

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧЕРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРИТЕТ

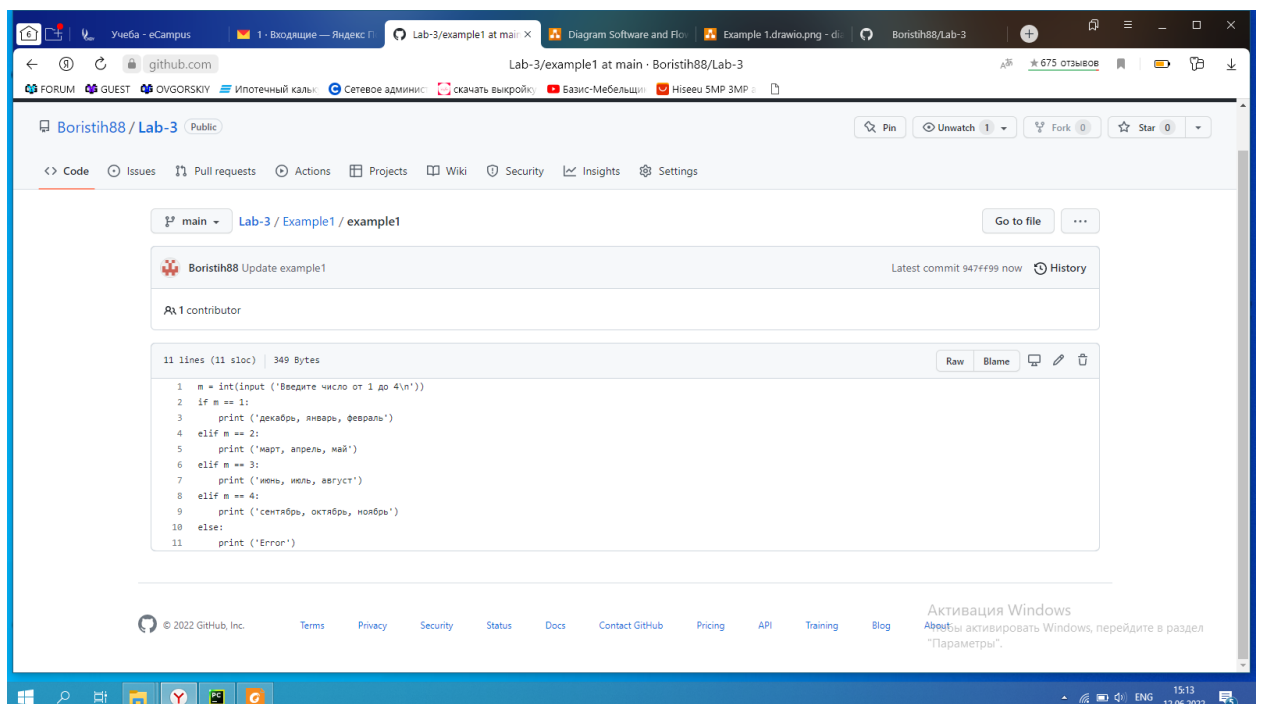
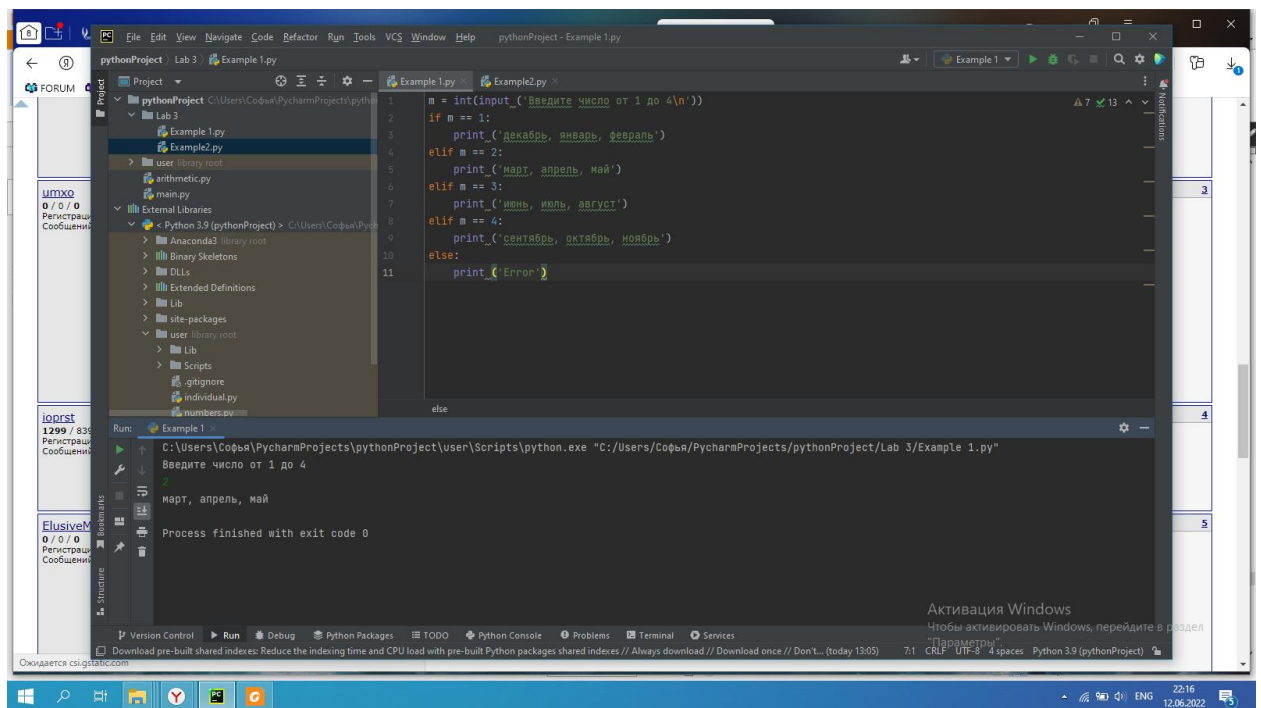
Лабораторная работа № 3.
«Условные операторы и циклы в языке Python»

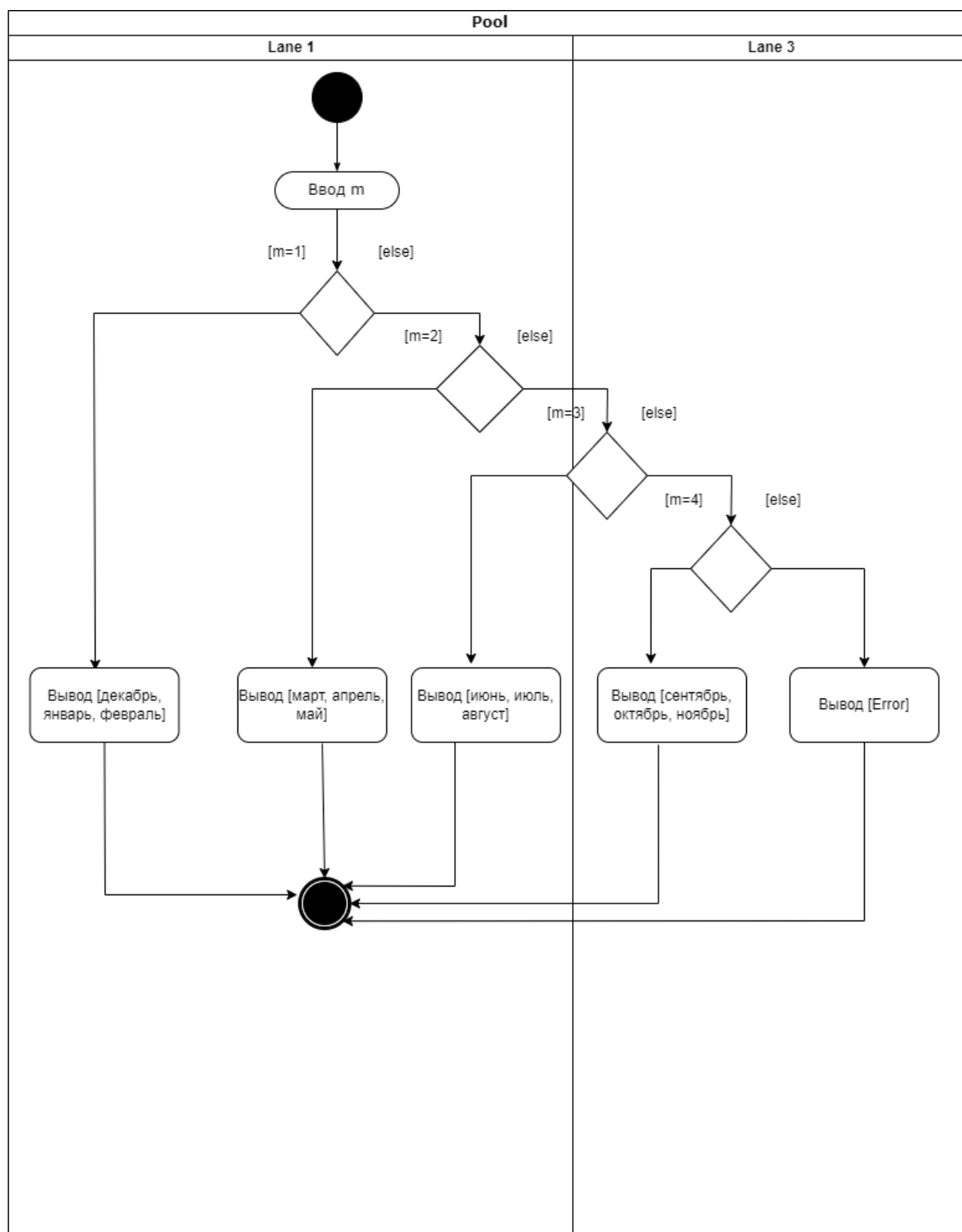
Выполнил
Тихоненко Борис Витальевич
1 курс, группа ИТС-б-з-21-1,
11.03.02
«Инфокоммуникационные
системы и сети», заочная
форма обучения.
«Основы
кроссплатформенного
программирования»
Преподаватель: Воронкин
Роман Александрович

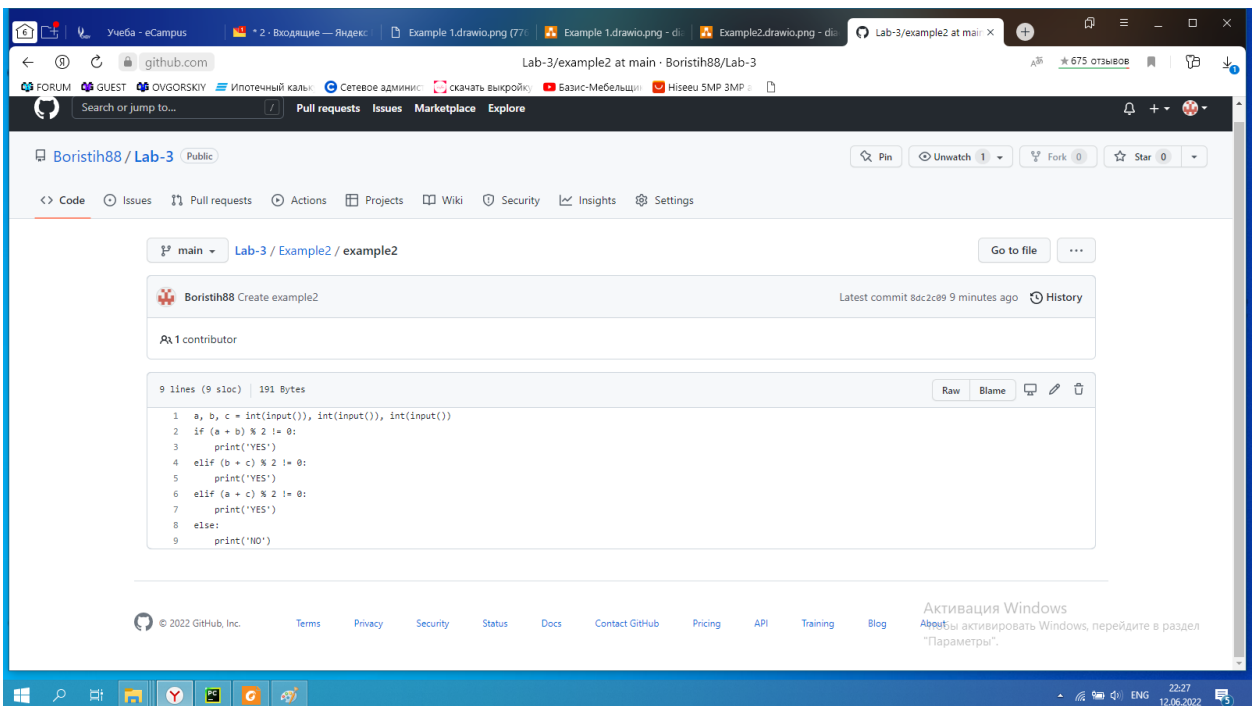
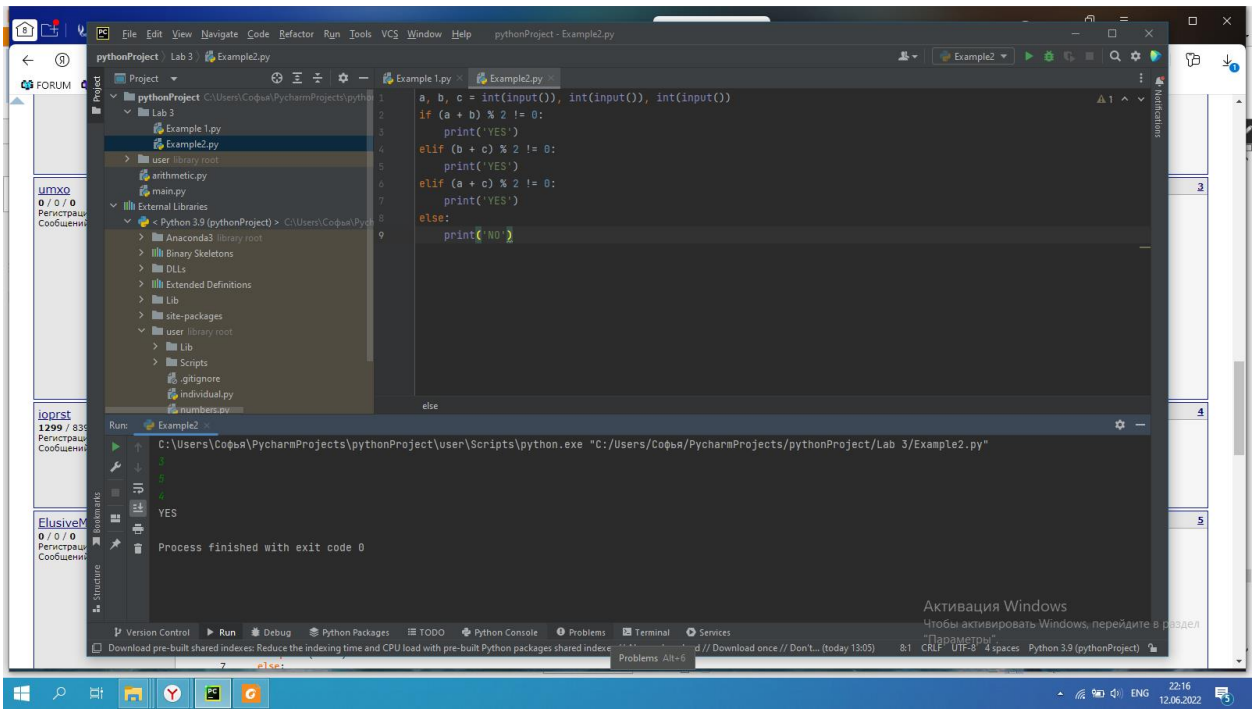
Ставрополь, 2022 г.

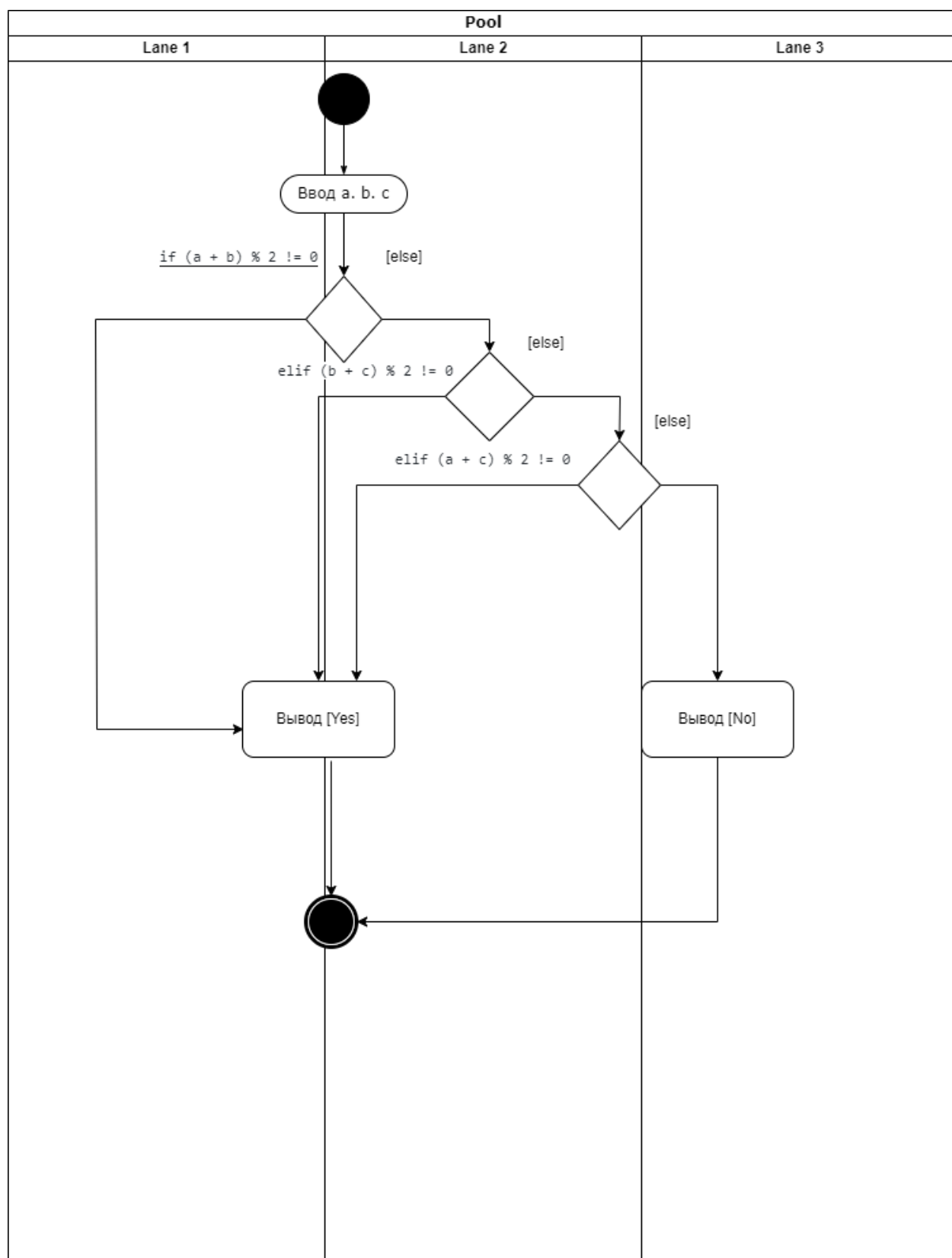
Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

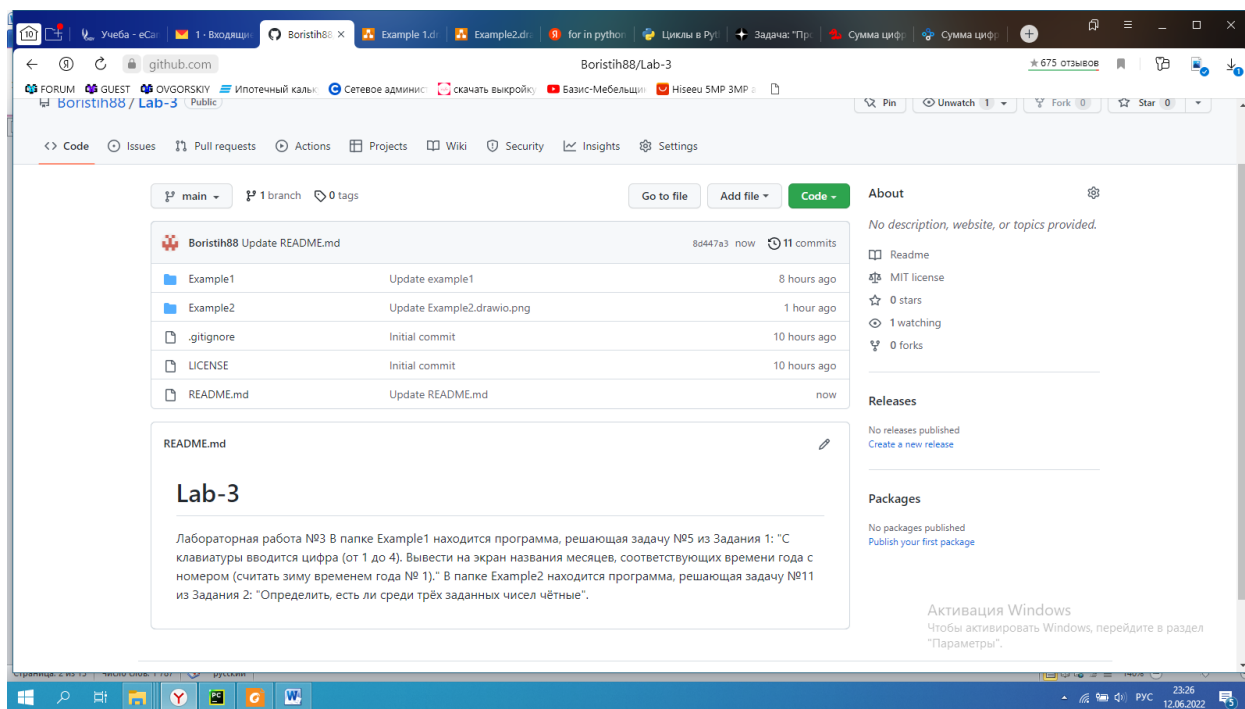
Согласно методике и порядку выполнения работ, были сделаны следующие скриншоты, подтверждающие выполнение лабораторной работы.











Ответы на контрольные вопросы:

1. Для чего нужны диаграммы деятельности UML?

Унифицированный язык моделирования (UML) является стандартным инструментом для создания «чертежей» программного обеспечения. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем.

2. Что такое состояние действия и состояние деятельности?

Выполняемые атомарные вычисления называются состояниями действия, поскольку каждое из них есть состояние системы, представляющее собой выполнение некоторого действия. Состояния действия не могут быть подвергнуты декомпозиции. Кроме того, они атомарны. Это значит, что внутри них могут происходить различные события, но выполняемая в состоянии действия работа не может быть прервана. Обычно предполагается, что длительность одного состояния действия занимает неощутимо малое время. В противоположность этому состояния деятельности могут быть подвергнуты дальнейшей декомпозиции, вследствие чего выполняемую деятельность можно представить с помощью других диаграмм деятельности. Состояния деятельности не являются атомарными, то есть могут быть прерваны. Предполагается, что для их завершения требуется заметное время. Можно считать, что состояние действия - это частный вид состояния деятельности, а конкретнее – такое состояние, которое не может быть подвергнуто дальнейшей декомпозиции. А состояние деятельности можно представлять себе как составное состояние, поток управления которого включает только другие состояния деятельности и действий.

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

В UML переход представляется простой линией со стрелкой. Точка ветвления представляется ромбом.

4. Какой алгоритм является алгоритмом 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.

5. Чем отличается разветвляющийся алгоритм от линейного?

Линейный алгоритм – это алгоритм, в котором операции выполняются последовательно. Разветвляющийся алгоритм – это алгоритм, в котором последовательность выполнения операций зависит от определенных условий.

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

Условный оператор или оператор ветвления - это оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения. Оператор ветвления if и операторы цикла while и for.

7. Какие операторы сравнения используются в Python?

`==` Возвращает True, если два операнда равны, в противном случае – False. `a == b`

`!=` Возвращает True, если два операнда не равны, в противном случае – False. `a != b`

`>` Возвращает True, если левый операнд больше правого, в противном случае – False. `a > b`

`<` Возвращает True, если левый операнд меньше правого, в противном случае – False. `a < b`

`> =` Возвращает True, если левый операнд больше или равен правому операнду, в противном случае – False. `a > b`

`<=` Возвращает True, если левый операнд меньше или равен правому операнду, в противном случае – False. `a < b`

8. Что называется простым условием? Если это выражение истинно, то выполняются инструкции, определяемые данным оператором. Выражение является истинным, если его результатом является число не равное нулю, непустой объект, либо логическое True. Например, при проверке, четное ли число. Если число делится на 2 без остатка, то это логическое «Да».

9. Что такое составное условие?

Если же для того, чтобы выражение считалось истинным, требуется несколько условий, то есть «Если число А больше числа Б и меньше числа В». Вот это и есть составное условие.

10. Какие логические операторы допускаются при составлении сложных условий?

Логические операторы: `and`, `or`. При создании инструкции `if...else...` мы передаем проверяемое условие. Если условие оценивается как `True`, выполняется блок `if`. В случае же его вычисления как `False`, запускается блок `else`. Если условие содержит несколько компонентов, то для их соединения необходимы логические операторы `and` и `or`.

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

Оператор множественного ветвления языка Python позволяет организовать более двух веток выполнения программы без необходимости вложения. Множественное ветвление: `if-elif-else`.

12. Какой алгоритм является алгоритмом 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.

В Python есть два типа циклов: `for` и `while`.

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

Функция `range` возвращает неизменяемую последовательность чисел в виде объекта `range`.

Параметры функции:

`start` - с какого числа начинается последовательность. По умолчанию - 0

`stop` - до какого числа продолжается последовательность чисел. Указанное число не включается в диапазон

`step` - с каким шагом растут числа. По умолчанию 1

Основная задача — выполнение цикла N-раз. С помощью `range()` можно перебрать все значения в последовательности. Или найти сумму первых 10-ти целых чисел. Получить все числа из заданного диапазона, делящиеся на любое нужное.

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

```
for i in range(15, 0, -2):  
    print(i, end=' ')
```

16. Могут ли быть циклы вложенными? Python позволяет также создавать вложенные циклы. Так, сначала программа запустит внешний и в первой его итерации перейдет во вложенный. Затем она снова вернется к началу внешнего и снова вызовет внутренний. Это будет происходить до тех пор, пока последовательность не завершится или не прервется.

17. Как образуется бесконечный цикл и как выйти из него? Бесконечный цикл возникает тогда, когда программа продолжает выполнение одного цикла и не может прекратить выполнять его. Чтобы остановить бесконечный цикл с помощью командной строки, нажмите клавиши Ctrl + C.

18. Для чего нужен оператор break ? Break – это ключевое слово в Python, которое используется для выхода управления программой из цикла. Оператор break разрывает циклы один за другим, т. е. в случае вложенных циклов сначала прерывает внутренний цикл, а затем переходит к внешним циклам.

19. Где употребляется оператор continue и для чего он используется? Оператор continue в Python используется для перевода управления программой в начало цикла. Оператор continue пропускает оставшиеся строки кода внутри цикла и запускает следующую итерацию. В основном он используется для определенного условия внутри цикла, чтобы мы могли пропустить конкретный код для определенного условия.

20. Для чего нужны стандартные потоки stdout и stderr? stdout и stderr соответственно определяют стандартный вывод и стандартный файл ошибок исполняемой программы. Допустимыми значениями являются PIPE, DEVNULL, существующий дескриптор файла (положительное целое число), существующий объект файла и None. PIPE указывает на необходимость создания нового пайп для нижестоящего элемента. DEVNULL указывает, что специальный os.devnull файла будет использоваться. При заданных по умолчанию параметрах None перенаправление не выполняется; дескрипторы файлов нижестоящего элемента наследуются от родительского элемента. Кроме того, stderr может быть STDOUT, что указывает на то, что данные stderr из дочернего процесса должны быть записаны в тот же дескриптор файла, что и для stdout.

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

```
from __future__ import print_function
import sys

def eprint(*args, **kwargs):
    print(*args, file=sys.stderr, **kwargs)
```

22. Каково назначение функции exit ? Выход из Python. Она реализуется путем вызова исключения SystemExit, поэтому выполняются действия по очистке, указанные в предложениях finally операторов try и можно перехватить попытку выхода на внешнем уровне.

Вывод: Выполняя данную лабораторную работу я приобрел навыки программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоил операторы языка Python версии 3.x if , while , for , break и continue , позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.