

ИТОГОВАЯ РАБОТА К МОДУЛЮ 4

ИНСТРУКЦИЯ

1. Итак, для того, чтобы реализовать поставленную задачу, нужно написать программу, за основу которой использовать уже написанную программу проверки надежности пароля и добавить к ней модуль (функцию) для генерации предложенного пароля в случае, если указанный пароль не удовлетворяет указанным требованиям.

Примечание:

- a. Требования к паролю: более 8 символов в длину И содержать в себе 3 и более спецсимволов/цифр.
- b. Генерируемая случайная последовательность из набора: 10 символов, буквы верхнего и нижнего регистра + цифры + спецсимволы

2. Чтобы реализовать эту программу, модернизируем одну из уже написанных следующим образом:

```
import re
import random # импорт стандартных библиотек регулярных
выражений, а также для выбора случайных символов

def check_symbols(value): # функция проверки символов
    symbols_pattern = r"<>,*&^%$#@!-=+'0-9]" # искомые
символы для проверки
    symbols_found = re.findall(symbols_pattern, value) #
Поиск по слову
    if len(symbols_found) > 2: #если 3 и более символов,
то требования удовлетворены
        return True
    else:
        return False

def generate_password(value): #функция генерации
случайного пароля
    letters =
'+-/*!&$#?=@<>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
STUVWXYZ1234567890'
    value = ''
    for i in range(10): #длина 10, символы из указанного
набора
        value += random.choice(letters)
```

```

    return value

def validate_password(value): # функция проверки длины +
    # проверки по функции символов
    password_length = len(value) # получение длины пароля
    if password_length > 8 and check_symbols(value): #
        # проверка соответствия двум условиям
        print("Your password is quite good! You can use
it.")
    else:
        good_pass = generate_password(value) #если
        # проверка не пройдена, генерируем новый пароль
        print(f"Your password is not good enough... May be
try this one: {good_pass}") # вывод рекомендации

value = input('Enter a password to check: ')
validate_password(value)

```

Снимок экрана для проверки отступов:

```

1  import re
2  import random
3
4
5  1 usage
6  def check_symbols(value): # функция проверки символов
7      symbols_pattern = r"[<>,*&^$#@!-=+'0-9]" # искомые символы для проверки
8      symbols_found = re.findall(symbols_pattern, value) # Поиск по слову
9      if len(symbols_found) > 2:
10         return True
11     else:
12         return False
13
14  1 usage
15  def generate_password(value):
16      letters = '+-/*!&$#?=@<>abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890'
17      value = ''
18      for i in range(10):
19         value += random.choice(letters)
20      return value

```

```

18     value += random.choice(letters)
19     return value
20
21
22 1 usage
23 def validate_password(value): # функция проверки длины + проверки по функции символов
24     password_length = len(value) # получение длины пароля
25     if password_length > 8 and check_symbols(value): # проверка соответствия двум условиям
26         print("Your password is quite good! You can use it.")
27     else:
28         good_pass = generate_password(value)
29         print(f"Your password is not good enough... May be try this one: {good_pass}")
30
31 value = input('Enter a password to check: ')
32 validate_password(value)
33

```

3. Далее нужно протестировать работу программы и для ответа прикрепить варианты ее выполнения. Нужно выполнить проверку для “хорошего” пароля и “плохого”:

```

/Users/ /PycharmProjects/pythonProject18/venv/bin/python /Users/
Enter a password to check: psswd2chek!7
Your password is quite good! You can use it.

Process finished with exit code 0

```

```

/Users/ /PycharmProjects/pythonProject18/venv/bin/python /Users/
Enter a password to check: verygoodpassword11
Your password is not good enough... May be try this one: 4!9+@d7FnF

Process finished with exit code 0

```

