

**Санкт-Петербургское государственное бюджетное
профессиональное образовательное учреждение
«Академия управления городской средой, градостроительства и печати»**



КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

**по текущему контролю успеваемости
и промежуточной аттестации
по учебной дисциплине
ОП.03 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ**

программы подготовки специалистов среднего звена

по специальности

10.02.05 Обеспечение информационной безопасности автоматизированных систем

Санкт-Петербург
2023 г.

Комплект контрольно-оценочных средств по учебной дисциплине разработан на основе Федерального государственного образовательного стандарта по специальности 10.02.05 Обеспечение информационной безопасности автоматизированных систем, утвержденного приказом Минобрнауки России от 09.12.2016 № 1553, в соответствии с рабочей программой учебной дисциплины ОП.03 Основы алгоритмизации и программирования.

Комплект контрольно-оценочных средств рассмотрен на заседании методического совета СПб ГБПОУ «АУГСГиП»

Протокол № 2 от «29» ноября 2023 г.

Комплект контрольно-оценочных средств одобрен на заседании цикловой комиссии общетехнических дисциплин и компьютерных технологий

Протокол № 4 от «21» ноября 2023 г.

Председатель цикловой комиссии: Караченцева М.С.



Разработчики: преподаватели СПб ГБПОУ «АУГСГиП»

СОДЕРЖАНИЕ

1. Паспорт комплекта оценочных средств.....	4
2. Результаты освоения учебной дисциплины, подлежащие проверке	6
3. Оценка освоения учебной дисциплины	7
3.1. Текущий контроль. Задания для текущей аттестации	7
3.2. Контрольно-оценочные материалы для промежуточной аттестации по дисциплине « Основы алгоритмизации и программирования»	25

1. Паспорт комплекта оценочных средств

В результате освоения учебной дисциплины «Основы алгоритмизации и программирования» обучающийся должен обладать предусмотренными ФГОС по специальности 10.02.05 «Обеспечение информационной безопасности автоматизированных систем» следующими умениями, знаниями, которые формируют профессиональные и общие компетенции:

умения:

- У1 разрабатывать алгоритмы для конкретных задач.
- У2 использовать программы для графического отображения алгоритмов.
- У3 определять сложность работы алгоритмов.
- У4 работать в среде программирования.
- У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- У6 оформлять код программы в соответствии со стандартом кодирования.
- У7 выполнять проверку, отладку кода программы.

знания:

- З1 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- З2 эволюцию языков программирования, их классификацию, понятие системы программирования.
- З3 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- З4 подпрограммы, составление библиотек подпрограмм.
- З5 объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.

общие компетенции:

- ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
- ОК 02 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
- ОК 04 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
- ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
- ОК 09 Использовать информационные технологии в профессиональной деятельности
- ОК 10 Пользоваться профессиональной документацией на государственном и иностранном языках

профессиональные компетенции

- ПК 2.1. Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.

ПК 2.2. Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.

ПК 2.3. Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.

Формой *промежуточной аттестации* по учебной дисциплине является **экзамен**.

Текущий контроль освоения обучающимися программного материала учебной дисциплины проводится с целью объективной оценки качества освоения программы учебной дисциплины, а также стимулирования учебной работы обучающихся, мониторинга результатов образовательной деятельности, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебно-воспитательного процесса.

2. Результаты освоения учебной дисциплины, подлежащие проверке

2.1. В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций:

Таблица 1

Контроль и оценка результатов освоения дисциплины

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<i>Умения</i>	
У1 разрабатывать алгоритмы для конкретных задач.	Оценка практических работ Оценка практических заданий на экзамене
У2 использовать программы для графического отображения алгоритмов.	
У3 определять сложность работы алгоритмов.	
У4 работать в среде программирования.	
У5 реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.	
У6 оформлять код программы в соответствии со стандартом кодирования.	
У7 выполнять проверку, отладку кода программы.	
<i>Знания</i>	
З1 понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	Устный зачет по темам Оценка практических заданий на экзамене
З2 эволюцию языков программирования, их классификацию, понятие системы программирования	
З3 основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.	
З4 подпрограммы, составление библиотек подпрограмм.	
З5 объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	

3. Оценка освоения учебной дисциплины
3.1. Текущий контроль. Задания для текущей аттестации

Проводится преподавателем на учебных занятиях согласно календарно-тематическому плану. Формы текущего контроля выбраны, исходя из методической целесообразности.

Таблица 2

Распределение контрольных точек по дисциплине

Дидактические единицы	Проверяемые З,У,ОК,ПК	Формы контроля (наименование контрольной точки)	
		Текущая аттестация	Промежуточная аттестация
Тема 1. Алгоритмизация	31-35 У1-У7 ОК1, ОК2, ОК4, ОК5, ОК9, ОК10 ПК1-ПК3	Устный зачет по теме 1	Выполнение практических заданий на экзамен
		Практическая работа № 1 Построение линейных и разветвляющихся алгоритмов	
		Практическая работа № 2 Построение циклических алгоритмов	
Тема 2. Языки и системы программирования	31-35 ОК1, ОК2, ОК4, ОК5, ОК9, ОК10 ПК1-ПК3	Устный зачет по теме 2.	
Тема 3. Программирование на языке Python	31-35 У1-У7 ОК1, ОК2, ОК4, ОК5, ОК9, ОК10 ПК1-ПК3	Устный зачет по теме 3	
		Практическая работа № 5 Использование языка программирования Python для создания программ с линейным алгоритмом	
		Практическая работа № 6 Использование языка программирования Python для создания программ в отдельном файле с использованием строк и операторов отношений	

Дидактические единицы	Проверяемые З,У,ОК,ПК	Формы контроля (наименование контрольной точки)	
		Текущая аттестация	Промежуточная аттестация
			<p>Практическая работа № 7 Использование языка программирования Python для создания программ с циклами while и for</p> <p>Практическая работа № 8 Использование языка программирования Python для создания программ по работе с файлами и файловой структурой</p>
Тема 4. Использование языка программирования Python для автоматизации управления СКС	31-35 У1-У7 ОК1, ОК2, ОК4, ОК5, ОК9, ОК10 ПК1-ПК3	Практическая работа № 9 Использование языка программирования Python для создания программ с использованием классов	
		Практическая работа № 10 Построение логически правильных и эффективных программ на языке программирования Python	
		Практическая работа № 11 Использование виртуальных окружений для изолирования различных проектов. Полезные функции и модули языка.	
		Практическая работа № 13 Чтение и запись данных в форматах CSV, JSON и YAML.	
		Практическая работа № 14 Подключение к оборудованию по SSH и Telnet	
		Практическая работа № 20 Работа с базами данных	

Устный зачет по теме 1

Инструкция для обучающихся

Зачет сдается в рамках учебного занятия. Каждый студент отвечает в устной форме на предложенные преподавателем 5 случайных вопроса.

Выполнение задания: одному студенту на ответ выделяется 3 мин., группа сдает зачет за одно учебное занятие.

Перечень вопросов:

1. Основные принципы построения алгоритмов.
2. Основные алгоритмические конструкции
3. Понятие линейного алгоритма
4. Понятие разветвляющегося алгоритма

Эталоны ответов: приведены в учебном пособии по дисциплине «Основы алгоритмизации и программирования».

Практическая работа № 1 «Построение линейных и разветвляющихся алгоритмов»

Инструкция для обучающихся

Внимательно прочитайте задание. Постройте блок-схемы алгоритмов.

Задание

Вариант № 1

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:
 $y = \begin{cases} x^3 - 4x, & \text{если } x \geq 5 \\ x^2 + 5x, & \text{если } 0 \leq x < 5 \\ x - 3, & \text{если } x < 0 \end{cases}$

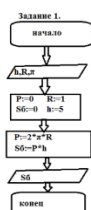
Вариант № 2

1. Вычислить площадь боковой поверхности цилиндра ($S_b = Ph$, $P = 2\pi R$), если высота цилиндра 6 см, а радиус 3 см

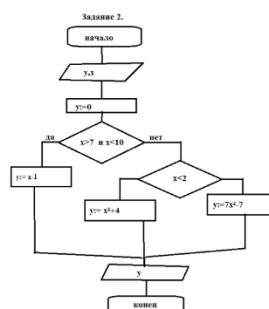
Составить алгоритм для следующей системы:
 $y = \begin{cases} x^2 + x, & \text{если } x \geq 2 \\ x^4 + 2x, & \text{если } 0 \leq x < 2 \\ x + 5, & \text{если } x < 0 \end{cases}$

Эталон ответа

Задание 1
Вычислить площадь боковой поверхности цилиндра ($S_b = Ph$, $P = 2\pi R$), если высота цилиндра 5 см, а радиус 2 см



Задание 2
Составить алгоритм для следующей системы:
 $y = \begin{cases} x^2 + x, & \text{если } x \geq 2 \\ x^4 + 2x, & \text{если } 0 \leq x < 2 \\ x + 5, & \text{если } x < 0 \end{cases}$



Практическая работа № 2 «Построение циклических алгоритмов»

Инструкция для обучающихся

Внимательно прочитайте задание. Постройте блок-схемы алгоритмов.

Задание

Вариант № 1

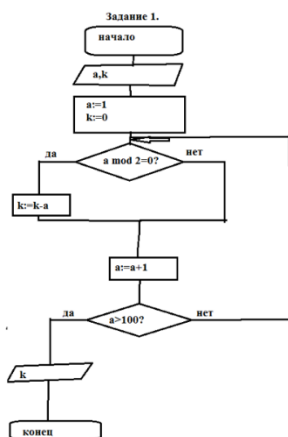
1. Составить блок-схему вычисления суммы всех целых нечётных чисел от 1 до 100
2. Составить блок-схему вычисления функции $y = a / (a + x)$ при x , изменяющимся от $x = 1$ до $x = 4$ с шагом $Dx = 0,5$
3. Вычислите $(2-x)^2 + (2-x)^4 + (2-x)^6 + (2-x)^8 + (2-x)^{10}$

Вариант № 2

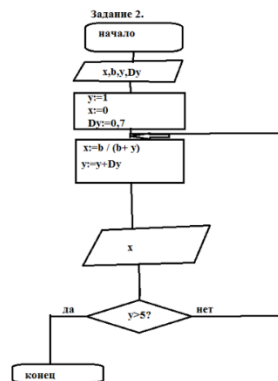
1. Составить блок-схему вычисления суммы всех целых чётных чисел от 1 до 100
2. Составить блок-схему вычисления функции $x = b / (b - y)$ при y , изменяющимся от $y = 4$ до $y = 1$ с шагом $Dy = 0,6$
3. Вычислите $(2t+3) + (2t+6) + (2t+9) + \dots + (2t+30)$

Эталон ответа

Задание 1
Вычислить сумму всех целых чётных чисел от 1 до 100



Задание 2
Составить блок-схему вычисления функции $x = b / (b - y)$ при y , изменяющимся от $y = 4$ до $y = 1$ с шагом $Dy = 0,7$



Устный зачет по теме 2

Инструкция для обучающихся

Зачет сдается в рамках учебного занятия. Каждый студент отвечает в устной форме на предложенные преподавателем 4 случайных вопроса.

Выполнение задания: одному студенту на ответ выделяется 3 мин., группа сдает зачет за одно учебное занятие.

Перечень вопросов:

1. Языки программирования

2. Классификация языков программирования
3. Этапы разработки программ
4. Модули системы программирования
5. Основные элементы ООП
6. Свойства ООП
7. Структуры данных
8. Типы данных
9. Понятие отладки программного кода

Эталоны ответов: приведены в учебном пособии по дисциплине «Основы алгоритмизации и программирования».

Устный зачет по теме 3

Инструкция для обучающихся

Зачет сдается в рамках учебного занятия. Каждый студент отвечает в устной форме на предложенные преподавателем 2 вопроса.

Выполнение задания: одному студенту на ответ выделяется 3 мин., группа сдает зачет за одно учебное занятие.

Перечень вопросов:

1. Интегрированная среда языка программирования Python
2. Синтаксис языка Python
3. Операторы отношений.
4. Условная инструкция if
5. Модуль tkinter

Эталоны ответов: приведены в учебном пособии по дисциплине «Основы алгоритмизации и программирования».

Практическая работа № 5

«Использование языка программирования Python для создания программ с линейным алгоритмом»

Инструкция для обучающихся

Внимательно прочитайте задание. Создайте программу на языке программирования Python.

Задание

Вариант № 1

1. Написать программу на языке Python для вычисления площади боковой поверхности цилиндра ($S_b = Ph$, $P = 2\pi R$), если высота цилиндра 5 см, а радиус 2 см

Написать программу на языке Python для следующей системы:

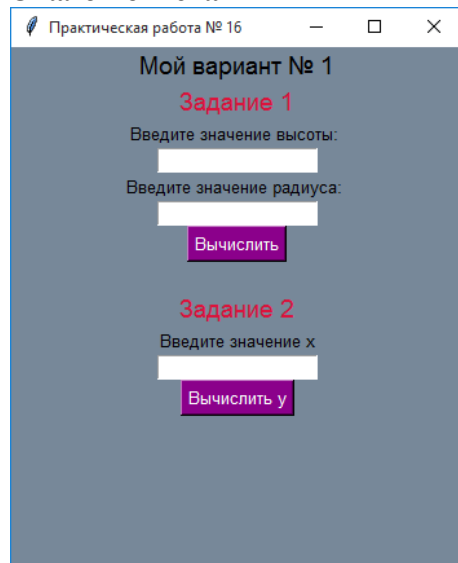
$$y = \begin{cases} x^3 - 4x, & \text{если } x \geq 5 \\ x^2 + 5x, & \text{если } 0 \leq x < 5 \\ x - 3, & \text{если } x < 0 \end{cases}$$

Вариант № 2

1. Вычислить площадь боковой поверхности цилиндра ($S_b = Ph$, $P = 2\pi R$), если высота цилиндра 6 см, а радиус 3 см

Написать программу для следующей системы:
 $y = \begin{cases} x^2 + x, & \text{если } x \geq 2 \\ x^4 + 2x, & \text{если } 0 \leq x < 2 \\ x + 5, & \text{если } x < 0 \end{cases}$

Эталон ответа



```

from tkinter import *
from math import *

def click1():
    try:
        H = float(a.get())
        R = float(b.get())
        lab3.config(text='Площадь цилиндра равна {}'.format(2*pi*R*H))
    except ValueError:
        lab3.config(text="Ошибка! Введите цифры")

def click2():
    try:
        x = float(c.get())
        if x < -20:
            y = 20
        if -20 < x <= 30:
            y = abs(x)
        if x > 30:
            y = 30
        lab6.config(text='Значение y равно {}'.format(y))
    except ValueError:
        lab6.config(text="Ошибка! Введите цифры")

root = Tk()
root.title('Практическая работа №16')
root.geometry('400x500')
root["bg"] = "grey"

lab = Label(root, text="Мой вариант №13", font="Courier 20", fg="pink", bg="grey")
lab.pack()

lab0 = Label(root, text="Задание 1", font="Courier 15", fg="red", bg="grey")
lab0.pack()

lab1 = Label(root, text="Введите значение высоты:", font="Courier 10", fg="pink", bg="grey")
a = Entry(root, width=25)
lab1.pack()
a.pack()

lab2 = Label(root, text="Введите значение радиуса:", font="Courier 10", fg="pink", bg="grey")
b = Entry(root, width=25)
lab2.pack()
b.pack()

button1 = Button (root, text = 'Вычислить', font="Courier 15", background = "red", fg = "pink", command=click1, width=10, height=2)
button1.pack()

lab3 = Label(root, text=" ", font="Courier 10", fg="pink", bg="grey")
lab3.pack()

lab4 = Label(root, text="Задание 2", font="Courier 15", fg="red", bg="grey")
lab4.pack()

lab5 = Label(root, text="Введите значение x:", font="Courier 10", fg="pink", bg="grey")
c = Entry(root, width=25)
lab5.pack()
c.pack()

button2 = Button (root, text = 'Вычислить y', font="Courier 15", background = "red", fg = "pink", command=click2, width=12, height=2)
button2.pack()

```

Практическая работа № 6

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

Инструкция для обучающихся

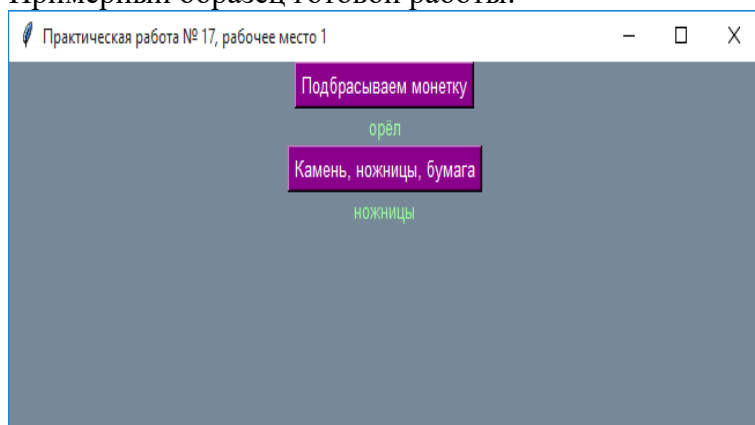
Внимательно прочитайте задание. Создайте программу на языке программирования Python.

Время выполнения задания – 60 минут.

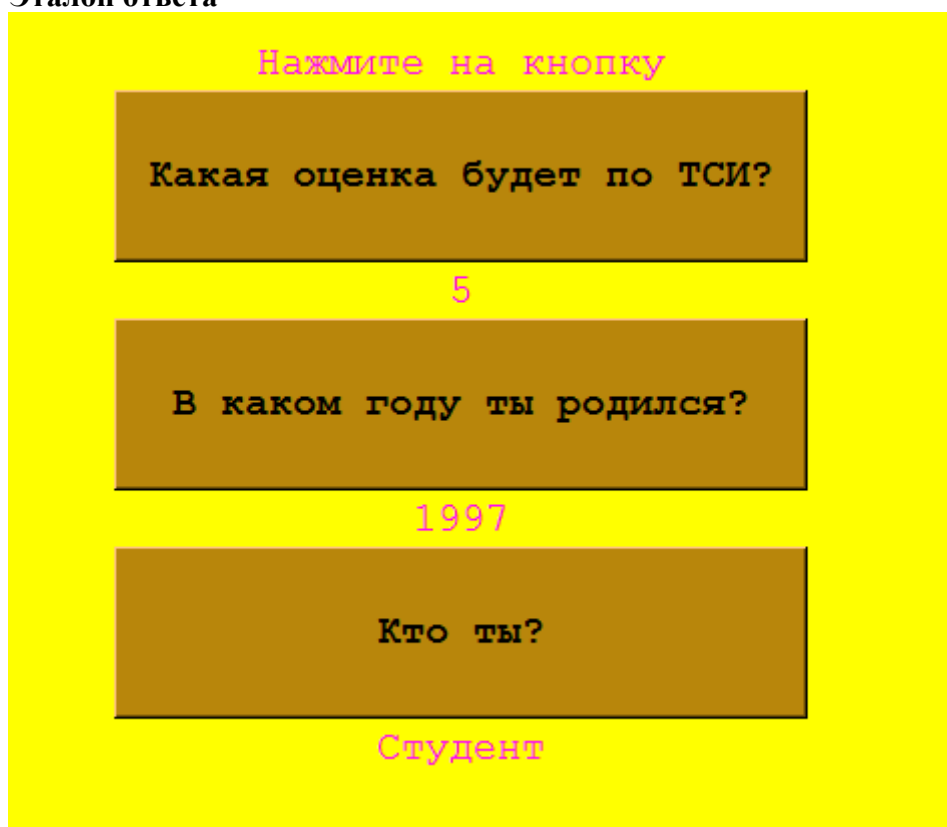
Задание

Написать программу с графическим интерфейсом, используя метки, кнопки, подключить модуль random для генерации случайных значений.

Примерный образец готовой работы:



Эталон ответа



```

from tkinter import *
from math import *
from random import *

def click2():
    try:
        A=[2,3,4,5]
        lab12.config(text='{}'.format(choice(A)))
    except ValueError:
        lab12.config(text="Ошибка!")

def click3():
    try:
        GOD=[1990,1991,1992,1993,1994,1995,1996,1997,1998]
        lab13.config(text='{}'.format(choice(GOD)))
    except ValueError:
        lab13.config(text="Ошибка!")

def click4():
    try:
        NAM=['Гордон Фримен','Джеки Чан','Джон Сина','Джеймс Бонд','Вэтман','Доктор Кто','Робин Гуд','Студент','Школьник']
        lab14.config(text='{}'.format(choice(NAM)))
    except ValueError:
        lab14.config(text="Ошибка!")

root = Tk()
root.title('Практическое занятие №8 7')
root.geometry('600x600')
root["bg"] = "yellow"

lab3 = Label(root, text=" ", font="Courier 40", fg="magenta", bg="yellow")
lab3.pack()

lab4 = Label(root, text="Задание 7", font="Courier 35 bold", fg="magenta", bg="yellow")
lab4.pack()

lab9 = Label(root, text="Нажмите на кнопку", font="Courier 15", fg="magenta", bg="yellow")
lab9.pack()

button2 = Button (root, text = 'Какая оценка будет по ТСИ?', font="Courier 15 bold", background = "#b8860b", fg = "#000000", command=click2, width=28, height=3)
button2.pack()

lab12 = Label(root, text="|?|", font="Courier 15", fg="magenta", bg="yellow")
lab12.pack()

button3 = Button (root, text = 'В каком году ты родился?', font="Courier 15 bold", background = "#b8860b", fg = "#000000", command=click3, width=28, height=3)
button3.pack()

lab13 = Label(root, text="|????|", font="Courier 15", fg="magenta", bg="yellow")
lab13.pack()

```

Практическая работа № 7 «Использование языка программирования Python для создания программ с циклами while и for»

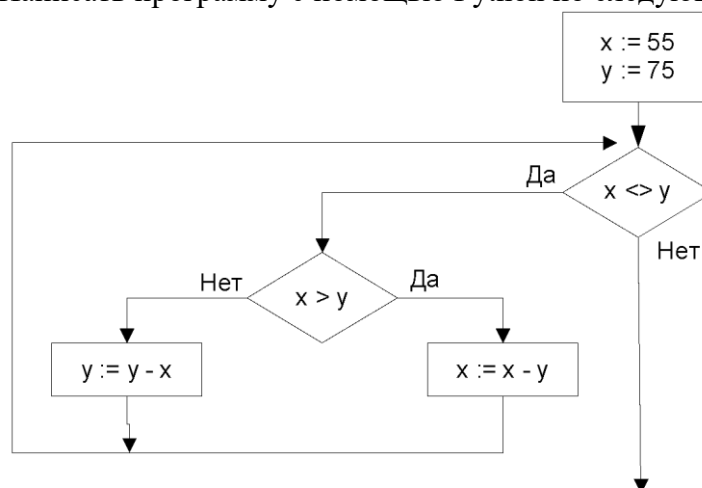
Инструкция для обучающихся

Внимательно прочитайте задание. Создайте программу на языке программирования Python.

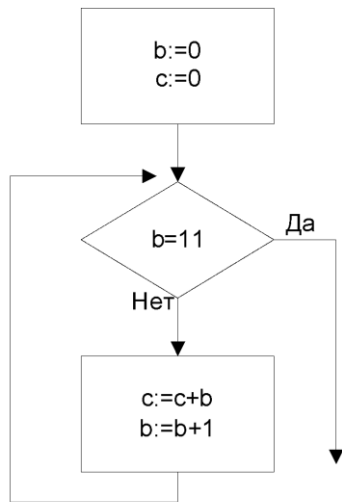
Задание

Вариант № 1

1. Написать программу с помощью Python по следующему алгоритму:

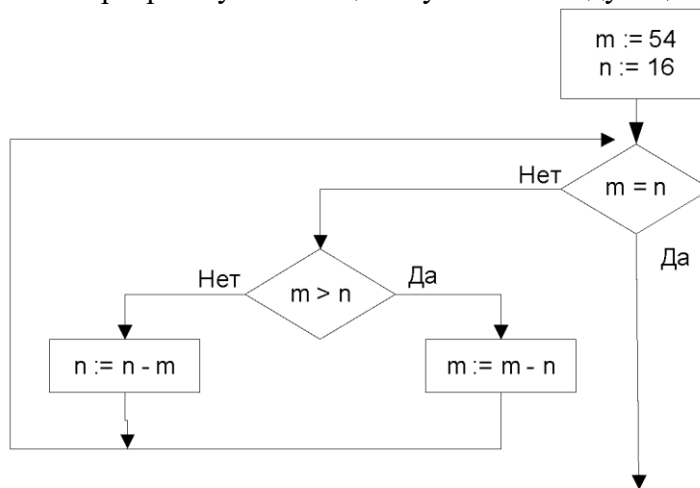


2. Написать программу с помощью Python по следующему алгоритму:

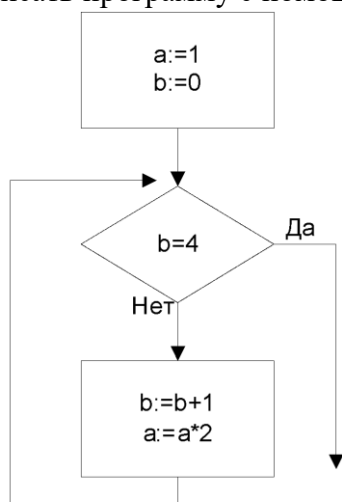


Вариант № 2

1. Написать программу с помощью Python по следующему алгоритму:



2. Написать программу с помощью Python по следующему алгоритму:



Эталон ответа

The image displays three screenshots of Python code editors, each showing a different program. The first screenshot shows a program with variables A and C, and nested if and while loops. The second screenshot shows a program with a for loop and an if-else statement. The third screenshot shows a program with multiple print statements, while and for loops, and conditional logic.

```
PR_18_1.py - D:\clouds\googledisk\tas...
File Edit Format Run Options Window Help
A=-3
C=3
A=A+1
C=C+1
if A > 0:
    while C < 0:
        A=A+1
        C=C+1
else:
    A=A+1
    C=C+1
    while C < 0:
        A=A+1
        C=C+1
print ('Значение A: ', A)
print ('Значение C: ', C)
Ln: 1 Col: 0

PR_18_2.py - D:\clouds\googledisk\tas...
File Edit Format Run Options Window Help
a=2
for b in range(3,0,-1):
    if b>0:
        b=b-1
        a=a*2+1
else:
    print ('Значение a равно: ',a)
|
Ln: 8 Col: 0

Пp18.py - D:\clouds\googledisk\tashavk\2018-2019\1-...
File Edit Format Run Options Window Help
print('Вариант 7:')
print('Задача №1')
f=15
t=35
while f!=t:
    if f>t:
        f=f-t
    else:
        t=t-f
    print('f=',f)
    print('t=',t)
else:
    print('Ответ: f=t=',f)
print('Задача №2')
a=3
b=5
c=0
for x in range(0,15,1):
    if c<14:
        b=a+b
        c=c+3
    print('b=',b)
    print('c=',c)
print('Ответы:')
print('b=',b)
print('c=',c)
Ln: 1 Col: 0
```

Практическая работа № 8

«Использование языка программирования Python для создания программ по работе с файлами и файловой структурой»

Инструкция для обучающихся

Внимательно прочитайте задание. Создайте программу на языке программирования Python.

Время выполнения задания – 60 минут.

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.

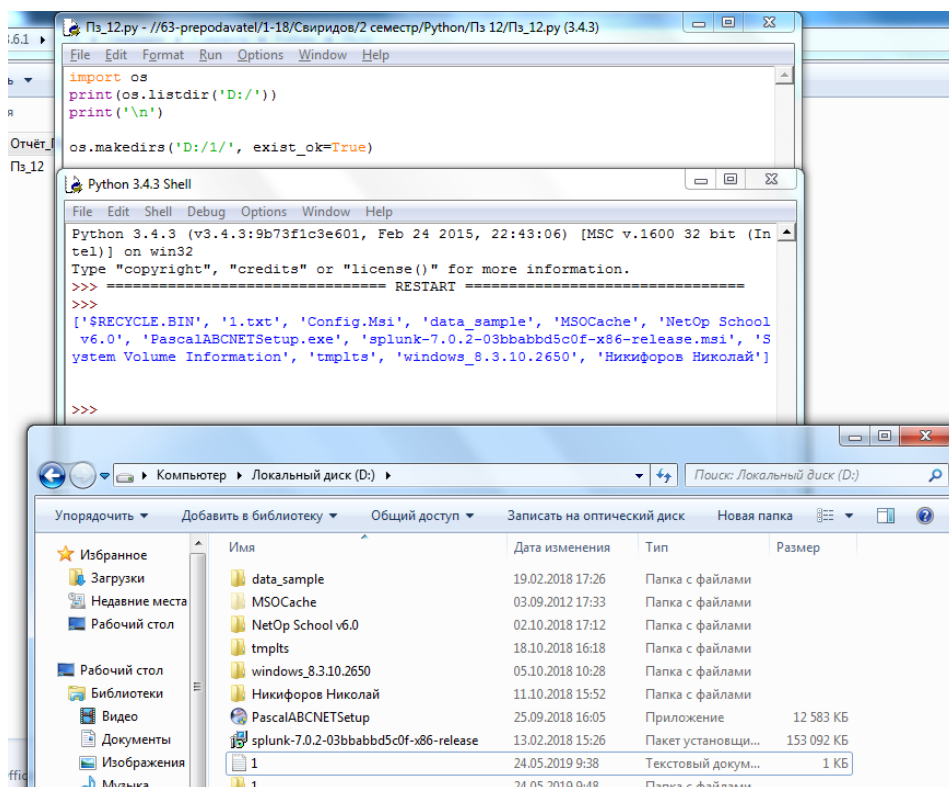
Эталон ответа

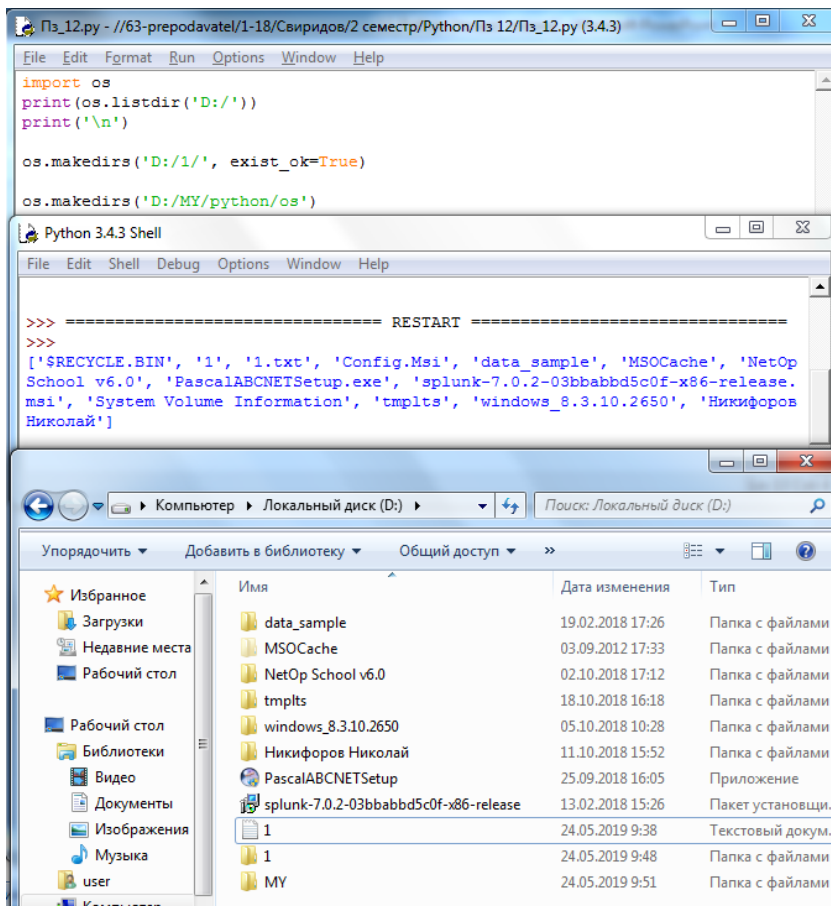
The screenshot shows a Python IDE window titled "Пз_12.py - //63-prepodavatel/1-18/Свиридов/2 семестр/Python/Пз 12/Пз_12.py (3.4.3)". The script contains the following code:

```
import os
print(os.listdir('D:/'))
print('\n')
```

The Python 3.4.3 Shell window shows the output of the script:

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
['$RECYCLE.BIN', '1.txt', 'Config.Msi', 'data_sample', 'MSOCache', 'NetOp School v6.0', 'PascalABCNETSetup.exe', 'splunk-7.0.2-03bbabd5c0f-x86-release.msi', 'System Volume Information', 'tplts', 'windows_8.3.10.2650', 'Никифоров Николай']
>>>
```





Практическая работа № 9 «Использование языка программирования Python для создания программ с использованием классов»

Инструкция для обучающихся

Внимательно прочитайте задание. Создайте программу на языке программирования Python.

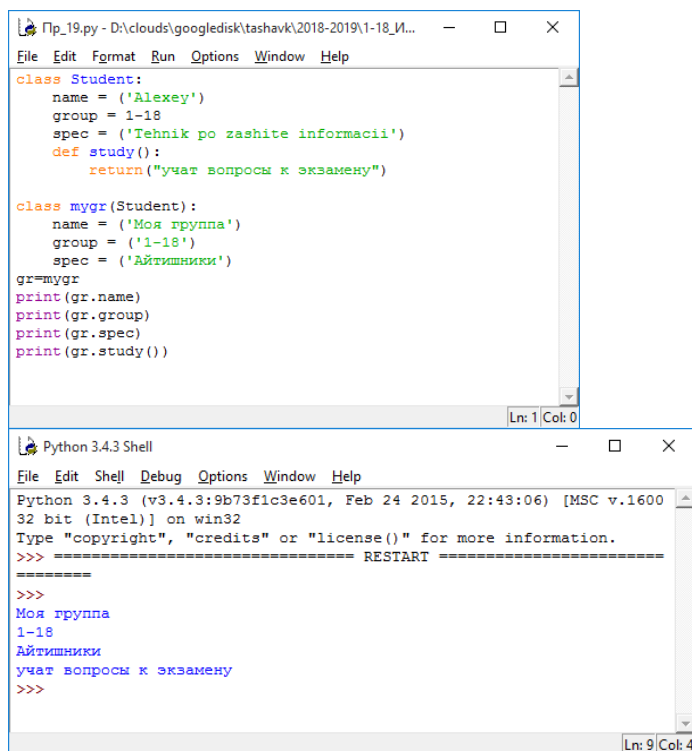
Время выполнения задания – 60 минут.

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 Mmgr, установите атрибуты, вызовите метод study.

Эталон ответа



```
class Student:
    name = 'Alexey'
    group = 1-18
    spec = ('Tehnik po zashite informacii')
    def study():
        return("учат вопросы к экзамену")

class mygr(Student):
    name = ('Моя группа')
    group = ('1-18')
    spec = ('Айтишники')
gr=mygr
print(gr.name)
print(gr.group)
print(gr.spec)
print(gr.study())
```

```
Python 3.4.3 Shell
Python 3.4.3 (v3.4.3:9b73fic3e601, Feb 24 2015, 22:43:06) [MSC v.1600
32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>>
Моя группа
1-18
Айтишники
учат вопросы к экзамену
>>>
```

Практическая работа № 10 «Построение логически правильных и эффективных программ на языке программирования Python»

Инструкция для обучающихся

Внимательно прочитайте задание. Создайте программу на языке программирования Python.

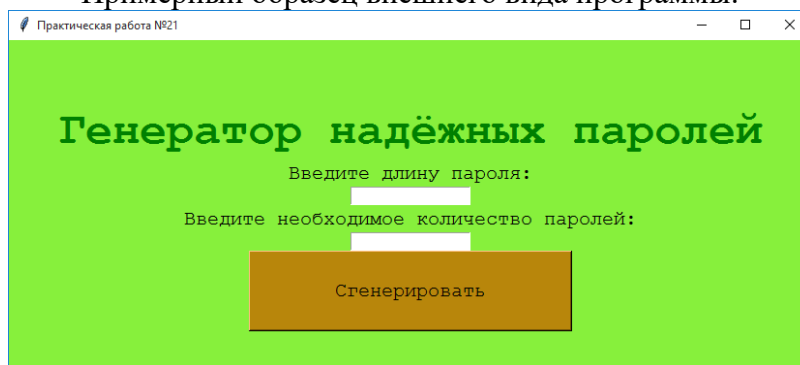
Время выполнения задания – 45 минут.

Задание

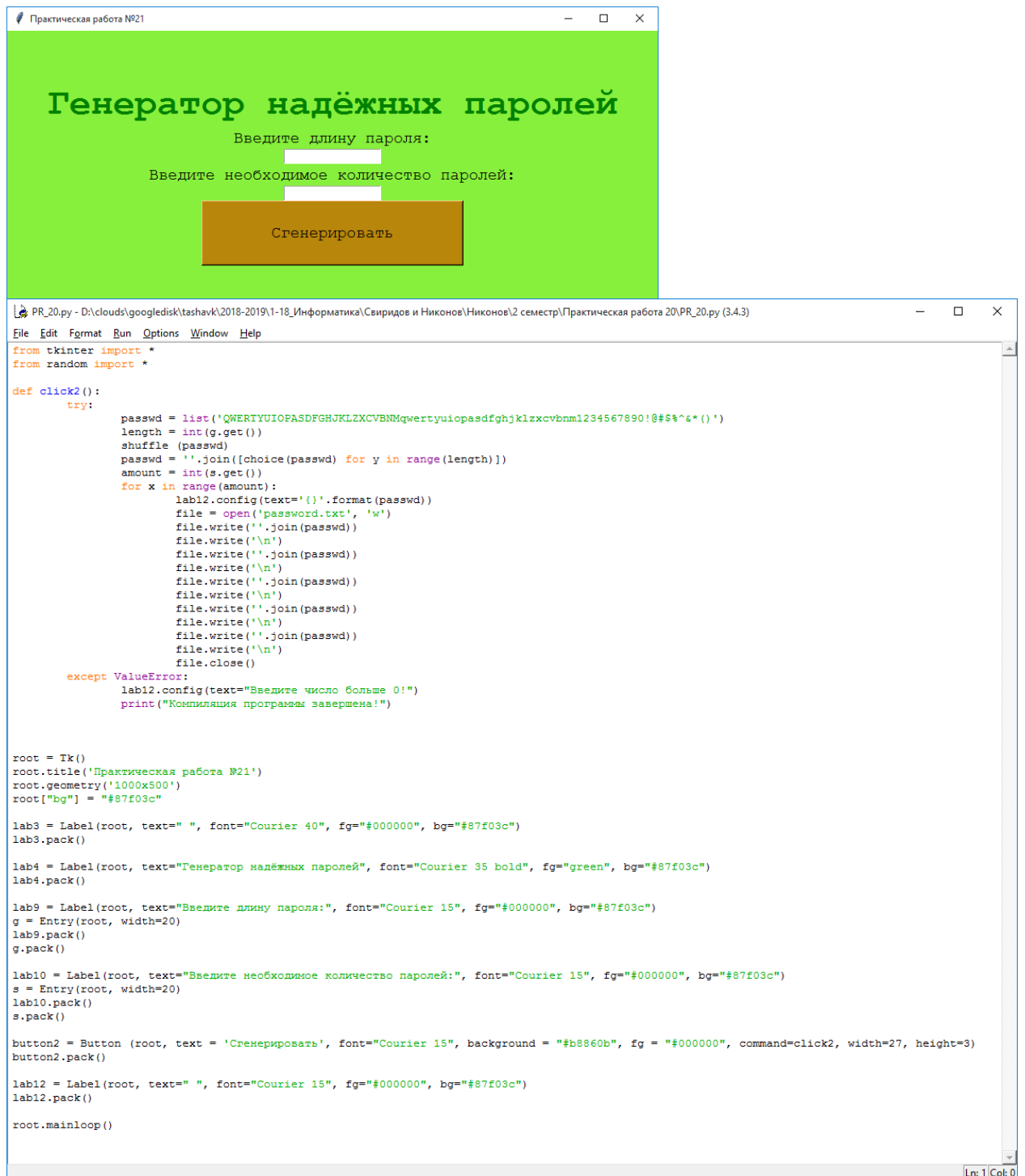
Создать программу Генератор паролей с графическим интерфейсом с запросом о количестве паролей и записью в файл всех сгенерированных паролей.

Допустимые символы: заглавные буквы латиницы, маленькие буквы латиницы, знаки _@#!%&*

Примерный образец внешнего вида программы:



Эталон ответа



Практическая работа № 11

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

Эталон ответа

```
1.
$ sudo apt-get install build-essential checkinstall python3-dev python3-setuptools
$ sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev
$ sudo apt-get install libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev \
,libffi-dev
$ wget https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tgz
$ tar xvf Python-3.7.3.tgz
$ cd Python-3.7.3
$ ./configure --enable-optimizations --enable-loadable-sqlite-extensions
$ sudo make altinstall
2.
$ mkvirtualenv --python=/usr/local/bin/python3.7 pyneng-py3-7
3.
$ sudo pip3.7 install virtualenvwrapper
4.
$ exec bash
5.
$ mkvirtualenv --python=/usr/local/bin/python3.7 pyneng
New python executable in PyNEng/bin/python
Installing distribute.....done.
Installing pip.....done.
(pyneng)$
6.
(pyneng)$ lssitepackages
ANSI.py pexpect-3.3-py2.7.egg-info
ANSI.pyc pickleshare-0.5-py2.7.egg-info
decorator-4.0.4-py2.7.egg-info pickleshare.py
decorator.py pickleshare.pyc
decorator.pyc pip-1.1-py2.7.egg
distribute-0.6.24-py2.7.egg pxssh.py
easy-install.pth pxssh.pyc
fdpexpect.py requests
fdpexpect.pyc requests-2.7.0-py2.7.egg-info
FSM.py screen.py
FSM.pyc screen.pyc
IPython setuptools.pth
ipython-4.0.0-py2.7.egg-info simplegeneric-0.8.1-py2.7.egg-info
```

```

ipython_genutils simplegeneric.py
ipython_genutils-0.1.0-py2.7.egg-info simplegeneric.pyс
path.py test_path.py
path.py-8.1.1-py2.7.egg-info test_path.pyс
path.pyс traitlets
pexpect traitlets-4.0.0-py2.7.egg-info

```

Практическая работа № 13 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.

Эталон ответа

Например, если как аргумент был передан список с одним файлом `sw3_dhcp_snooping.txt`:

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
↔-----					
↔00:E9:BC:3F:A6:50	100.1.1.6	76260	dhcp-snooping	3	↔
↔FastEthernet0/20					
↔00:E9:22:11:A6:50	100.1.1.7	76260	dhcp-snooping	3	↔
↔FastEthernet0/21					
Total number of bindings: 2					

В итоговом csv файле должно быть такое содержимое:

```

switch,mac,ip,vlan,interface
sw3,00:E9:BC:3F:A6:50,100.1.1.6,3,FastEthernet0/20
sw3,00:E9:22:11:A6:50,100.1.1.7,3,FastEthernet0/21

```

Практическая работа № 14 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).

Эталон ответа

```
from pprint import pprint
import yaml
from netmiko import (
    ConnectHandler,
    NetmikoTimeoutException,
    NetmikoAuthenticationException,
)
def send_show_command(device, commands):
    result = {}
    try:
        with ConnectHandler(**device) as ssh:
            ssh.enable()
        for command in commands:
            output = ssh.send_command(command)
            result[command] = output
    return result
```

Практическая работа № 20 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...

Эталон ответа

```

$ python add_data.py
Добавляю данные в таблицу switches...
При добавлении данных: ('sw1', 'London, 21 New Globe Walk') Возникла ошибка:
↳UNIQUE constraint failed: switches.hostname
При добавлении данных: ('sw2', 'London, 21 New Globe Walk') Возникла ошибка:
↳UNIQUE constraint failed: switches.hostname
При добавлении данных: ('sw3', 'London, 21 New Globe Walk') Возникла ошибка:
↳UNIQUE constraint failed: switches.hostname
Добавляю данные в таблицу dhcp...
При добавлении данных: ('00:09:BB:3D:D6:58', '10.1.10.2', '10', 'FastEthernet0/1',
↳ 'sw1') Возникла ошибка: UNIQUE constraint failed: dhcp.mac
При добавлении данных: ('00:04:A3:3E:5B:69', '10.1.5.2', '5', 'FastEthernet0/10',
↳ 'sw1') Возникла ошибка: UNIQUE constraint failed: dhcp.mac
При добавлении данных: ('00:05:B3:7E:9B:60', '10.1.5.4', '5', 'FastEthernet0/9',
↳ 'sw1') Возникла ошибка: UNIQUE constraint failed: dhcp.mac
При добавлении данных: ('00:07:BC:3F:A6:50', '10.1.10.6', '10', 'FastEthernet0/3',
↳ 'sw1') Возникла ошибка: UNIQUE constraint failed: dhcp.mac
При добавлении данных: ('00:09:BC:3F:A6:50', '192.168.100.100', '1',
↳ 'FastEthernet0/7', 'sw1') Возникла ошибка: UNIQUE constraint failed: dhcp.mac

```


**3.2. Контрольно-оценочные материалы
для промежуточной аттестации по дисциплине « Основы алгоритмизации и
программирования»**

Промежуточная аттестация проводится в форме экзамена.

Перечень экзаменационных заданий:

№ п/п	Перечень практических заданий
1.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} \operatorname{tg}x - x^2, & \text{если } x < 2 \\ x^2, & \text{если } 2 \leq x < 4 \\ x, & \text{если } x \geq 4 \end{cases}$
2.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} \sin x + x, & \text{если } x < 0 \\ x^3, & \text{если } 0 \leq x \leq 10 \\ x^2 - 2, & \text{если } x > 10 \end{cases}$
3.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} 1x + x^3, & \text{если } x < 3 \\ \operatorname{ctg}x, & \text{если } 3 \leq x < 5 \\ x, & \text{если } x \geq 5 \end{cases}$
4.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x - 2, & \text{если } x > 0 \\ 3 - x, & \text{если } -3 \leq x \leq 0 \\ x^2, & \text{если } x < -3 \end{cases}$
5.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $x=f(y)$</u> . x — вводимое значение. $y = \begin{cases} \sin x + \operatorname{tg}x, & \text{если } x > 2 \\ x^4 - 2, & \text{если } 0 < x \leq 2 \\ x, & \text{если } x \leq 0 \end{cases}$
6.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x - x^2, & \text{если } x < 0 \\ x - 1, & \text{если } 0 \leq x \leq 3 \\ 1x + x, & \text{если } x > 3 \end{cases}$
7.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x - 3x + x^2, & \text{если } x < 6 \\ \sin x + x^3, & \text{если } 6 \leq x < 8 \\ x^4, & \text{если } x \geq 8 \end{cases}$
8.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} 5x^2 - 4x + 1, & \text{если } x > 7 \\ x^3 - x, & \text{если } 0 < x \leq 7 \\ x + 3x, & \text{если } x \leq 0 \end{cases}$
9.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x^2 + \operatorname{tg}x, & \text{если } x < 9 \\ x^3, & \text{если } 9 \leq x < 12 \\ x + x^3 - 1, & \text{если } x \geq 12 \end{cases}$
10.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} \cos x + x, & \text{если } x > 0 \\ x^5, & \text{если } -2 \leq x \leq 0 \\ x^3 - 3, & \text{если } x < -2 \end{cases}$

11.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x^2 - 3x + 12, & \text{если } x < 0 \\ 12 - 3x + x, & \text{если } 0 \leq x \leq 7 \\ x + 12 - 3x, & \text{если } x > 7 \end{cases}$
12.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x + 3x^2, & \text{если } x < 1 \\ 4x^2 - x, & \text{если } 1 \leq x \leq 3 \\ x - 1, & \text{если } x > 3 \end{cases}$
13.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} x - 5, & \text{если } x > 0 \\ 5 - x, & \text{если } -3 \leq x \leq 0 \\ x, & \text{если } x < -3 \end{cases}$
14.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} 1x - 2x^3, & \text{если } x > 1 \\ 3x^3 - 1x^2, & \text{если } -1 \leq x \leq 1 \\ x - x, & \text{если } x < -1 \end{cases}$
15.	Написать программу на языке Python с графическим интерфейсом (кнопка, метка, пустое поле для ввода), <u>вычисляющую значение функции $y=f(x)$</u> . x — вводимое значение. $y = \begin{cases} 10x^2 - 3x^3 + x, & \text{если } x < 0 \\ x^4 - x + 2, & \text{если } 0 \leq x \leq 5 \\ 2x^2 - 10 + x^3, & \text{если } x > 5 \end{cases}$

Критерии оценки практических заданий

«Отлично» – студент владеет системой знаний и представлений по предложенной в задаче проблеме; выбор способов решения задачи грамотный; рассуждения носят аргументированный характер; предложенные способы решения задачи имеют профессиональную направленность; технологические решения носят осознанный характер; студент умеет проектировать технологический процесс; проявляет творческий подход к решению поставленных задач, грамотно выполняет расчеты, отсутствуют ошибки в графической части.

«Хорошо» – студент владеет системой знаний и представлений по предложенной в задаче проблеме; в выборе способов решения задачи допускает незначительные неточности, рассуждения аргументированы; проектировочно-технологические решения носят осознанный характер; в расчетах и графической части имеются незначительные погрешности, неточности.

«Удовлетворительно» – знания и представления студента по предложенной в задаче проблеме носят разрозненный характер; в выборе способов решения задачи допущены ошибки; проектировочные и технологические решения носят ограниченный, репродуктивный характер.

«Неудовлетворительно» – студент имеет существенные пробелы в знаниях и представлениях по предложенной в задаче проблеме; при выборе способов решения задачи допущены ошибки; рассуждения бездоказательны.