Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Академия управления городской средой, градостроительства и печати»

УТВЕРЖДАЮ
Заместитель директора
по учебно-методической работе
О.В.Фомичева
«26» декабря 2025 г.

Методические рекомендации по организации и проведению практических занятий

ОП.10 «ОСНОВЫ АЛГОМИРИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Форма обучения -очная

Санкт-Петербург
2025

Разработчик: Ипатова С.В./Оболенская Е.Г., методисты СПб ГБПОУ АУГСГиП

Одобрены на заседании цикловой комиссии

Общетехнических дисциплин и компьютерных технологий

Протокол № 4

От 09.12.2025 г.

Председатель цикловой комиссии:

Шурухина И.Е.

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания

формируемые ПК, ОК, ЛР	Умения	Знания
ОК 01-02, ОК 05-06 ПК 1.1; ПК1.3-1.4 ПК 1.6-1,7; ЛР13-17	– разрабатывать алгоритмы для конкретных задач; – использовать программы для графического отображения алгоритмов; – определять сложность работы алгоритмов; – работать в среде программирования; – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; – оформлять код программы в соответствии со стандартом кодирования; – выполнять проверку, отладку кода программы;	– понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; – эволюцию языков программирования, их классификацию, понятие системы программирования; основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; – подпрограммы, составление библиотек подпрограмм; – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения.

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.3. Оформлять программный код в соответствии с техническим заданием.

ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.

ПК 1.6. Выполнять тестирование программного кода.

ПК 1.7. Составлять тестовые сценарии.

Практические работы

тема	название ПР	часы
Тема 1.1. Основы алгоритмизации	**Практические занятия** Составление блок-схем линейных алгоритмов.	2
	Практические занятия Составление блок-схем разветвляющихся алгоритмов.	1
	Практические занятия Составление блок-схем циклических алгоритмов.	1
Тема 2.1 Операторы языка программирования	**Практические занятия** Линейный алгоритм. Пример программы, позволяющей решить линейное уравнение. Составление программ линейной структуры	1
	Практические занятия Программирование циклических алгоритмов: цикл с параметром.	1
	Практические занятия Программирование циклических алгоритмов: цикл с предусловием. Программирование циклических алгоритмов: цикл с постусловием.	1
	Практические занятия Программирование циклических алгоритмов: вложенные циклы.	1
	Практические занятия Обработка одномерных и двумерных массивов.	1
	Практические занятия Различные методы упорядочения алгоритмов.	1
Тема 2.2 Процедуры и функции	**Практические занятия** Множественное присваивание. Операции обработки числовых данных	1
	Практические занятия Функция модуля числа. Операции: безостаточного деления (a//b) и остаток от деления(a%b).	1
	Практические занятия Функция генерации целого случайного числа.	1
	Практические занятия Функции модуля Math. Функция округления к целому значению. Функция выделения целой части.	1
	Практические занятия Логические высказывания. Переменные типа bool().	2
	Практические занятия Формы инструкции ветвления. Пример использования.	2
	Практические занятия Основные логические операции: конъюнкция, дизъюнкция, инверсия.	2
	Практические занятия Запись каскадного ветвления с использованием примера решения квадратного уравнения.	1
	Практические занятия Особенности работы цикла с предусловием. Решение задач с применением циклических алгоритмов с предусловием.	1
	Практические занятия Инструкции управления циклом break(), continue().	1
	Практические занятия Решение задач с применением циклических алгоритмов с постусловием.	1
	Практические занятия Запись цикла с параметром. Применение функции генерации множества значений из диапазона.	2
Тема 2.3 Этапы решения задачи на	**Практические занятия** Словари. Ключ в словаре. Вхождение в словаре.	1
	Практические занятия Определение и обработка	1

компьютере.	исключений.	
	Практические занятия Командный способ организации систем управления.	1
	Практические занятия Пакетный процесс организации систем управления.	1
	Практические занятия Диалоговый способ организации систем управления.	1
	Практические занятия Принцип обратной связи – адаптивное управление.	1
	Практические занятия Каскадная модель разработки программного обеспечения.	1
	Практические занятия Введение в язык программирования Python.	1
	Практические занятия История и особенности языка программирования Python. Первый запуск среды разработки.	1
	Практические занятия Функции. Применение функций при решении задач.	1
	Практические занятия Линейные последовательности данных – кортежи, списки.	1
	Практические занятия Обработка списков. Линейный поиск.	1
	Практические занятия Обработка списков. Двоичный поиск и сортировка.	1
	Практические занятия Реализация матриц на языке Python. Обработка матриц.	1
	Практические занятия Метод последовательной детализации.	1
	Практические занятия Рекурсивные методы	1
	Практические занятия Методы перебора в задачах поиска.	1
	Практические занятия Методы сортировки данных и сложность алгоритмов	1
Тема 4.1 Концепция объектно-ориентированного программирования	**Практические занятия** Классы, объекты: свойства, методы.	2
	Практические занятия Конструкторы.	2
Тема 4.2 Применение модуля turtle-приложения с графическим пользовательским интерфейсом.	**Практические занятия** Рисование стандартных фигур: правильный n-угольник, угол. Примеры программ.	1
	Практические занятия Рисование стандартных фигур: главная диагональ. Примеры программ.	1
	Практические занятия Создание приложения с использованием виджетов.	1
	Практические занятия Создание графического интерфейса без использования программы-визуализатора.	1
	Практические занятия Управление макетом графического интерфейса. Блочный, сеточный макет.	1
	Практические занятия Рисование на форме. Модель обработки данных в приложении с графическим интерфейсом. Представление в приложении с графическим интерфейсом.	1
	Практические занятия Создание собственного виджета.	2
		56

Описание порядка выполнения практических работ

Практическая работа
Построение линейных и разветвляющихся алгоритмов

Задание:

Вариант № 1

Оформление работы:

- Задание выполняется с использованием стандартного приложения ОС Windows Paint или Microsoft Word. Сохранить под именем Пр1_№ своего варианта.
- В отчёте к заданию должна быть титульная страница с названием и номером работы.
- Каждое задание в отчёте на отдельной странице.
- Страница должна начинаться с заголовка Задание №__
- Вторая страница в отчёте — оглавление с заголовками Задание №__

Задание:

1. Составить алгоритм вычисления площади боковой поверхности цилиндра ($S_b = Ph$, $P = 2\pi R$), если высота цилиндра
   5 см, а радиус 2 см

Составить алгоритм для следующей системы:
$y=\{x3-4x$, если $x\geq 5$ $x2+5x$, если $0\leq x<5$ $x-3$,если $x<0$

Вариант № 2

1. Вычислить площадь боковой поверхности цилиндра (Sb = Ph, P = 2πR), если высота цилиндра
   6 см, а радиус 3 см

Составить алгоритм для следующей системы:
$y=\{x2+x$, если $x\geq 2$ $x4+2x$, если $0\leq x<2$ $x+5$,если $x<0$

Практическая работа
Построение циклических алгоритмов

Задание:

Вариант № 1

Оформление работы:

- Задание выполняется с использованием стандартного приложения ОС Windows Paint или Microsoft Word. Сохранить под именем Пр2_№ своего варианта.
- В отчёте к заданию должна быть титульная страница с названием и номером работы.
- Каждое задание в отчёте на отдельной странице.
- Страница должна начинаться с заголовка Задание №__
- Вторая страница в отчёте — оглавление с заголовками Задание №__

Задание:

1. Составить блок-схему вычисления суммы всех целых нечётных чисел от 1 до 100
2. Составить блок-схему вычисления функции y = a / (a+ x) при x, изменяющимся от x = 1 до x = 4 с шагом Dx = 0,5
3. Вычислите $(2-x)^2+(2-x)^4+(2-x)^6+(2-x)^8+(2-x)^{10}$

Вариант № 2

Задание:

1. Составить блок-схему вычисления суммы всех целых чётных чисел от 1 до 100
2. Составить блок-схему вычисления функции x = b / (b -y) при y, изменяющимся от y = 4 до y = 1 с шагом Dy = 0,6
3. Вычислите (2t+3) + (2t+6) + (2t+9) + ...+ (2t+30)

Практическая работа
Разработка алгоритмов шифрования

Задание 1 Составить алгоритм шифрования текста с помощью шифра Цезаря.
Описание шифра: чтобы зашифровать текст, записанный с помощью русских букв и знаков препинания, его можно переписать, заменив каждую букву непосредственно следующей за ней буквой по алфавиту (буква Я заменяется на А). Обобщив этот способ шифровки, можно производить сдвиг не на одну букву, а на N букв (N – натуральное число).

Задание 2 Составить алгоритм шифрования текста с помощью шифра «Матричная шифровка». Описание шифра: чтобы зашифровать текст из 121 буквы, его можно записать в квадратную матрицу порядка 11 по строкам, а затем прочитать по спирали, начиная с центра (т.е. с элемента, имеющего индексы 6,6). Такой способ можно обобщить и для произвольной длины текста, подбирая нужный размер матрицы.

Задание 3 Составить алгоритм шифрования текста с помощью шифра «Шифровка решеткой». Шифровка текста с помощью решетки заключается в следующем. Решетка, т.е. квадрат из клетчатой бумаги 10 × 10 клеток, некоторые клетки в котором вырезаны, совмещается с целым квадратом 10 × 10 клеток, и через прорези на бумагу наносятся первые буквы текста. Затем решетка поворачивается на 90 градусов и через прорези записываются следующие буквы. Это повторяется еще дважды. Таким образом на бумагу наносятся 100 букв текста. Решетку можно изобразить квадратной матрицей порядка 10 из нулей и единиц (нуль изображает прорезь). Доказано, что матрица [Aij], i = 1,…, 10, j = 1, …, 10 может служить ключом шифра, если из элементов в точности один равен нулю.

Задание 4 Составить алгоритм шифрования текста с помощью шифра «Шифровка зафиксированной перестановкой». Описание шифра. Зафиксируем натуральное k и перестановку чисел 1, …, k (ее можно задать с помощью последовательности натуральных чисел p1, …, pk, в которую входит каждое из чисел 1,… k). При шифровке в исходном тексте к каждой из последовательных групп по k символов применяется зафиксированная перестановка. Пусть k = 4 и перестановка есть 3, 2, 4, 1 Тогда группа символов s1, s2, s3, s4 заменяется на s3, s2, s4, s1. Если в последней группе меньше четырех символов, то к ней добавляются пробелы.
Задание 5 Составить алгоритм шифрования текста с помощью шифра «Атбаш». Описание шифра. Шифр простой замены, использованный для еврейского алфавита и получивший оттуда свое название. Шифрование происходит заменой первой буквы алфавита на последнюю, второй на предпоследнюю (алеф (первая буква) заменяется на тав (последнюю), бет (вторая) заменяется на шин (предпоследняя); из этих сочетаний шифр и получил свое название).
Шифр Атбаш для английского алфавита:
Исходный алфавит: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Алфавит замены: Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

Задание 6 Составить алгоритм шифрования текста с помощью шифра «Тарабарская грамота».
Суть шифровки, которая использовалась в XV–XVI веках на Руси, в следующем. Все согласные буквы русской азбуки записывались в два ряда; одна половина букв вверху, другая половина – внизу, причем в обратном порядке (одна буква под другой):
Б В Г Д Ж З К Л М Н
Щ Ш Ч Ц Х Ф Т С Р П
При зашифровке слов согласные взаимно заменялись, а гласные, Й и буквы Ъ, Ь вписывались без изменений. Слова записывались без промежутков между ними, как вообще писался любой текст до XVI века, и это еще больше затрудняло разгадывание.

Практическая работа
Построение алгоритмов различных конструкций

Оформление работы:

- Задание выполняется с использованием стандартного приложения ОС Windows Paint или Microsoft Word. Сохранить под именем Пр3_№ своего варианта.
- В отчёте к заданию должна быть титульная страница с названием и номером работы.
- Каждое задание в отчёте на отдельной странице.
- Страница должна начинаться с заголовка Задание №__
- Вторая страница в отчёте — оглавление с заголовками Задание №__

Задание 1. Создание линейного алгоритма

Задание 2. Создание алгоритма ветвления

Задание 3. Создание алгоритма цикла

Задание 4. Создание алгоритма массива

Практическая работа
Использование языка программирования Python для создания программ с линейным алгоритмом

Задание:

Вариант № 1

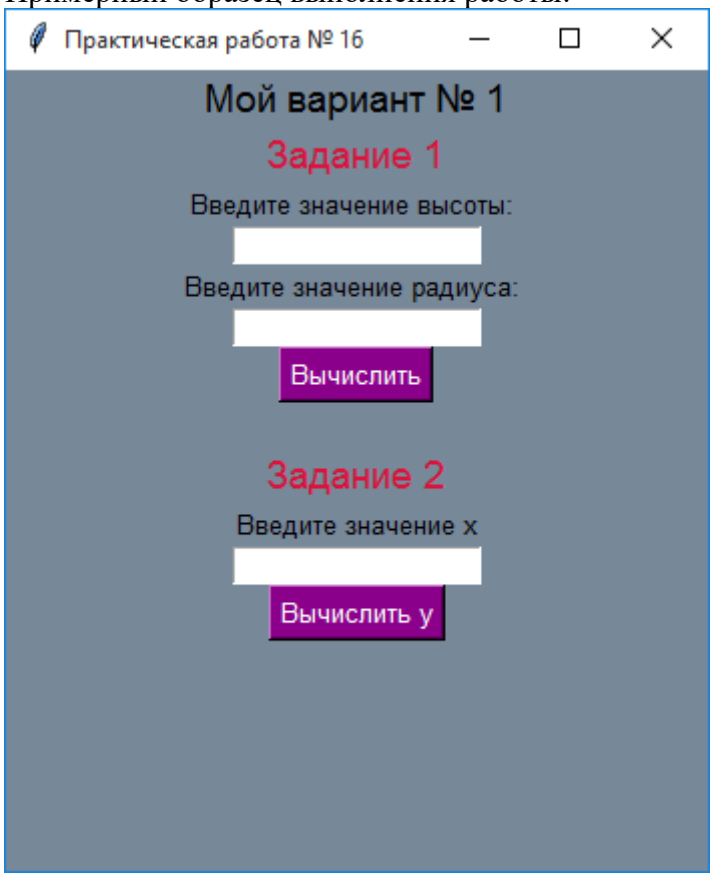
1. Написать программу на языке Python для вычисления площади боковой поверхности цилиндра (Sb = Ph, P = 2πR), если высота цилиндра 5 см, а радиус 2 см

Написать программу на языке Python для следующей системы:
y={x3-4x, если x≥5 x2+5x, если 0≤x<5 x-3,если x<0

Вариант № 2

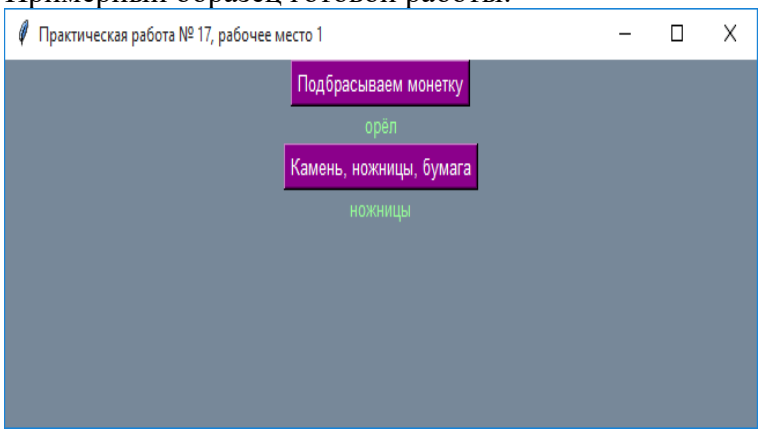
1. Вычислить площадь боковой поверхности цилиндра (Sb = Ph, P = 2πR), если высота цилиндра 6 см, а радиус 3 см

Написать программу для следующей системы:
y={x2+x, если x≥2 x4+2x, если 0≤x<2 x+5,если x<0
Примерный образец выполнения работы:



Практическая работа
Использование языка программирования Python для создания программ в отдельном файле с использованием строк и операторов отношений

Задание:
Написать программу с графическим интерфейсом, используя метки, кнопки, подключить модуль random для генерации случайных значений.
Примерный образец готовой работы:

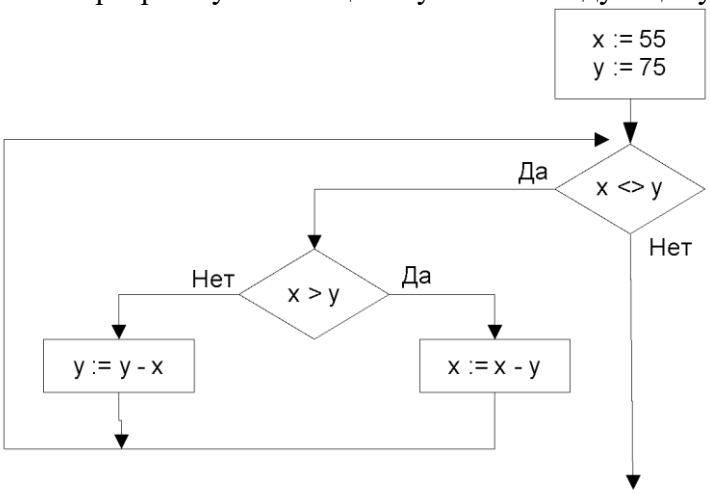


Практическая работа
Использование языка программирования Python для создания программ с цикла-ми while и for

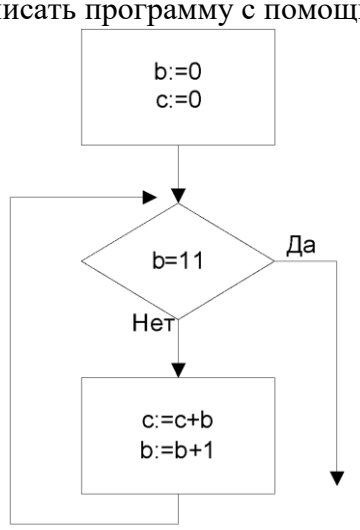
Задание:

Вариант № 1

1. Написать программу с помощью Python по следующему алгоритму:

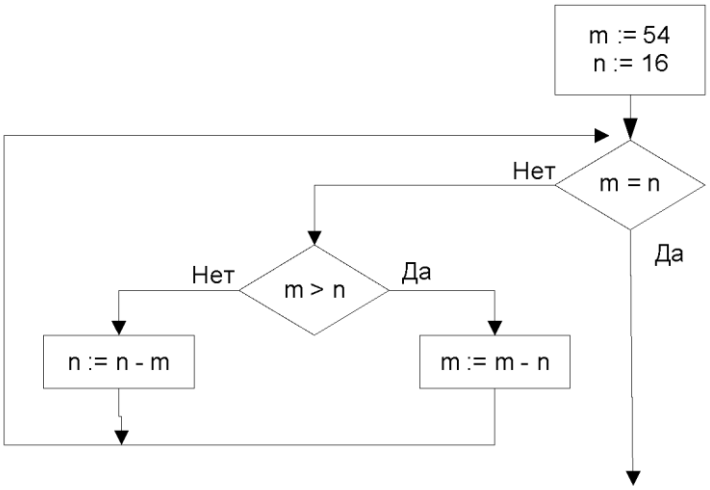


2.
3. Написать программу с помощью Python по следующему алгоритму:

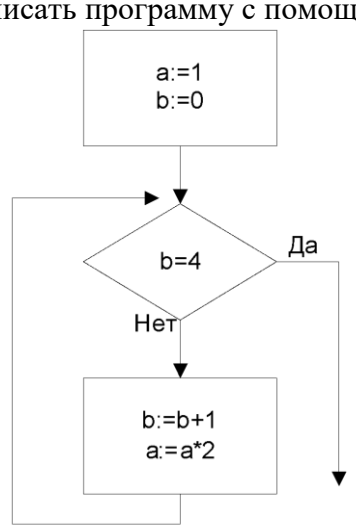


Вариант № 2

1. Написать программу с помощью Python по следующему алгоритму:



2. Написать программу с помощью Python по следующему алгоритму:



Практическая работа
Использование языка программирования Python для создания программ по работе с файлами и файловой структурой

Задание:

1. В своей папке создать файл 1.txt, 2.txt. Сделать несколько строк записей в этих файлах.
2. С помощью модуля os в Python получить список папок/файлов в вашей папке.
3. С помощью модуля os в Python создать папку 1 на диске D.
4. Создать дерево каталогов MY\Python\os на диске D
5. С помощью модуля os в Python переместить файлы 1.txt и 2.txt в папку 1 на диске D и переименовать в my.txt и в my_1.txt
6. Открыть файл с помощью Python.
7. Все набранные команды в окне программирования на языке Python в виде скриншота вставить в отчёт, выполненный в формате docx.

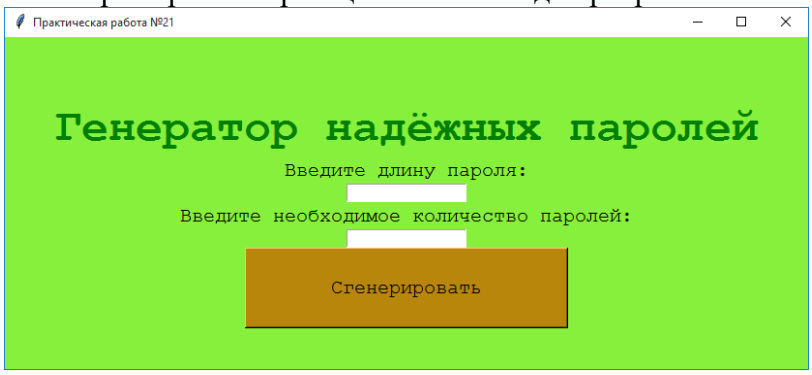
Практическая работа
Использование языка программирования Python для создания программ с использованием классов

Задание:
Создать программу на языке Python по следующему пояснению:
Создайте класс Student. Определите атрибуты name (имя), group (номер группы) и spec (специальность). Добавьте метод под названием study («учит»). Создайте объект класса Mygr, установите атрибуты, вызовите метод study.

Практическая работа
Построение логически правильных и эффективных программ на языке программирования Python

Задание:
Создать программу Генератор паролей с графическим интерфейсом с запросом о количестве паролей и записью в файл всех сгенерированных паролей.
Допустимые символы: заглавные буквы латиницы, маленькие буквы латиницы, знаки _@#!%&*

Примерный образец внешнего вида программы:



Практическая работа
Использование виртуальных окружений для изолирования различных проектов. Полезные функции и модули языка.

Задание:

1. Установить Python 3.7 на Debian
2. Задать использование Python 3.7 по умолчанию в виртуальном окружении.
3. Установите с помощью pip virtualenvwrapper

   Отредактируйте файл .bashrc, дописав в него:

   ```
   export VIRTUALENVWRAPPER_PYTHON=/usr/local/bin/python3.7
   export WORKON_HOME=~/venv
   . /usr/local/bin/virtualenvwrapper.sh
   ```

4. Перезапустите командный интерпретатор
5. Создайте новое виртуального окружения, в котором Python 3.7 используется по умолчанию
6. Укажите список установленных пакетов в виртуальном окружении

Практическая работа
Использование регулярных выражений для проверки конфигурации сетевого оборудования.

Задание:

Создать функцию get_ip_from_cfg, которая ожидает как аргумент имя файла, в котором находится конфигурация устройства.

Функция должна обрабатывать конфигурацию и возвращать IP-адреса и маски, которые настроены на интерфейсах, в виде списка кортежей:

• первый элемент кортежа - IP-адрес

• второй элемент кортежа - маска

Для получения такого результата, используйте регулярные выражения.

Проверить работу функции на примере файла config_r1.txt.

Обратите внимание, что в данном случае, можно не проверять корректность IP-адреса, диапазоны адресов и так далее, так как обрабатывается вывод команды, а не ввод пользователя

Практическая работа
Чтение и запись данных в форматах CSV, JSON и YAML.

Задание:
Создать функцию write_dhcp_snooping_to_csv, которая обрабатывает вывод команды show dhcp snooping binding из разных файлов и записывает обработанные данные в csv файл.
Аргументы функции:
• filenames - список с именами файлов с выводом show dhcp snooping binding
• output - имя файла в формате csv, в который будет записан результат Функция ничего не возвращает.
Проверить работу функции на содержимом файлов sw1_dhcp_snooping.txt, sw2_dhcp_snooping.txt, sw3_dhcp_snooping.txt. Первый столбец в csv файле имя коммутатора надо получить из имени файла, остальные - из содержимого в файлах.

Практическая работа
Подключение к оборудованию по SSH и Telnet

Задание:
Создать функцию send_show_command.
Функция подключается по SSH (с помощью netmiko) к ОДНОМУ устройству и выполняет указанную команду.
Параметры функции:
• device - словарь с параметрами подключения к устройству
• command - команда, которую надо выполнить
Функция возвращает строку с выводом команды.
Скрипт должен отправлять команду command на все устройства из файла devices.yaml с помощью функции send_show_command (эта часть кода написана).

Практическая работа
Одновременное подключение к нескольким устройствам

Задание:

Создать функцию ping_ip_addresses, которая проверяет пингуются ли IP-адреса. Проверка IPадресов должна выполняться параллельно в разных потоках.
Параметры функции:
• ip_list - список IP-адресов
• limit - максимальное количество параллельных потоков (по умолчанию 3)
Функция должна возвращать кортеж с двумя списками:
• список доступных IP-адресов
• список недоступных IP-адресов
Для выполнения задания можно создавать любые дополнительные функции.
Для проверки доступности IP-адреса, используйте ping.

Практическая работа
Создание шаблонов конфигурации с помощью Jinja2

Задание:

Создать функцию generate_config.
Параметры функции:
• template - путь к файлу с шаблоном (например, «templates/for.txt»)
• data_dict - словарь со значениями, которые надо подставить в шаблон
Функция должна возвращать строку с конфигурацией, которая была сгенерирована.
Проверить работу функции на шаблоне templates/for.txt и данных из файла data_files/for.yml.

```python
import yaml


# так должен выглядеть вызов функции
if __name__ == "__main__":
    data_file = "data_files/for.yml"
    template_file = "templates/for.txt"
    with open(data_file) as f:
        data = yaml.safe_load(f)
    print(generate_config(template_file, data))
```

Практическая работа
Обработка вывода команд с помощью TextFSM

Задание:
Создать функцию parse_command_output. Параметры функции:
• template - имя файла, в котором находится шаблон TextFSM (templates/sh_ip_int_br.template)
• command_output - вывод соответствующей команды show (строка)
Функция должна возвращать список:
• первый элемент - это список с названиями столбцов
• остальные элементы это списки, в котором находятся результаты обработки вывода
Проверить работу функции на выводе команды output/sh_ip_int_br.txt и шаблоне templates/sh_ip_int_br.template.

```python
from netmiko import ConnectHandler


# вызов функции должен выглядеть так
if __name__ == "__main__":
    r1_params = {
        "device_type": "cisco_ios",
        "host": "192.168.100.1",
        "username": "cisco",
        "password": "cisco",
        "secret": "cisco",
    }
    with ConnectHandler(**r1_params) as r1:
        r1.enable()
        output = r1.send_command("sh ip int br")
    result = parse_command_output("templates/sh_ip_int_br.template", output)
    print(result)
```

Практическая работа
Использование объектно-ориентированного программирования для чтения «чужого» кода, кода netmiko.

Задание:

Создать класс MyNetmiko, который наследует класс CiscoIosSSH из netmiko.
Переписать метод __init__ в классе MyNetmiko таким образом, чтобы после подключения по
SSH выполнялся переход в режим enable.
Для этого в методе __init__ должен сначала вызываться метод __init__ класса CiscoIosBase, а
затем выполнялся переход в режим enable.
Проверить, что в классе MyNetmiko доступны методы send_command и send_config_set

Практическая работа
Использование наследования для создания новых классов на основе существующих.

Задание:

Создать класс CiscoSSH, который наследует класс BaseSSH из файла base_connect_class.py.
Создать метод __init__ в классе CiscoSSH таким образом, чтобы после подключения по SSH
выполнялся переход в режим enable.
Для этого в методе __init__ должен сначала вызываться метод __init__ класса ConnectSSH, а
затем выполняться переход в режим enable

```
In [2]: from task_24_1 import CiscoSSH

In [3]: r1 = CiscoSSH(**device_params)

In [4]: r1.send_show_command('sh ip int br')
Out[4]: 'Interface                  IP-Address      OK? Method Status
↪   Protocol\nEthernet0/0               192.168.100.1   YES NVRAM  up
↪         up      \nEthernet0/1               192.168.200.1   YES NVRAM  up
↪               up      \nEthernet0/2               190.16.200.1    YES NVRAM
↪up                   up      \nEthernet0/3               192.168.230.1   YES
↪NVRAM  up                    up      \nEthernet0/3.100           10.100.0.1
↪  YES NVRAM  up                    up      \nEthernet0/3.200           10.200.
↪0.1      YES NVRAM  up                    up      \nEthernet0/3.300
↪10.30.0.1       YES NVRAM  up                    up      '
```

Практическая работа
Работа с базами данных

Задание:
Необходимо создать два скрипта:
1. create_db.py
2. add_data.py
Код в скриптах должен быть разбит на функции. Какие именно функции и как разделить код,
надо решить самостоятельно. Часть кода может быть глобальной.
1. create_db.py - в этот скрипт должна быть вынесена функциональность по созданию БД:
• должна выполняться проверка наличия файла БД
• если файла нет, согласно описанию схемы БД в файле dhcp_snooping_schema.sql, должна быть создана БД
• имя файла бд - dhcp_snooping.db
В БД должно быть две таблицы (схема описана в файле dhcp_snooping_schema.sql):
• switches - в ней находятся данные о коммутаторах
• dhcp - тут хранится информация полученная из вывода sh ip dhcp snooping binding
Пример выполнения скрипта, когда файла dhcp_snooping.db нет:
$ python create_db.py
Создаю базу данных...
После создания файла:
$ python create_db.py
База данных существует
2. add_data.py - с помощью этого скрипта, выполняется добавление данных в БД. Скрипт должен добавлять данные из вывода sh ip dhcp snooping binding и информацию о коммутаторах Соответственно, в файле add_data.py должны быть две части:
• информация о коммутаторах добавляется в таблицу switches
– данные о коммутаторах, находятся в файле switches.yml
• информация на основании вывода sh ip dhcp snooping binding добавляется в таблицу dhcp
– вывод с трёх коммутаторов: файлы sw1_dhcp_snooping.txt, sw2_dhcp_snooping.txt, sw3_dhcp_snooping.txt
– так как таблица dhcp изменилась, и в ней теперь присутствует поле switch, его нужно также заполнять. Имя коммутатора определяется по имени файла с данными
Пример выполнения скрипта, когда база данных еще не создана:
$ python add_data.py
База данных не существует. Перед добавлением данных, ее надо создать
Пример выполнения скрипта первый раз, после создания базы данных:
$ python add_data.py
Добавляю данные в таблицу switches...
Добавляю данные в таблицу dhcp...