**Основы информационной безопасности. Обучение в записи**

**Урок 10. Семинар: Безопасная разработка приложений**

Оглавление

[Задание 1 2](#_Toc200154783)

[Задание 2 4](#_Toc200154784)

[Задание 3 6](#_Toc200154785)

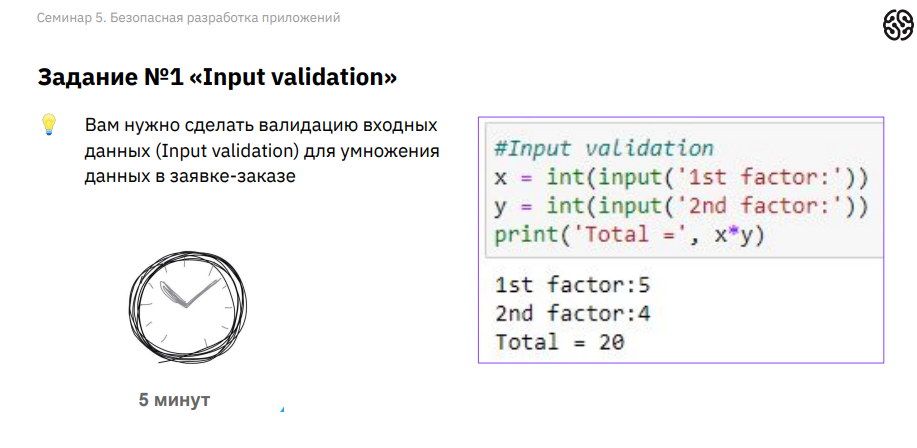
[Задание 4 8](#_Toc200154786)

[Задание 5 9](#_Toc200154787)

[Задание 6 10](#_Toc200154788)

[Домашнее задание: 11](#_Toc200154789)

### Задание 1



*def error\_checking():*

*while type:*

*x = input('1st factor: ')*

*y = input('2st factor: ')*

*try:*

*x = int(x)*

*y = int(y)*

*while (x <= 0 or y <= 0 or x > 1000 or y > 1000):*

*print('Заказ не может быть меньше/равен 0 или больше 1000. Повторите ввод:')*

*x = int(input('1st factor: '))*

*y = int(input('2st factor: '))*

*except ValueError:*

*print('Вы ввели некорректное значение '*

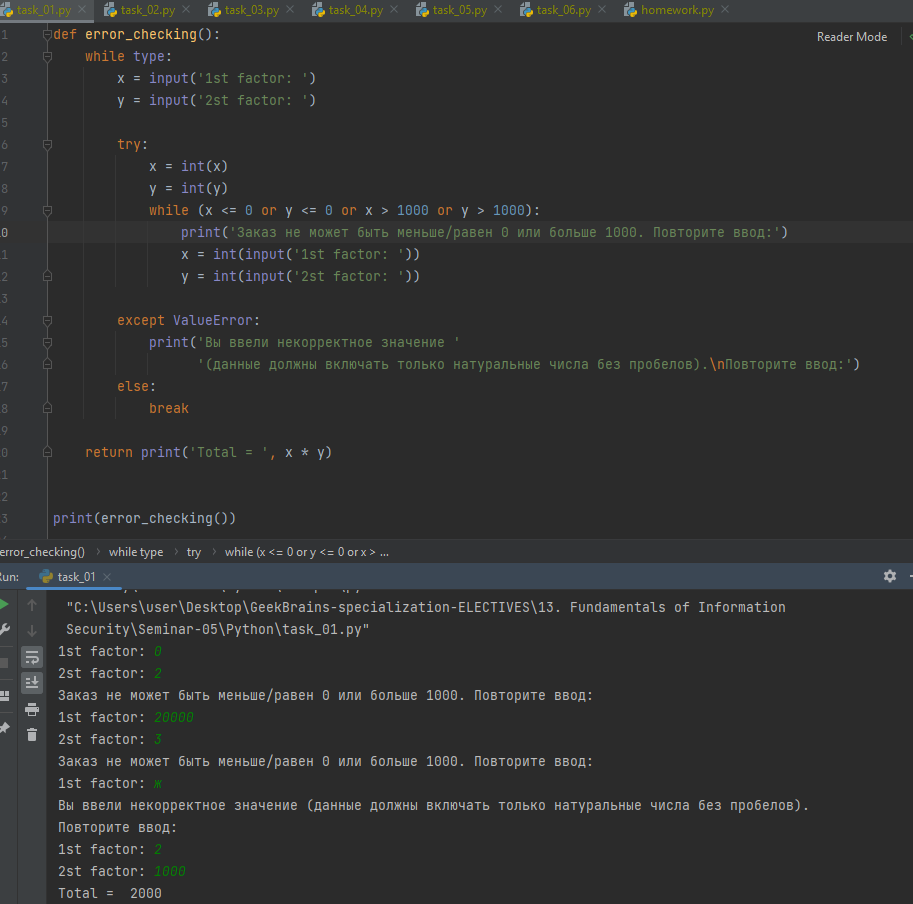
*'(данные должны включать только натуральные числа без пробелов).\nПовторите ввод:')*

*else:*

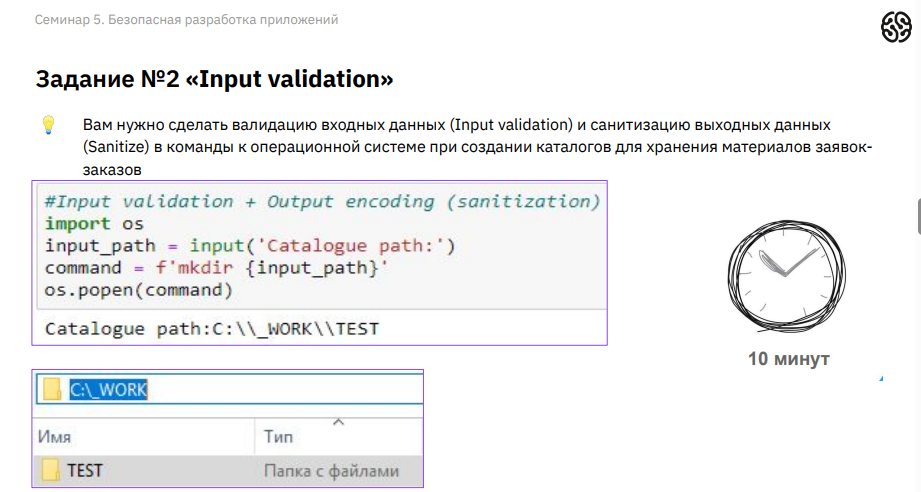
*break*

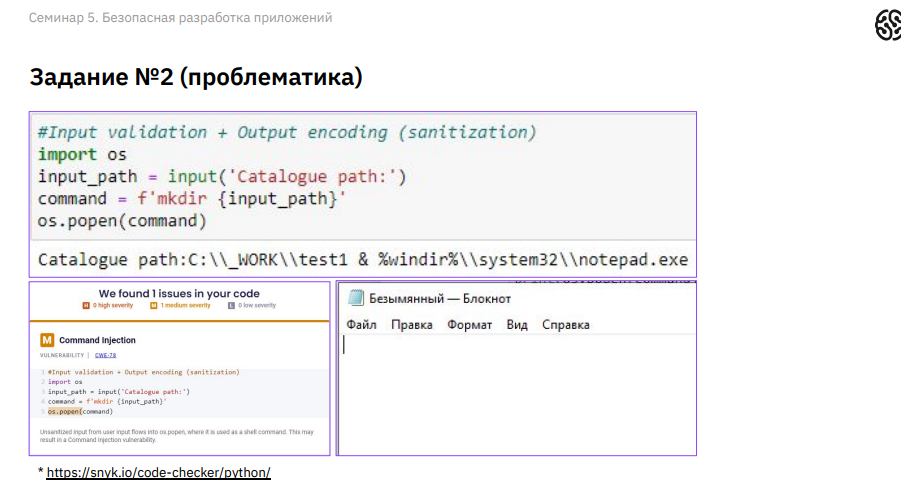
*return print('Total = ', x \* y)*

*print(error\_checking())*



### Задание 2





*import os*

*input\_path = input('Catalog path: ')*

*spec\_symbols = ['\*', '?', "<", '>', '|', '&']*

*check = [characters in input\_path for characters in spec\_symbols]*

*while True in check:*

*for i in range(len(check)):*

*check[i] = False*

*print('Incorect catalogue path')*

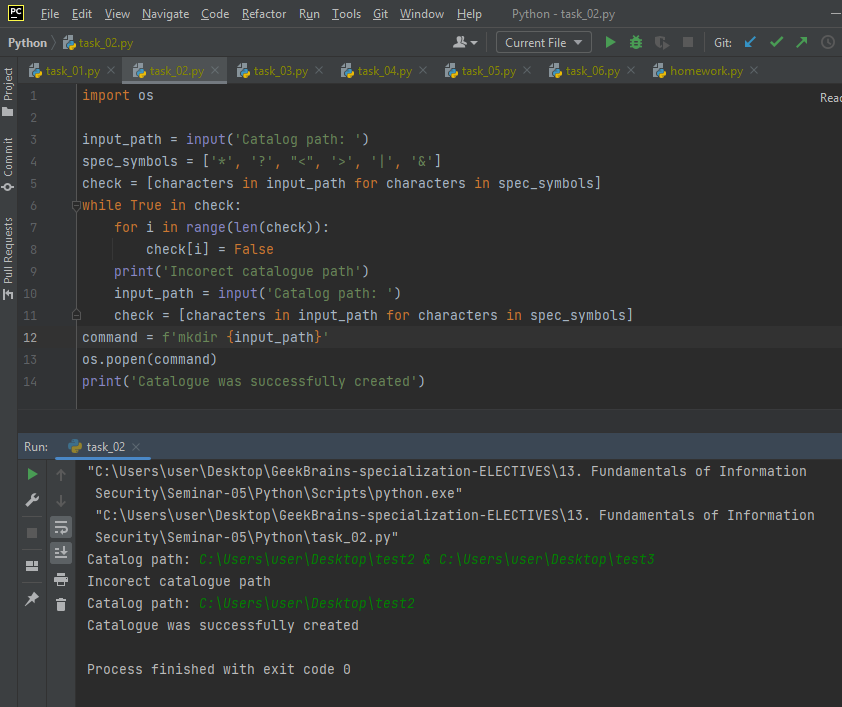
*input\_path = input('Catalog path: ')*

*check = [characters in input\_path for characters in spec\_symbols]*

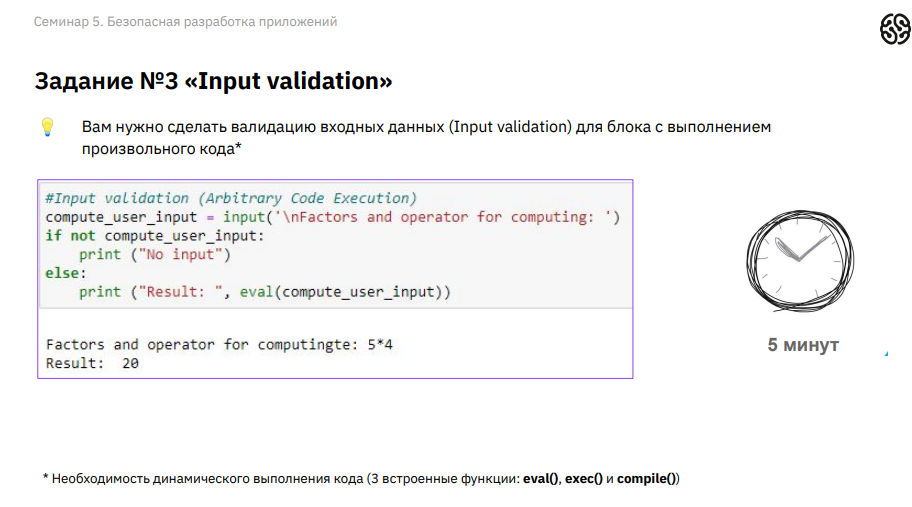
*command = f'mkdir {input\_path}'*

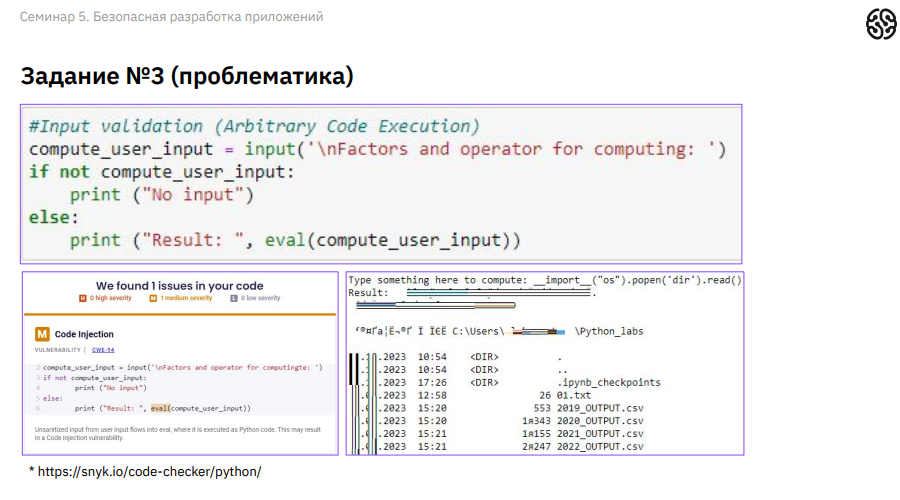
*os.popen(command)*

*print('Catalogue was successfully created')*



### Задание 3





compute\_*user\_input = input('\nFactors and operator for computing: ')*

*if not compute\_user\_input:*

*print('No input')*

*else:*

*print('Result: ', eval(compute\_user\_input, {'\_buitins\_':{}}))*

*'''*

*Предупреждение:*

*Использование eval() может быть небезопасно, особенно при обработке пользовательских вводов,*

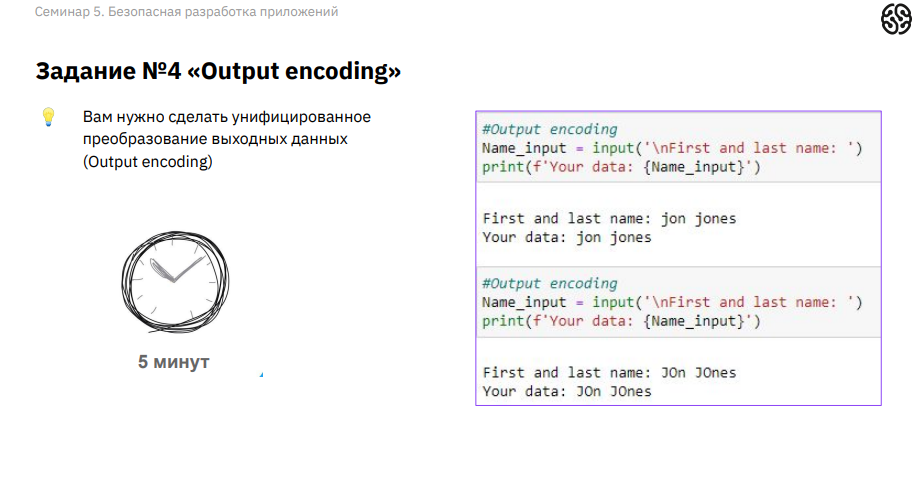
*так как пользователь может ввести строку кода, которая может выполнить произвольный код на вашей машине. Поэтому рекомендуется быть осторожным и избегать использования eval() при обработке пользовательских вводов, если вы не уверены в их безопасности.*

*Альтернативы eval():*

*Если вам нужно выполнить код, который не является выражением, то лучше использовать exec(), который позволяет выполнять произвольный код, но не возвращает результат. Если вам нужно скомпилировать строку кода и выполнить ее позже, то можно использовать функцию compile(), которая компилирует строку кода в объект кода, который можно затем выполнить с помощью exec() или eval().*

*'''*

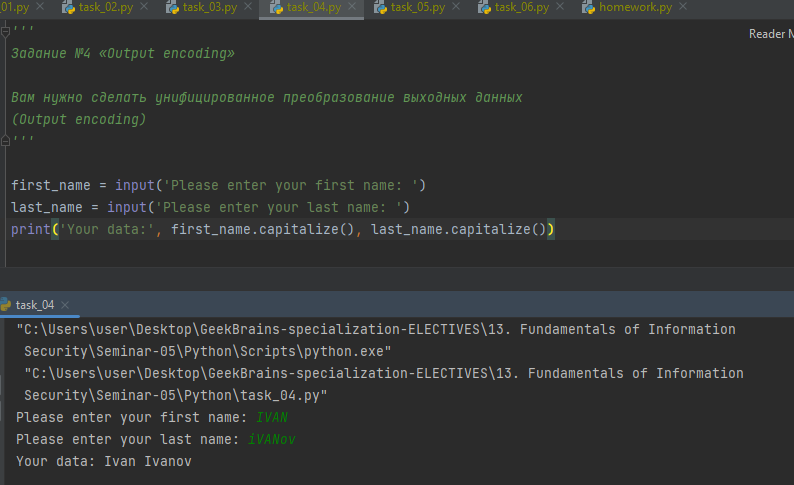
### Задание 4



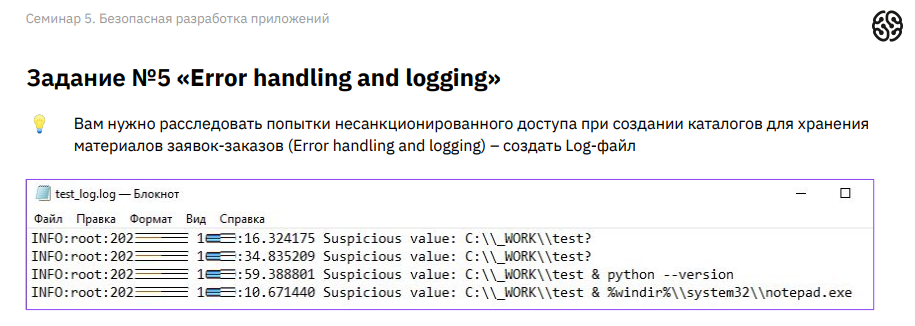
*first\_name = input('Please enter your first name: ')*

*last\_name = input('Please enter your last name: ')*

*print('Your data:', first\_name.capitalize(), last\_name.capitalize())*

**

### Задание 5



*import os*

*import datetime*

*import logging*

*logging.basicConfig(level=logging.DEBUG, filename='test\_log.log', filemode='w')*

*input\_path = input('Catalogue path: ')*

*spec\_symbols = ['\*', '?', "<", '>', '|', '&']*

*check = [characters in input\_path for characters in spec\_symbols]*

*while True in check:*

*for i in range(len(check)):*

*check[i] = False*

*logging.info(f'{datetime.datetime.now()} Suspicious value: {input\_path}')*

*print('Incorect catalogue path')*

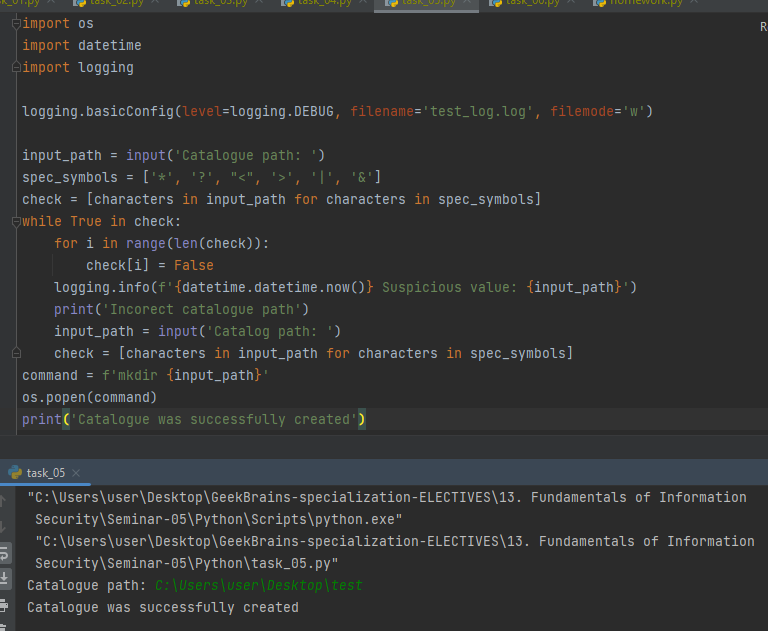
*input\_path = input('Catalog path: ')*

*check = [characters in input\_path for characters in spec\_symbols]*

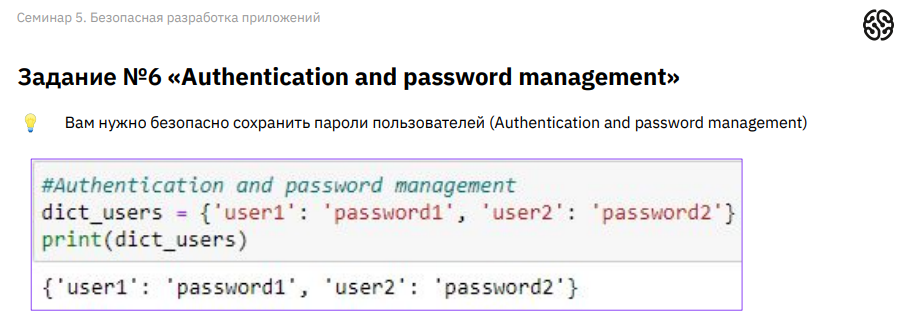
*command = f'mkdir {input\_path}'*

*os.popen(command)*

*print('Catalogue was successfully created')*



### Задание 6



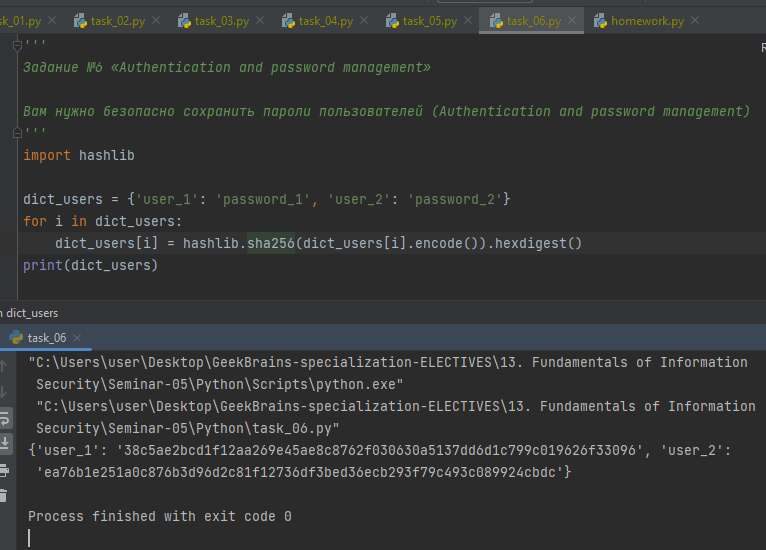
*import hashlib*

*dict\_users = {'user\_1': 'password\_1', 'user\_2': 'password\_2'}*

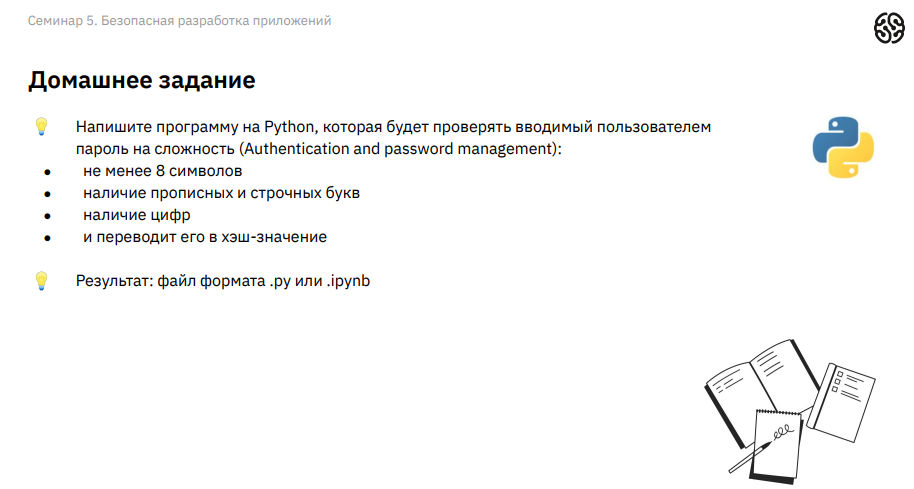
*for i in dict\_users:*

*dict\_users[i] = hashlib.sha256(dict\_users[i].encode()).hexdigest()*

*print(dict\_users)*



### Домашнее задание:



*import hashlib*

*import re*

*def password\_check(password):*

*count = 0*

*for char in password:*

*if 'а' <= char <= 'я' or 'А' <= char <= 'Я':*

*print('Пароль должен содержать только латинские буквы!')*

*count += 1*

*break*

*if not re.search(r'\d', password):*

*print('Пароль должен содержать не менее одного числа!')*

*count += 1*

*if len(password) < 8:*

*print('Длина пароля должна быть не меньше 8 символов!')*

*count += 1*

*if not re.search(r'[A-Z]', password):*

*print('Пароль должен содержать не менее одной прописной буквы!')*

*count += 1*

*if not re.search(r'[a-z]', password):*

*print('Пароль должен содержать не менее одной строчной буквы!')*

*count += 1*

*if count > 0:*

*return '\n------------------------------------------------\n' \*

*'Пароль НЕ соответствует требованиям надежности!\n' \*

*'------------------------------------------------'*

*else:*

*print('Сложность пароля удовлетворяет требованиям сервиса!\nХэш-значение пароля: ')*

*return hashlib.sha256(password.encode()).hexdigest()*

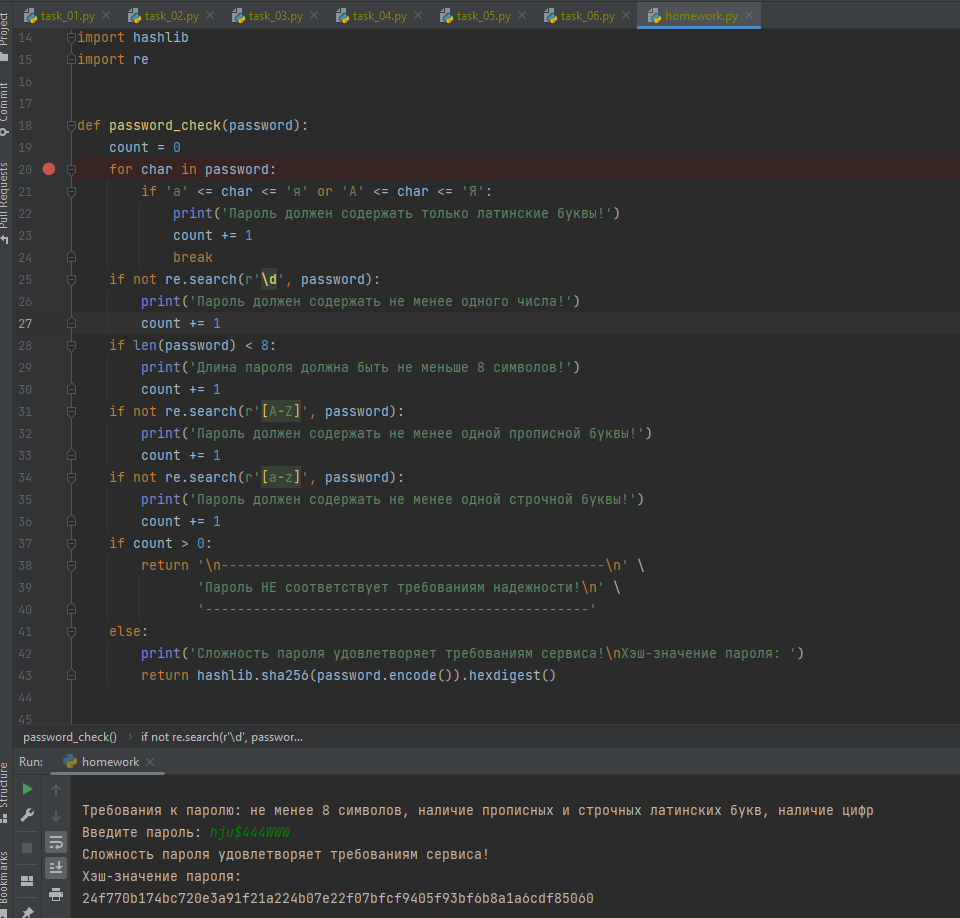
*passw = input('\nТребования к паролю: '*

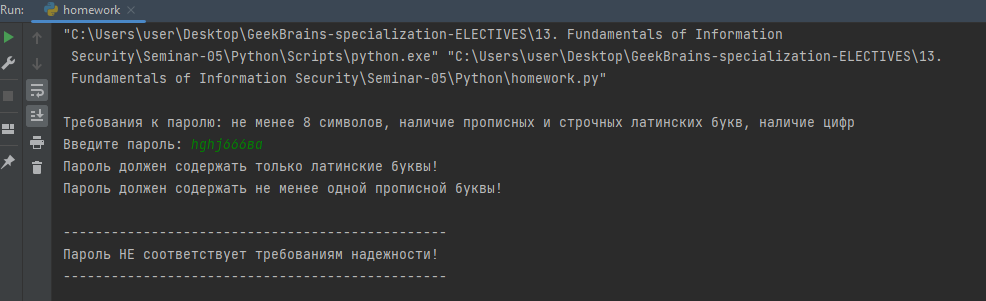
*'не менее 8 символов, наличие прописных и строчных латинских букв, '*

*'наличие цифр'*

*'\nВведите пароль: ')*

*print(password\_check(passw))*





Ссылка на репозиторий:

<https://github.com/olgashenkel/GeekBrains-specialization-ELECTIVES/tree/main/13.%20Fundamentals%20of%20Information%20Security>