 Государственное бюджетное профессиональное образовательное учреждение Краснодарского края

«Краснодарский торгово – экономический колледж»

Жоха В. Н.

**Методическая разработка**

**открытого урока**

по теме: «Понятие логических выражений и операций. Синтаксис инструкций if, if-else, if-elif-else»

(ОД.05 Информатика)

Краснодар, 2025

Разработчик: Жоха В. Н. – преподаватель математики и информатики ГБПОУ КК «КТЭК»

Рецензенты:

|  |  |
| --- | --- |
|  |  |
|  |  |

Методическая разработка открытого урока по теме: «Понятие логических выражений и операций. Синтаксис инструкций if, if-else, if-elif-else» в рамках учебного предмета ОУД 05. Информатика предназначена для преподавателей в целях обмена опытом работы в проведении уроков с использованием технологии комбинированного обучения по специальностям:

* Экономика и бухучет (по отраслям);
* Банковское дело;
* Туризм и гостиничное дело;
* ПРОФЕССИОНАЛИТЕТ: Сервис на транспорте;
* ПРОФЕССИОНАЛИТЕТ: Туризм и гостиничное дело;

Утверждено на заседании цикловой методической комиссией ЦМК ПП МО №1

Протокол № \_\_ от «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2025 г.

Председатель ЦМК \_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Пояснительная записка | 4 |
| План урока | 6 |
| Ход урока  Часть 1. Теоретическая  Часть 2. Практическая  Список используемой литературы | 8  9  20  24 |

**Пояснительная записка**

Формирование у студентов профессиональных компетенций является обязательным требованием, в связи с переходом учебных заведений на государственные стандарты нового поколения. Профессиональные компетенции обеспечат будущим специалистам более полную подготовку в своей будущей профессиональной деятельности.

В наше время, когда современные технологии становятся все более важными и влияют на все сферы нашей жизни, умение программировать становится неотъемлемым навыком, весьма востребованным навыком. Программирование – это язык будущего, с помощью которого создаются новые, инновационные и полезные продукты и сервисы для современного общества.

Освоение программирования полезно не только для профессионалов в этой области, но и для каждого человека. Навыки программирования помогают развивать логическое мышление, улучшать критическое мышление и способствуют развитию аналитических навыков.

Еще одно преимущество обучения навыкам программирования – это рост возможностей для карьеры. В настоящее время все больше компаний нуждаются в специалистах, которые могут разрабатывать и поддерживать программное обеспечение, создавать приложения для своей работы при выполнении задач, входящих в круг специалиста. Умение программировать делает такого специалиста конкурентоспособным на рынке труда и открывает двери к новым карьерным возможностям. Ведь в будущем практически все сферы деятельности будут связаны с информационными технологиями.

Таким образом, освоение основ программирования является важным и полезным навыком для каждого человека, умение программировать поможет развивать важные навыки и повышать свою конкурентоспособность на рынке труда. Эти навыки открывают перед молодыми специалистами новые возможности и позволят, в частности, вносить вклад в современное общество с помощью информационных технологий.

Методическая разработка состоит из основных разделов:

* подготовительный этап, в котором планируется порядок проведения урока;
* структура занятия состоит в подробном распределении элементов занятия и изучаемых вопросов, а также методов обучения и времени, необходимого для изучения темы;
* методика проведения занятия включает в себя подробное описание каждого элемента занятия и его цели;
* заключительный этап анализирует результаты работы и подводит итоги проведенного урока.

**План урока**

**Тема:** «Понятие логических выражений и операций. Синтаксис инструкций if, if-else, if-elif-else»

**Методическая цель урока:** использование технологий комплексного обучения для активизации профессиональной деятельности студентов.

**Дидактические цели урока:**

*- образовательные* – сформировать знания об одном популярном языке программирования Python и показать профессиональную значимость полученных знаний;

- *развивающие* – развивать познавательную активность, учебную самостоятельность, профессиональное внимание, умение использовать полученные знания в практической деятельности.

*- воспитательные* – содействовать воспитанию положительных мотивов учебно-познавательной, трудовой деятельности, прививать навыки коллективного труда, интерес к будущей профессии.

**Тип урока: урок** изучения логических операторов языка программирования Python, а именно if…; if…else…; if…elif…else… с использованием интерактивной среды программирования Google Colab, новых понятий программирования.

**Вид учебного занятия:** урок-практическое занятие, с элементами беседы, показа примеров и решения заданий в среде Google Colab.

**Форма организации:** объяснение, решение примеров и выполнение практических примеров.

**Средства обучения:**

-Интерактивная среда программирования Google Colab;

- обучающие конспекты.

**Междисциплинарные связи:** информатика

**Студент должен знать:**

- основные понятия информатики, файл, запись в файл, сохранение, типы данных, синтаксис при построении арифметических операций, синтаксис логических конструкций изучаемого языка программирования.

- обозначения элементов блок-схем

**Студент должен уметь:**

* запускать интерактивную среду программирования Google Colab;
* подключать используемые в коде библиотеки Python, используя директиву import;
* объявлять переменные различных типов;
* строить на языке программирования арифметические выражения с использованием арифметических операторов: +;-; \*; /; //; %.
* Вычислять значения арифметических выражений по заданным в коде частным числовым значениям;
* Строить алгебраические выражения с арифметическими операторами без использования библиотеки math;
* Строить алгебраические выражения с использованием простейших функций из библиотеки math;
* Строить алгебраические выражения, в которых значения переменных вводятся из среды программирования;
* Знать код стандартного ввода данных в Python;
* Оформлять написанный код в соответствии с требованиями внутреннего кодекса PEP-8.

**Ход урока**

1. *Организационный момент (2 мин):*

* проверка присутствующих, готовности студентов к уроку;
* сообщение темы и цели урока;

1. *Актуализация знаний, обучающихся методом фронтального опроса (3 мин):*
2. Что такое программирование?
3. Назовите несколько языков программирования!
4. Что такое программа?
5. Что такое алгоритм?
6. Какие виды алгоритмов вы знаете?
7. Что такое данные и чем они отличаются от значений вычисленных данных?
8. *Мотивация учебной деятельности (5 мин)*

Уважаемые студенты!

Вы на этом занятии познакомитесь с выполнением логических выражений и операций на языке программирования Python (Питон). Использовать будем интерактивную среду программирования Google Colab.

Вы познакомитесь с принципами построения логических выражений, используя блок-схемы, и синтаксисом логических выражений на достаточном количестве примеров, чтобы освоить эти полезные навыки в программировании. А также выполним небольшие задания в практической части занятия.

**Тема: Понятие логических выражений и операций. Синтаксис инструкций if…, if…else, if…elif…else**

Цель: ознакомить учащихся с синтаксисом языка Python при выполнении логических условий if… if…else, if…elif…else

Ход работы

**Часть 1. (35 мин)**

**Теоретическая часть**

Синтаксис инструкций if, if-else, if-elif-else

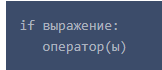
Все ранее рассматриваемые программы на прошлом занятии имели линейную структуру: все инструкции выполнялись последовательно одна за одной, каждая записанная инструкция обязательно выполнялась.

По-другому работает программа, в которую входит, хотя бы один раз, условный оператор if. В такой программе реализуется лишь та ветка, которая отвечает конкретному условию, записанному в условие.

Оператор **if** – условие или логическое выражение, с помощью которого сравниваются данные, и решение принимается на основе результата сравнения.

В Python оператор if похож на другие языки.

Синтаксис оператора if:



Если логическое выражение имеет значение ИСТИНА (TRUE), то выполняется блок операторов внутри оператора if.

Если логическое выражение оценивается как ЛОЖЬ (FALSE), то выполняется первый набор кода после конца оператора(ов) if.

Блок-схема выполнения программы с условным оператором if:

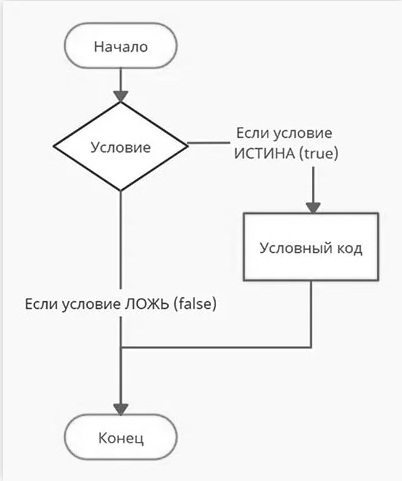


Рис. 1. Блок-схема оператора if

Пример выполнения условного перехода.

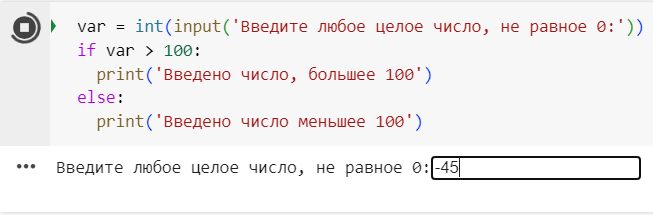


Рис. 2. Введено число, меньшее 100

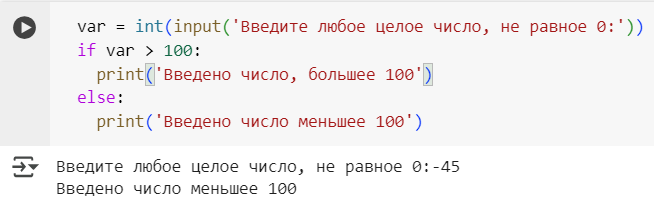


Рис. 3. Результат выполнения: программа сообщила вывод о числе.

Примечание. Для выделения блока инструкций, относящихся к инструкции if или else в языке Питон используются отступы.

Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов в начале строки. Рекомендуется использовать отступ в 4 пробела.

Структуру программы на Python можно изобразить следующим образом:



Рис. 4. Структура программы

Основная ветка программы выполняется всегда, а вложенный код лишь тогда, когда в серой строчке, обозначающей заголовок условного оператора, случается истина.

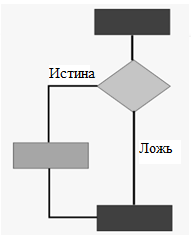


Рис. 5. Блок-схема оператора ветвления

Условный оператор может включать не одну ветку, а две, реализуя тем самым полноценное ветвление:

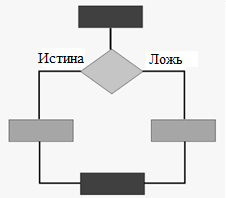


Рис. 6. Блок-схема оператора ветвления с двумя ветками

**Инструкция if … (неполная инструкция)**

Синтаксис конструкции:

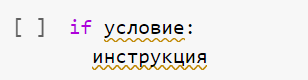


Рис. 4. Инструкция if

Пример 1.

В примере при заданном s условие не выполнено, и программа ничего не выполняет:

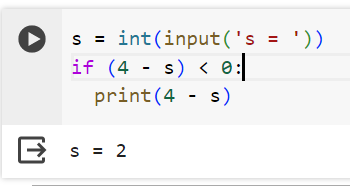


Рис. 7. Условие не выполнено, программа ничего не выполняет

В следующем примере условие для введенного s выполнено, и программа выполняет инструкцию:

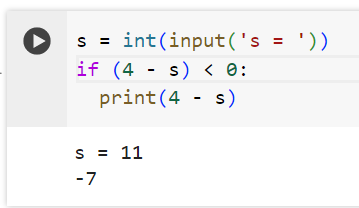


Рис. 7. Условие выполнено, программа сообщает о реализации оператора if.

**Инструкция if … else…**

Синтаксис конструкции:

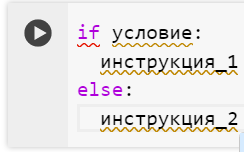
****

Рис. 8. Синтаксис операцииif … else…

Пример 2.

Найти наибольшее число среди двух введенных целых чисел стандартным образом:

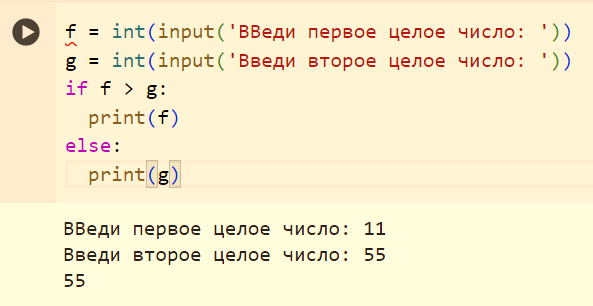


Рис. 10. Код программы, в котором реализована блок-схема оператора if…else…

Пример 2:

Найти разность введенных чисел, если первое число больше второго, и сумму, если первое число меньше второго числа.

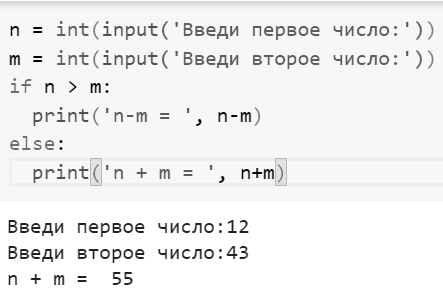


Рис. 11. Код программы, в котором реализована блок-схема оператора if…else… , когда выполнено условие n < m.

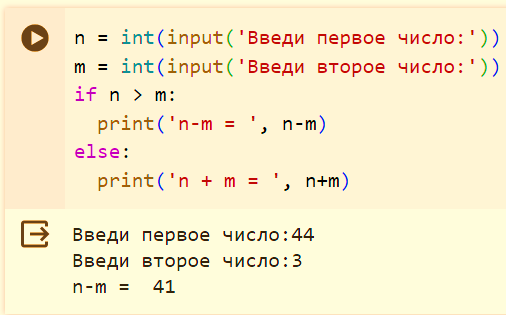


Рис. 12. Код программы, в котором реализована блок-схема оператора if…else…, когда выполнено условие n > m.

Пример 3.

Программа, при вводе года, определяет, является ли год високосным и выводит сообщение. Ввод года реализован с помощью стандартного ввода

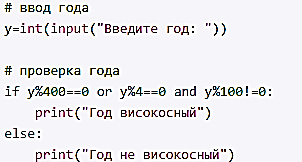


Рис. 13. Код программы, в котором реализована блок-схема оператора if…else…

**Инструкция if … elif: … else: …**

Условный оператор **if-elif-else** в Python — это способ написать программный код так, чтобы он выдавал результат в зависимости от того, выполняется определенное условие или нет.

Когда есть несколько условий, можно использовать **elif** (сокращение от **else if** — «иначе если»), чтобы проверять их поочередно. Если ни одно из условий не истинно, используют блок **else**, чтобы выполнить код по умолчанию.

Синтаксис выглядит так:

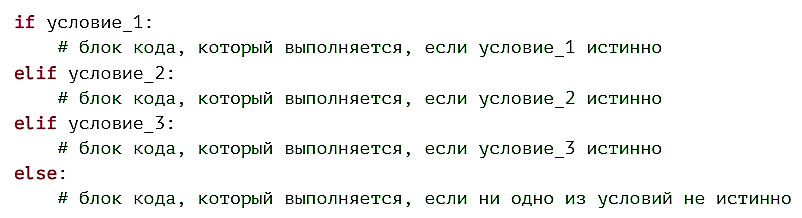


Рис. 14. Синтаксис инструкции if…elif…else…

Блок-схема инструкции:

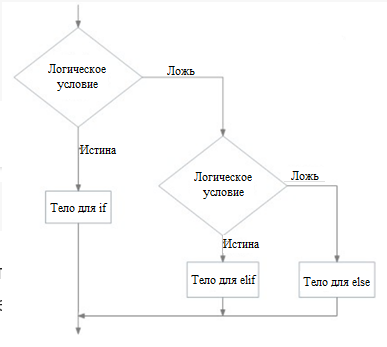


Рис. 15. Блок-схема оператора if … elif …else …

Пример 4. Алгоритм, определяющий вид треугольника по введенным стандартным вводом длинам обоих катетов и гипотенузы:

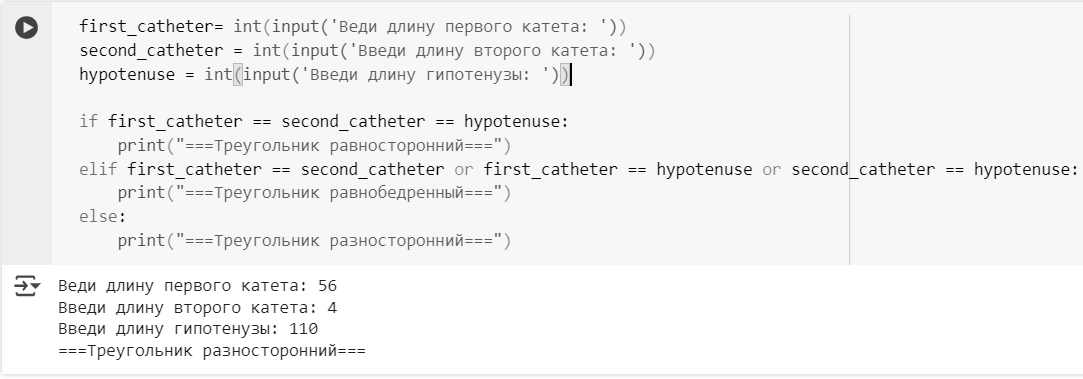


Рис. 16. Реализация алгоритма с оператором if … elif …else …

Пример 5.

Фрагмент кода, в котором реализован возврат оценки за работу студента.

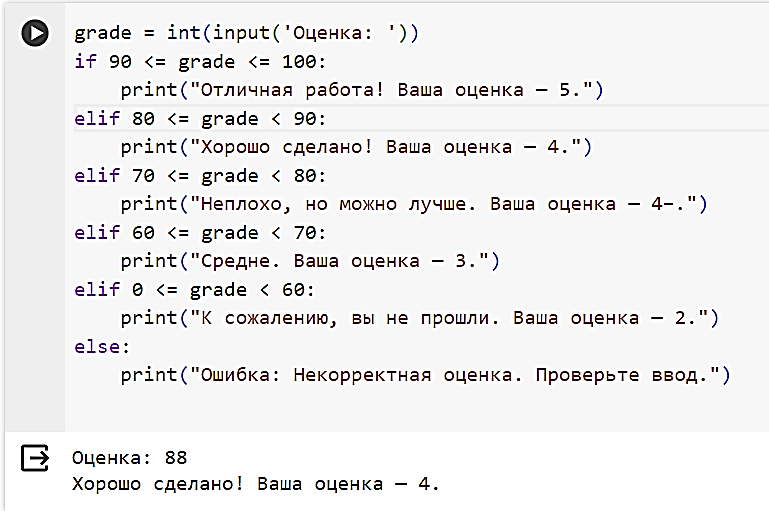


Рис. 17. Реализация алгоритма с оператором if … elif …else …

**Использование логических функций в логических условиях**

Внутри одного логического условия можно выполнять несколько сравнений, что расширяет возможности использования условных конструкций.

Это можно реализовать с помощью логических операторов and, not и or:

* **and** — означает «И» для двух условий. Возвращает **tru**e, если оба условия истинны, и **false** — если ложны;
* **or** — означает «ИЛИ» для двух условий. Возвращает **true**, если хотя бы одно из условий истинно, и **false** в противном случае;
* **not** — означает «НЕ» для одного условия. Возвращает **true**, если условие ложно, и **false** — если истинно.

Эти операции используют, чтобы строить сложные логические выражения.

Логическим операциям присваивается приоритет при выполнении.

Приоритеты операций:

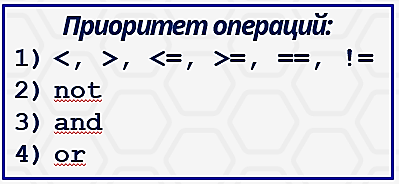


Рис. 18. Приоритеты операций.

**Примеры использования операторов and (логическое И), or (логическое** **ИЛИ), оператора not (логическое отрицание)**

Пример 5. Пользователь вводит два числа t и r. Программа проверяет выполнение условия:



и выводит результат проверки: если условие выполнено, программа возвращает результат проверки в виде логического значения (True, False).

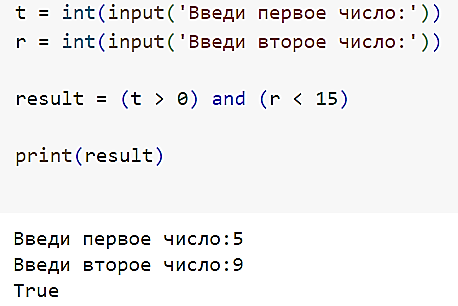


Рис. 19. Реализация условия с логическим and (логическое И)

Пример 6. (использование оператора or (логическое ИЛИ)). Пользователь вводит два числа. Программа проверяет условия: если одно из чисел больше 10, то сообщить «Сумма чисел меньше 20 и равна f+g; иначе если одно из введенных чисел меньшие 10, то «Сумма чисел меньше 20 и равна: f+g.

Алгоритм выглядит следующим образом:

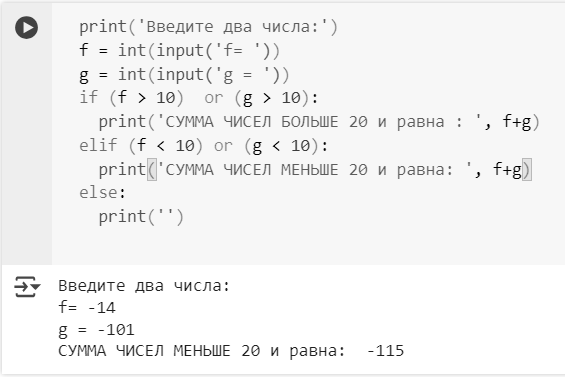


Рис. 20. Реализация алгоритма с условием or (логическое ИЛИ). Введенные числа оба больше 10

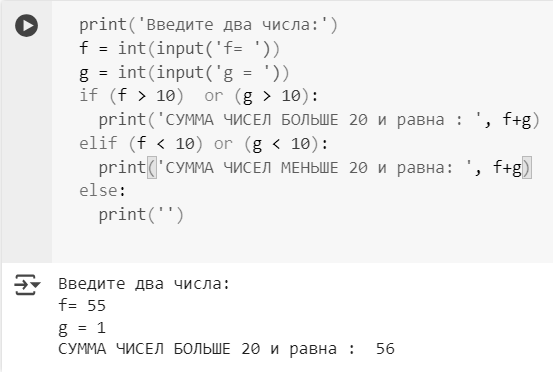


Рис. 21. Оба введенных числа меньше 10

Пример 7. (использование оператора not (логическое отрицание НЕ)):

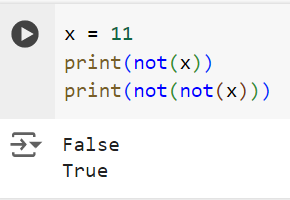


Рис. 22. Отрицание. Отрицание отрицания.

**Часть 2. (35 мин)**

**Практическая часть**

**Задания для работы на ПК в классе в среде Google Colab**

Задание 1.

Напишите программу, в которую пользователь вводит два числа, а программа выводит сообщение, какое из чисел наибольшее. Реализуйте алгоритм в соответствии с блок-схемой:

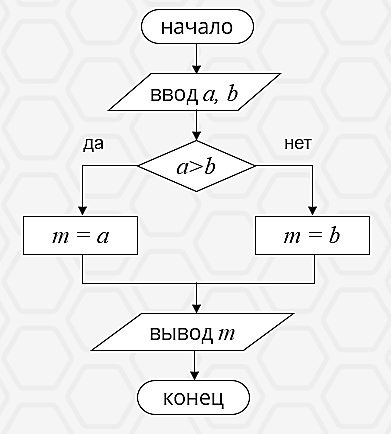


Рис. 17. Блок-схема нахождения наибольшего числа из двух введенных чисел (один из вариантов)

Задание 2.

Ввести стандартным вводом/выводом два числа w\_a и w\_b и реализовать в программе, используя оператор **if … elif … else** … следующий алгоритм:

* если w\_a> 0 и w\_b> 0, тогда вычислить квадрат их суммы w\_a и w\_b;
* иначе если w\_a <0 и w\_b <0 вычислить квадрат их разницы;
* иначе если w\_a> 0 и w\_b <0 вычислить их произведение;
* иначе если w\_a <0 и w\_b> 0 вычислить их частное;
* иначе вывести на экран сообщение "Оба числа равны нулю!"

отобразив результат в обоих случаях на экране с помощью функции print().

Задание 3.

Ввести стандартным вводом/выводом два символа: v\_a и v\_b, равные либо 0 либо 1 и реализовать в программе, используя оператор **if … elif … else** … следующий алгоритм:

* Если введены 1 и 1, то выполнить 5v\_a + 5v\_b;
* Если введены 0 и 0, то выполнить операцию 4v\_a + 4v\_b;
* Если введены 1 и 0, выполнить операцию 4v\_b + 3v\_a;
* Если введены 0 и 1, выполнить операцию 5v\_b + 5v\_a
* Если введены другие символы сообщить "Введены не подходящие символы!"

Задание 4.

Число делится на 6 только в случае соблюдения двух условий:

* последняя его цифра четная;
* сумма всех цифр делится на 3.

Напишите программу, которая возвращает «Число Х делится на 6» или «Число Х неделимо на 6» в зависимости от того, можно ли его разделить на 6. В качестве аргумента может быть передано любое целое число (использовать стандартный ввод в Python.

Задание 5.

Вычислите значения алгебраических выражений для различных значений переменных:

* если :
* если :
* если
* если :
* если
* Иначе, если введенные числа нарушают условия, сообщить, используя функцию - print(«Введены не подходящие под условия задачи числа!»)

*Закрепление полученных знаний (5 мин):* *ответьте на вопросы*:

1. Кратко охарактеризуйте среду программирования Google Colab?
2. Перечислите логические операции в Python.
3. Кратко охарактеризуйте алгоритмы, содержащие логические выражения.
4. В чем состоит отличие операции if … else… от операции if…elif…else…?

*Итоговая часть занятия (2 мин):*

Выставление и комментирование оценок

*Домашнее задание (3 мин):*

1. Построить алгоритм делимости введенного числа на 15 и реализуйте на языке Python.
2. Постройте алгоритм решения квадратного уравнения при вводе 3 ненулевых коэффициента уравнения и реализуйте его в среде Python

**Использованная литература:**

1. М.С. Цветкова, И. Ю. Хлобыстова «Информатика», Учебник, 2025
2. «Изучаем Python», Марк Лутц.
3. «Программируем на Python», Майкл Доусон
4. 3.«Изучаем программирование на Python», Пол Бэрри
5. «Легкий способ выучить Python», Зед А. Шоу

**Интернет –ресурсы:**

<https://colab.research.google.com/>

<https://edu.vsu.ru/mod/url/view.php?id=200520>

<https://blog.skillfactory.ru/chto-takoe-google-colaboratory-i-komu-on-nuzhen/>

<https://gb.ru/blog/rabota-v-google-colab/>

<https://rutube.ru/video/abc2f3047d5ed7a180646ac141324f4f/>

<https://sky.pro/wiki/python/chtenie-i-obrabotka-dannykh-iz-google-drive-v-colab/>

<https://younglinux.info/python/if>

<https://sky.pro/media/uproshhaem-uslovnye-vyrazheniya-v-python-if-then-else-v-odnu-stroku/>