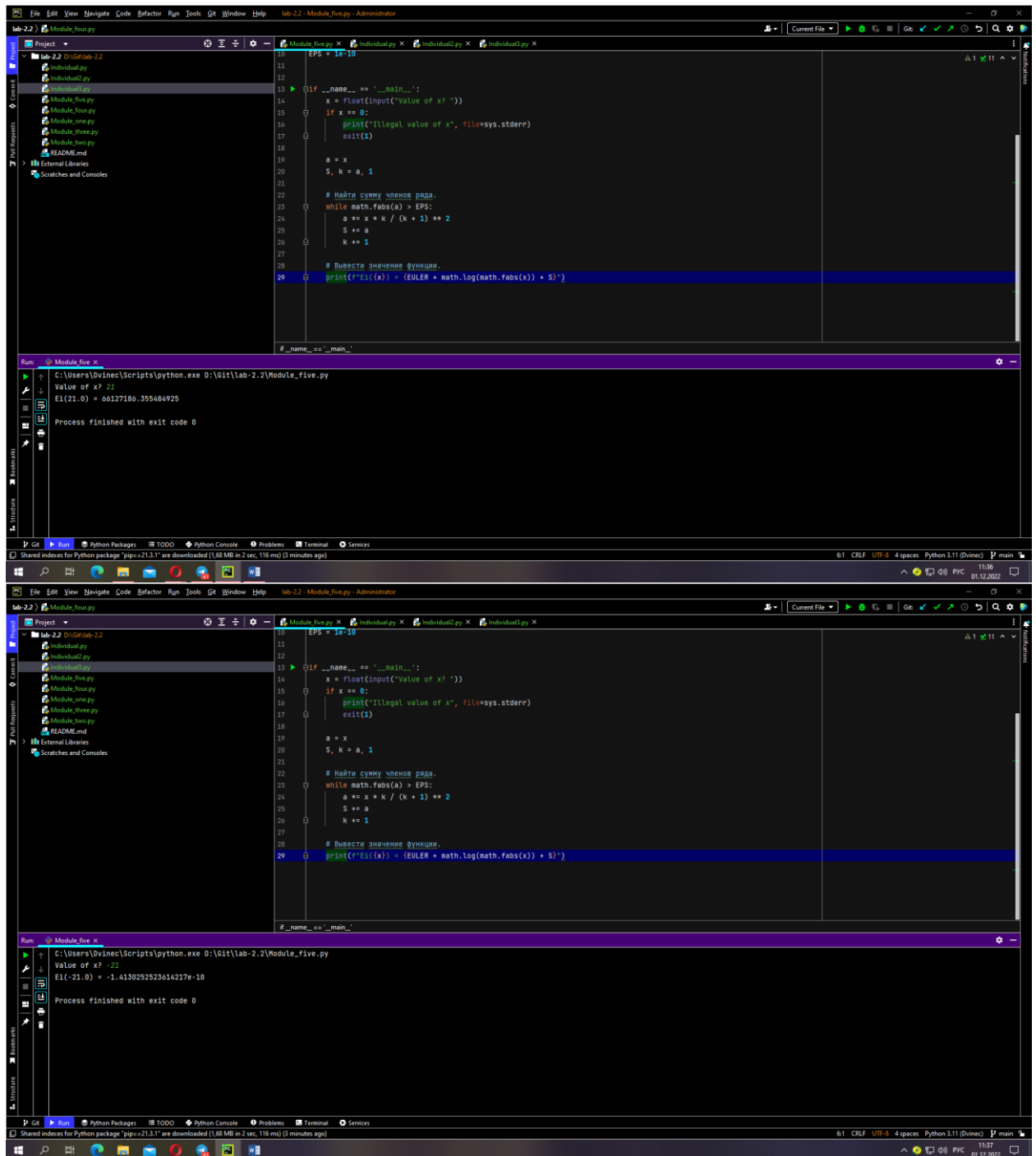
S = 0.0

Process finished with exit code 0

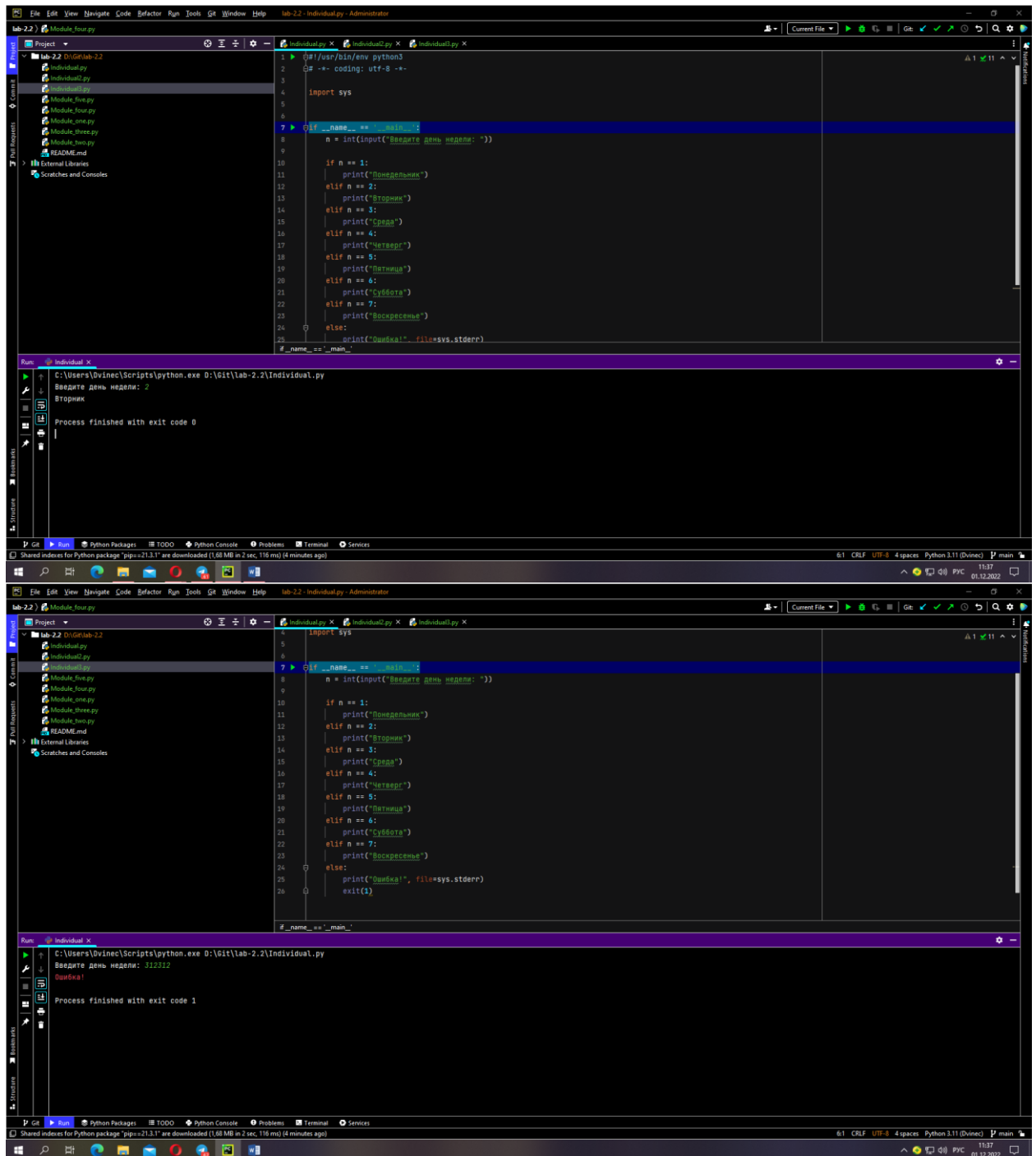
Модуль 4



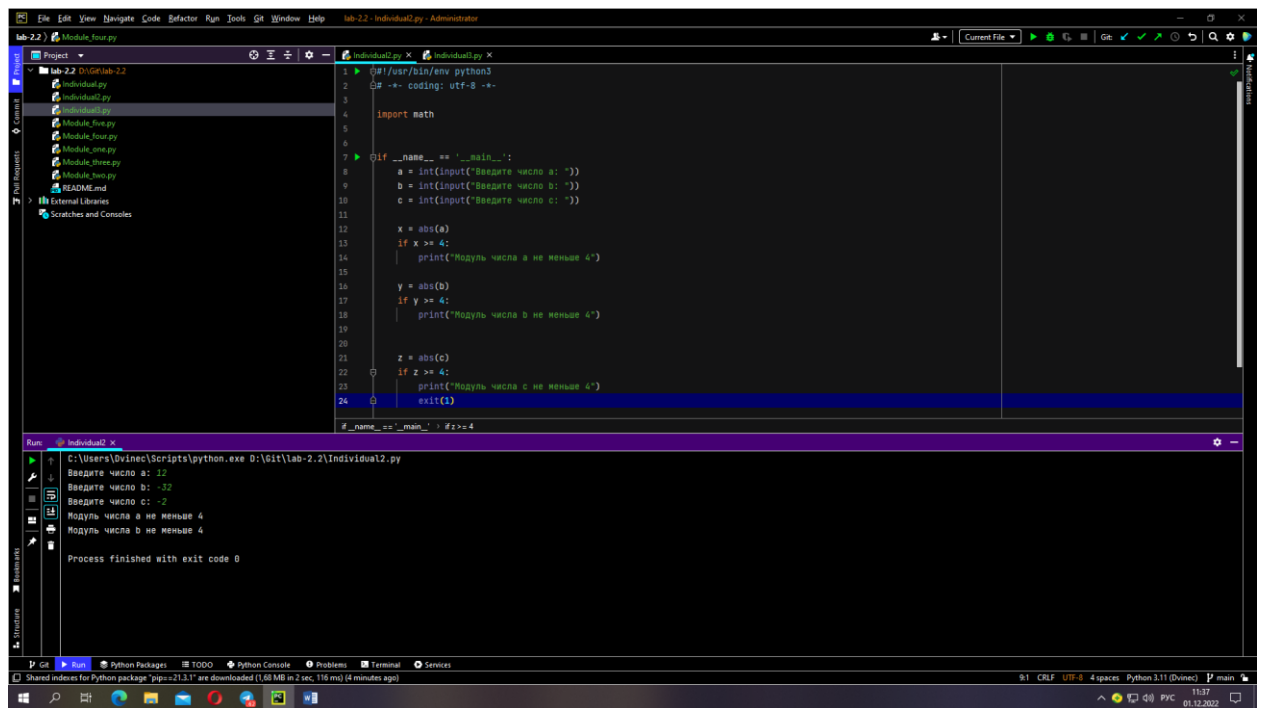
Модуль 5



Индивидуальное задание 1



Индивидуальное задание 2



Индивидуальное задание 3

