Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5**

**дисциплины «Программирование на Python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Лейс Алексей Вячеславович  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики: кандидат тех. наук доцент кафедры инфокоммуникаций: Воронкин Р.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

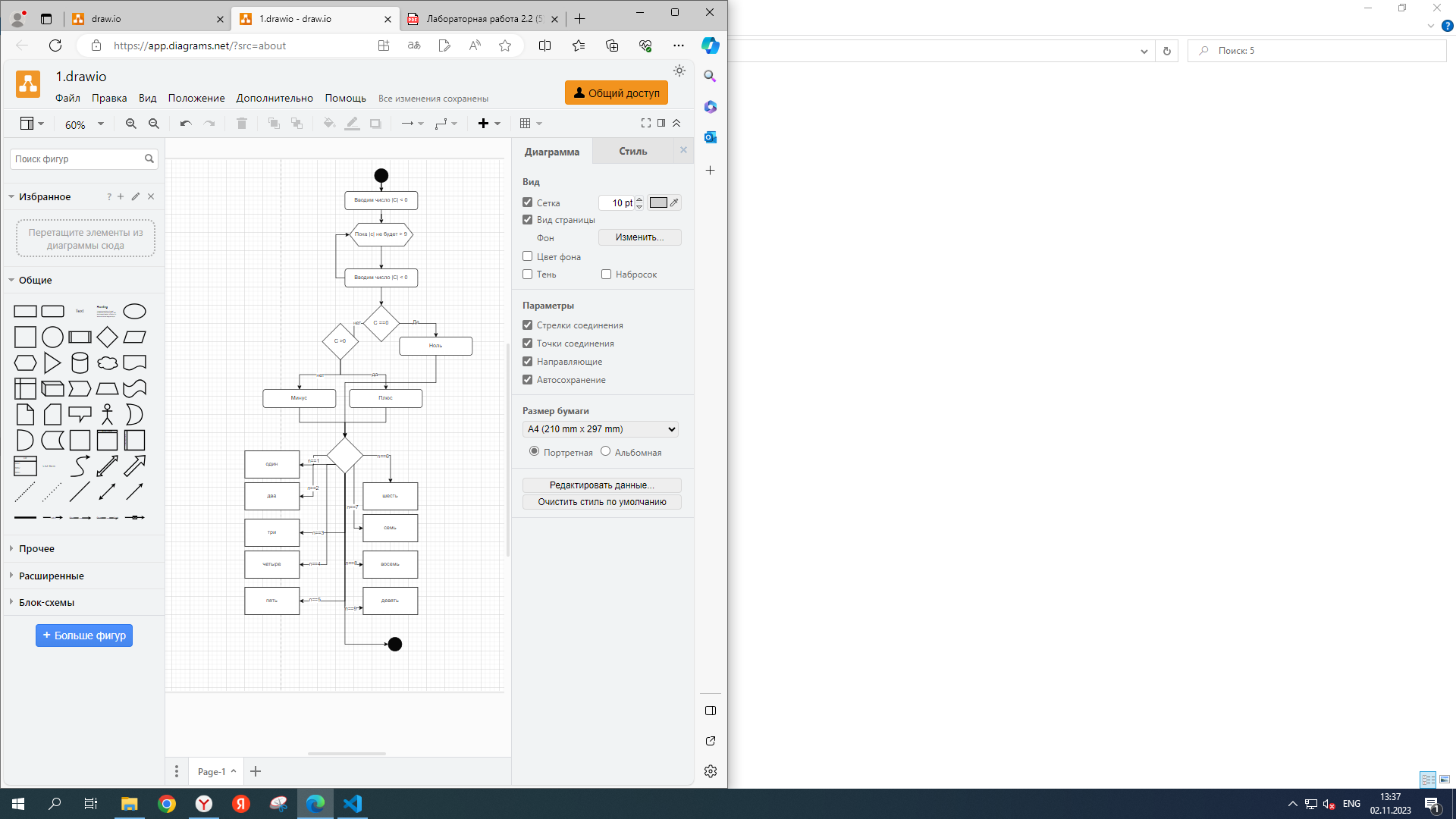
Ставрополь, 2022 г.

**Тема:** Основы языка Python

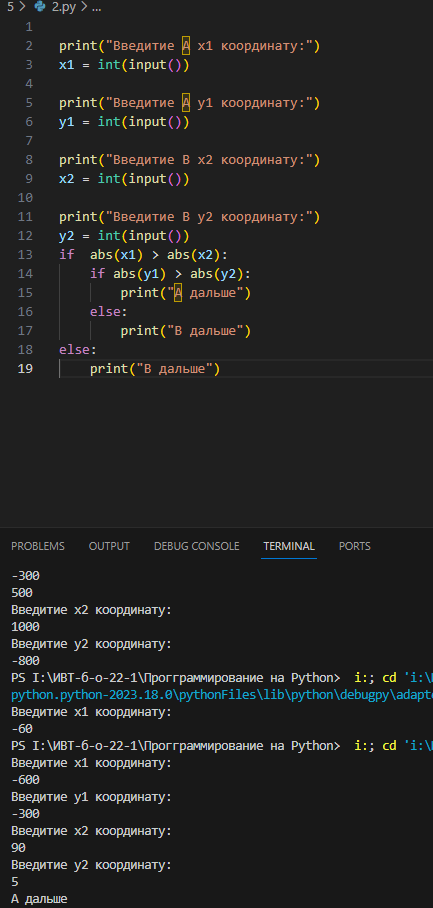
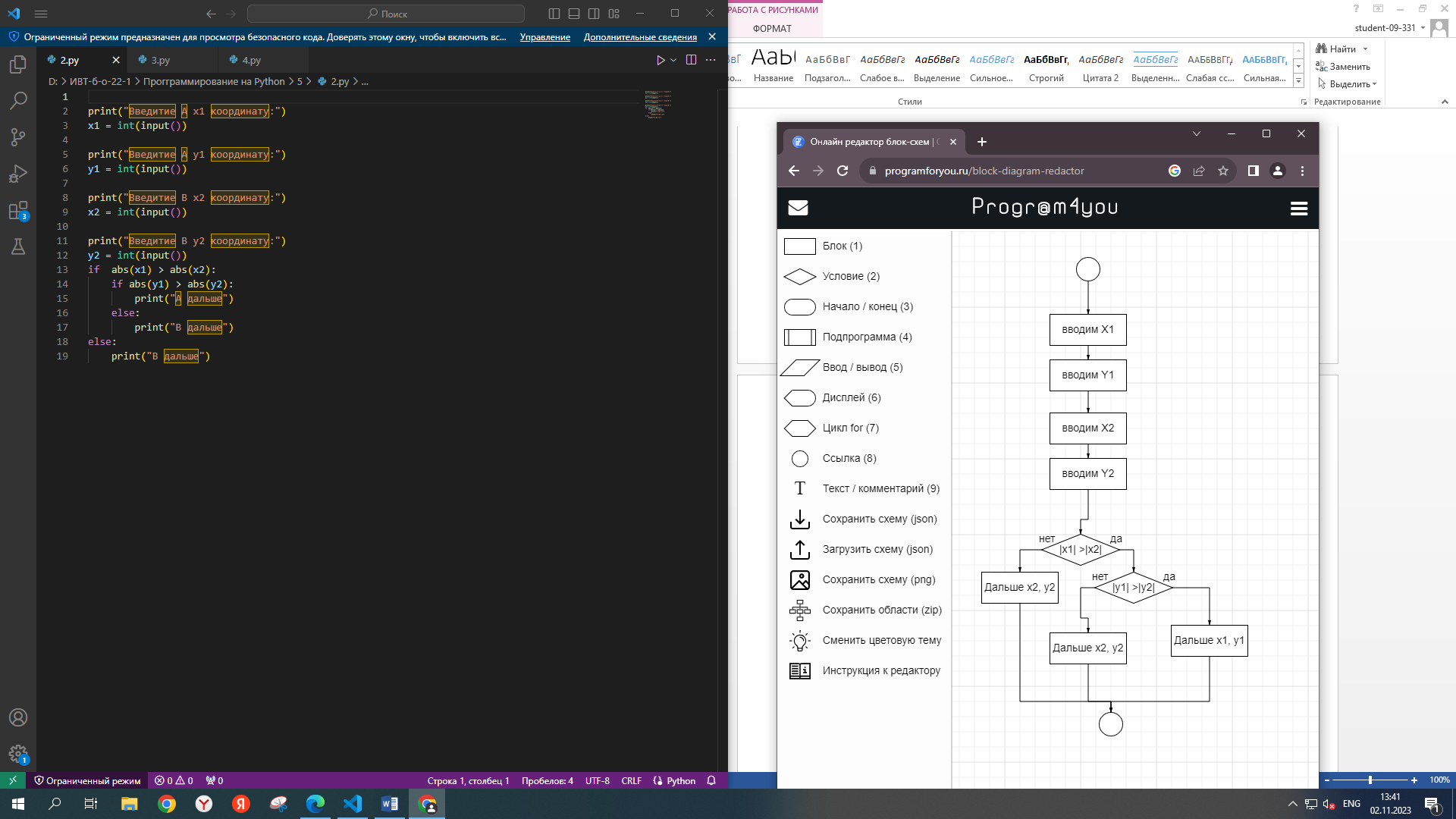
**Цель работы:** исследование процесса установки и базовых возможностей языка Python версии 3.x.

**Порядок выполнения работы:**

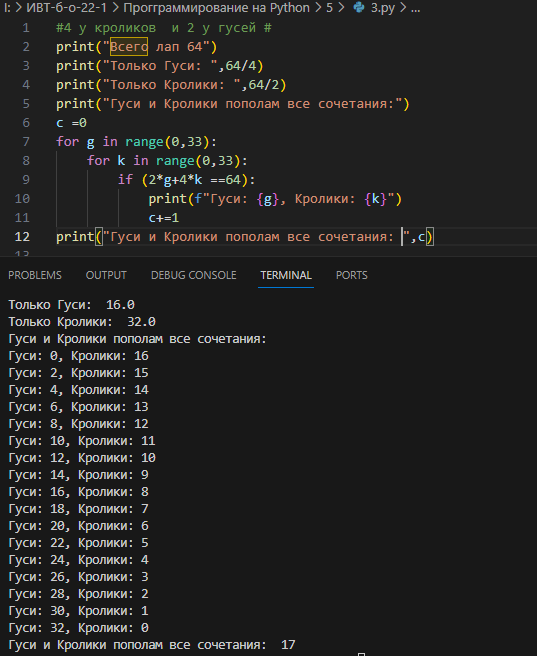
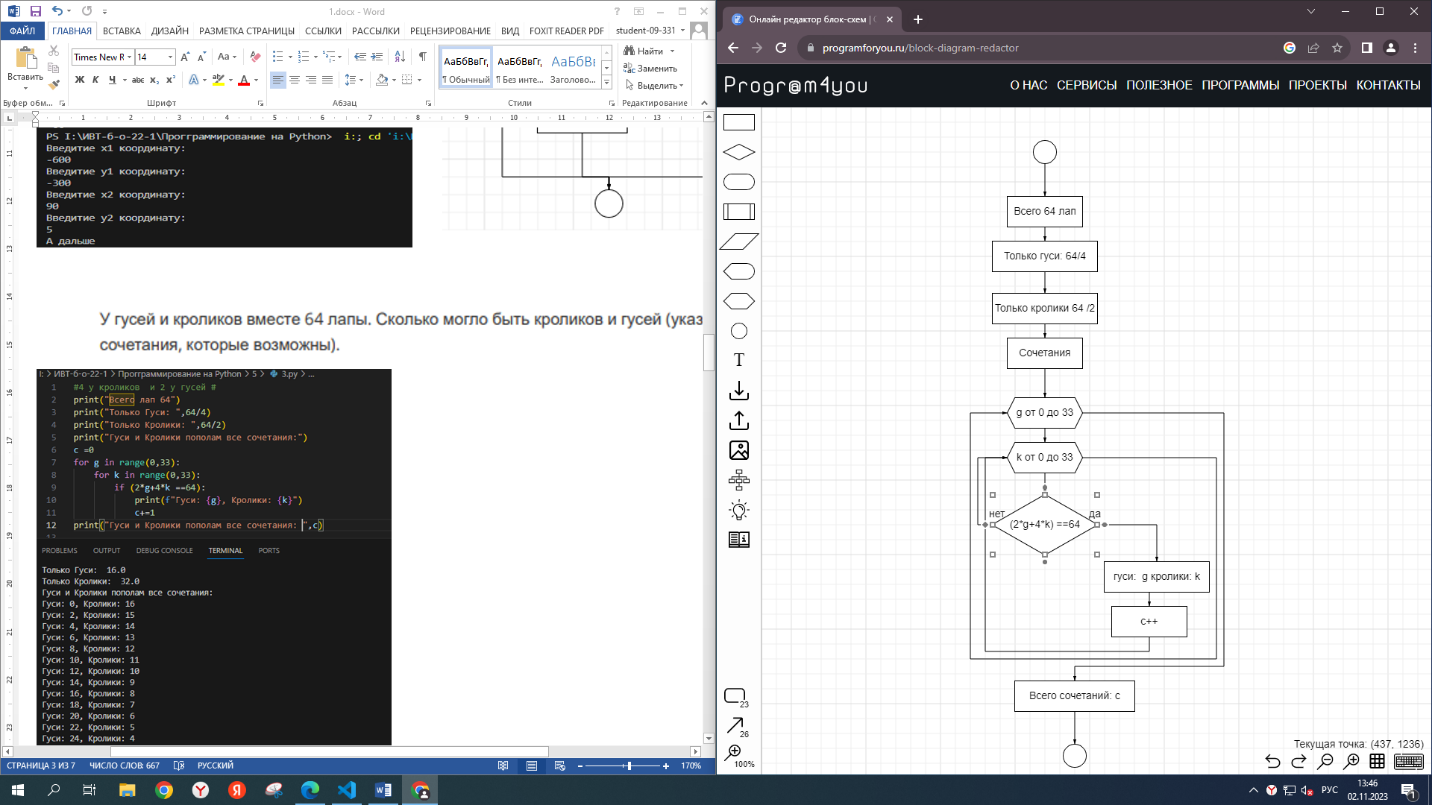


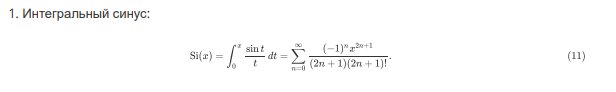


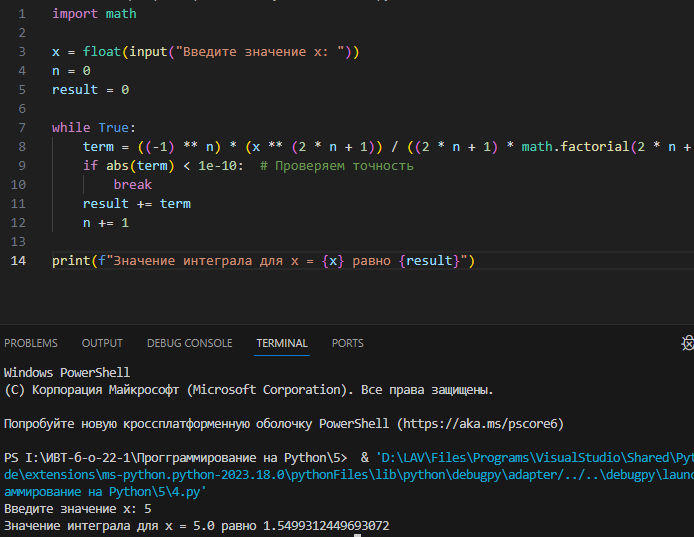


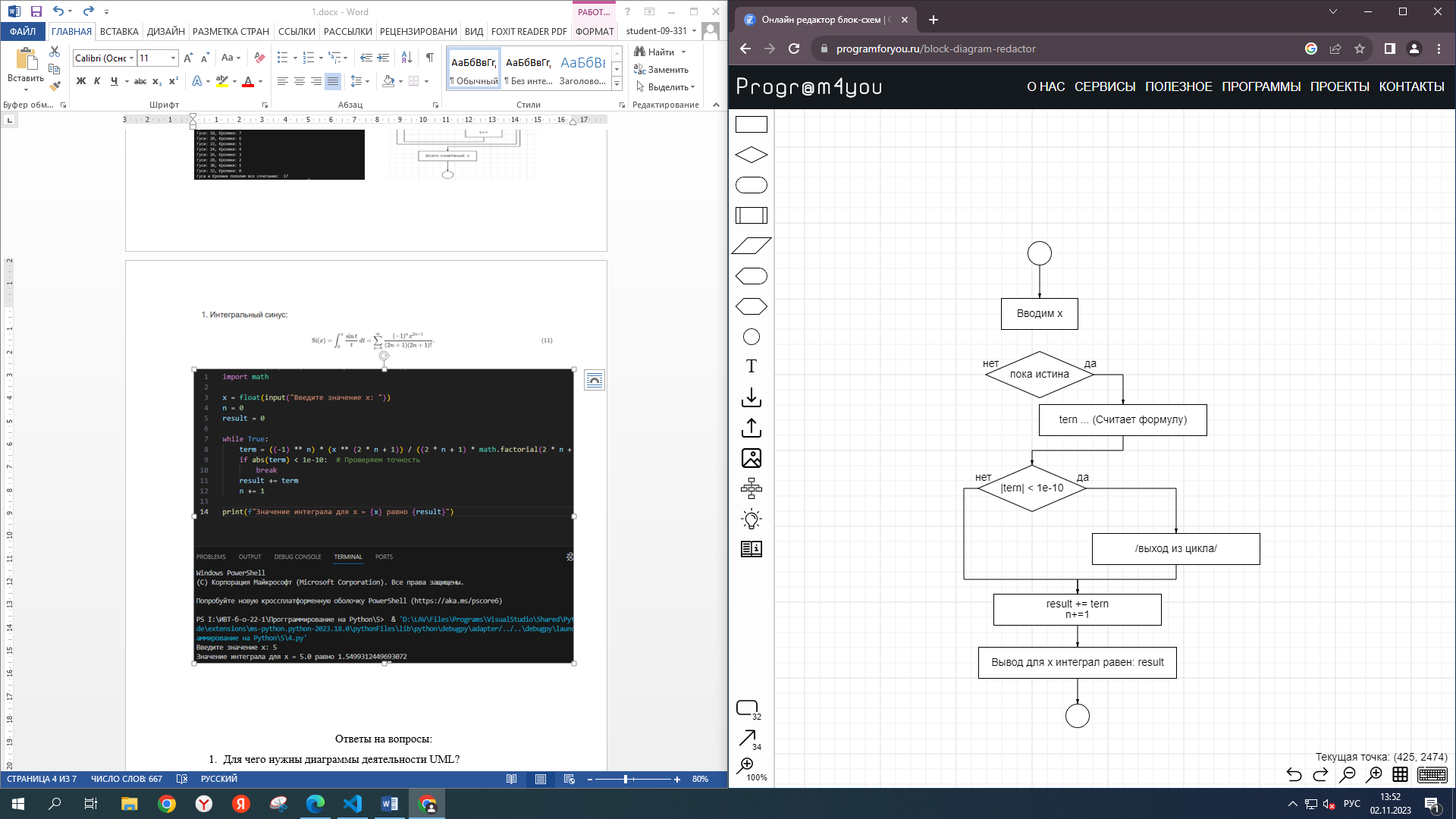












Ответы на вопросы:

1. Для чего нужны диаграммы деятельности UML?
   * Диаграммы деятельности UML используются для визуализации и моделирования деятельности или процессов в системах. Они помогают описать последовательность шагов, роли, решения и потоки управления в системе.
2. Что такое состояние действия и состояние деятельности?
   * Состояние действия (action state) в диаграммах деятельности UML представляет мгновенное выполнение какой-либо операции или действия. Состояние деятельности (activity state) представляет долгосрочное выполнение действий и может содержать вложенные диаграммы деятельности.
3. Какие нотации существуют для обозначения переходов и ветвлений в диаграммах деятельности?
   * Для обозначения переходов используется стрелка с меткой условия, а для ветвлений - ромбовидный символ с условием.
4. Какой алгоритм является алгоритмом разветвляющейся структуры?
   * Алгоритм разветвляющейся структуры представляет собой условное ветвление, такое как оператор if в Python.
5. Чем отличается разветвляющийся алгоритм от линейного?
   * Разветвляющийся алгоритм позволяет программе принимать разные пути выполнения в зависимости от условий, в то время как линейный алгоритм выполняется последовательно без ветвлений.
6. Что такое условный оператор? Какие существуют его формы?
   * Условный оператор позволяет выполнить определенный блок кода при определенных условиях. В Python, основные формы условных операторов - if, elif (else if), и else.
7. Какие операторы сравнения используются в Python?
   * В Python используются следующие операторы сравнения: == (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно).
8. Что называется простым условием? Приведите примеры.
   * Простое условие - это условие, которое содержит только одну проверку. Например: **if x > 5:**.
9. Что такое составное условие? Приведите примеры.
   * Составное условие - это условие, которое содержит несколько проверок, объединенных логическими операторами. Например: **if x > 5 and y < 10:**.
10. Какие логические операторы допускаются при составлении сложных условий?
    * В Python используются логические операторы: and (и), or (или), not (не).
11. Может ли оператор ветвления содержать внутри себя другие ветвления?
    * Да, оператор ветвления может содержать внутри себя другие операторы ветвления, создавая вложенные условия.
12. Какой алгоритм является алгоритмом циклической структуры?
    * Алгоритм циклической структуры позволяет выполнить блок кода многократно. Примером является цикл for или цикл while в Python.
13. Типы циклов в языке Python.
    * В Python существуют два основных типа циклов: цикл for (для перебора элементов) и цикл while (для повторения операций до выполнения условия).
14. Назовите назначение и способы применения функции range.
    * Функция range используется для генерации последовательности чисел, которые могут быть использованы в циклах for. Синтаксис: **range(start, stop, step)**.
15. Как с помощью функции range организовать перебор значений от 15 до 0 с шагом 2?
    * **for i in range(15, -1, -2):**
16. Могут ли быть циклы вложенными?
    * Да, циклы могут быть вложенными, то есть один цикл может находиться внутри другого.
17. Как образуется бесконечный цикл и как выйти из него?
    * Бесконечный цикл образуется, если условие цикла всегда истинно. Чтобы выйти из бесконечного цикла, можно использовать комбинацию клавиш Ctrl+C или применить оператор break.
18. Для чего нужен оператор break?
    * Оператор break используется для прерывания выполнения цикла и выхода из него досрочно, если выполняется определенное условие.
19. Где употребляется оператор continue и для чего он используется?
    * Оператор continue используется внутри цикла для перехода к следующей итерации, игнорируя оставшуюся часть текущей итерации.
20. Для чего нужны стандартные потоки stdout и stderr?
    * Стандартные потоки stdout (стандартный вывод) и stderr (стандартный поток ошибок) используются для вывода информации и ошибок из программы.
21. Как в Python организовать вывод в стандартный поток stderr?
    * Можно использовать sys.stderr, например: **import sys; sys.stderr.write("Ошибка")**.
22. Каково назначение функции exit?
    * Функция **exit** используется для завершения выполнения программы. Она может быть использована для выхода из программы с определенным кодом завершения.

Вывод: в ходе работы был исследован процесс установки и базовых возможностей языка Python версии 3.x.