Prova d'accés a App2U

Pregunta 2

"Com pot ser? primer diu que el procés ha anat bé i després diu que ha tingut errors... i realment, el procés ha anat bé, perquè ha descarregat els dos fitxers que volíem. Saps trobar on està el BUG que fa que aparegui el missatge d'error? Guarda una còpia del projecte un cop solucionat."

El programa crea dues operacions. Una del tipus GetFilesSFTPOperation i l'altra GetFilesHTTPOperation. Al crear un objecte de la classe OperationController i cridar la funció start, aquesta crida la funció execute_next_operation(). Aquesta funció fa les operacions que li haguem passat en el constructor de la classe OperationController i te en compte dos tipus d'operació: GetFilesSFTPOperation i GetFilesHTTPOperation. Si no es de cap d'aquests dos tipus llença un error.

O AIXÒ HAURIA!! Perquè de la manera en la que esta escrita el codi, el que comprova es que si es del tipus SFTP es descarrega. Si es del tipus HTTP també ho descarrega i, si no es HTTP, llença un error. Es per això que apareix un error quan es fa una operació del tipus SFTP, perquè tot hi que ho pugui executar, despres entra dins les altres condicions if-else. Per a evitar això, hem de fer que aquest seguit d'instruccions sigui un conjunt. Es pot fer de la següent manera:

```
if isinstance(operation, GetFilesSFTPOperation):
    operation.set_controller(self)
    operation.download_from_sftp()
elif isinstance(operation, GetFilesHTTPOperation):
    operation.set_controller(self)
    operation.download_from_web()
else:
    self.operations_error()
```

Amb una estructura elif, saltarà l'excepció en cas que no sigui ni SFTP ni HTTP i no només en una.

Pregunta 3

"...el nostre controlador (la classe OperationController) importa els diferents tipus d'operacions que existeixen i depen d'aquestes per poder funcionar. Pots modificar el codi del projecte per evitar aquest problema fent servir els principis de la orientació a objectes?"

Oi tant que puc! Un controlador només hauria de ser l'encarregat de rebre i enviar ordres però sense tenir en compte la naturalesa d'aquesta. Una possible proposta seria crear un objecte anomenat Operation que fos l'encarregat de gestionar quin tipus d'operacions es fan segons el tipus d'instància.

```python
class Operation:
    def __init__(self, operation) -> None:
        self.operation = operation

    def execute(self, controller) -> None:
        from operation_controller.operations.get_files_http_operation import GetFilesHTTPOperation
        from operation_controller.operations.get_files_sftp_operation import GetFilesSFTPOperation

        if isinstance(self.operation, GetFilesSFTPOperation):
            self.operation.set_controller(controller)
            self.operation.download_from_sftp()
            return 0
        elif isinstance(self.operation, GetFilesHTTPOperation):
            self.operation.set_controller(controller)
            self.operation.download_from_web()
            return 0
        else:
            return -1
```

D'aquesta manera, el controlador només hauria de cridar la funció execute de cada operació:

```python
def execute_next_operation(self) -> None:

    if self.current_operation_index < len(self.operations):
        op = Operation(self.operations[self.current_operation_index])
        response = op.execute(self)

        if response == -1:
            self.operations_error()
    else:
        self.operations_completed()
```

Amb aquesta proposta de solució ja quedaria diferenciada la part del controlador amb la part que gestiona el tipus d'operació segons la seva naturalesa.

Cal anotar que es podrien fer diferents classes heretades de la classe Operation segons el tipus d'operació i així seria una arquitectura més escalable on les instruccions a cada classe serien específiques al tipus d'operació.

Pregunta 4

"Ens agradaria que l'operation controller printi per pantalla el nom de la operació que s'està executant dins del mètode execute_next_operation. També que el printi quan finalitza la operació o en cas d'error."

Per a mostrar per pantalla el nom de l'operació he creat un mètode dins la classe Operation que mostra la informació d'aquesta:

```python
def operation_info(self) -> None:
    from operation_controller.operations.get_files_http_operation import GetFilesHTTPOperation
    from operation_controller.operations.get_files_sftp_operation import GetFilesSFTPOperation

    if isinstance(self.operation, GetFilesSFTPOperation):
        print('Download from %s...' % self.operation.host)
    elif isinstance(self.operation, GetFilesHTTPOperation):
        print('Download from %s...' % self.operation.base_url)
    else:
        print('Unknown operation')
```

D'aquesta manera mostrem de quina URL volem descarregar la informació.

Per altra banda, per a mostrar el codi d'error, si ens fixem en els fitxers get_files_http_operation.py i get_files_sftp_operation.py, veiem que utilitza el mètode try-except a l'hora de descarregar el fitxer de la URL. Veiem que en l'except ja es genera un missatge que crida el mètode operation_error de la classe OperationController. Així dons, per a mostrar el codi d'error si la descarrega falla només haurem d'actualitzar el mètode operation_error per a mostrar el missatge d'error. Aquest missatge d'error ja mostra quina operació és la que ha donat error.

```python
def operation_error(self, operation, message: str) -> None:
    self.operations_error()

    # Show error
    print(message)
```

Pregunta 5

"Si et fixes, les operacions depenen del controller perquè un cop acaben d'executar-se el notifiquen per indicar-li el seu resultat. Què passaria si per exemple les operacions s'executessin des de un altre lloc diferent al controller? com podrien notificar a aquest altre lloc sense haver de modificar el seu codi? Se t'acut com modificar el codi de les operacions per tal d'evitar aquesta dependència (import a OperationController) i que al mateix temps tot segueixi funcionant igual que abans?"

Com s'ha esmentat abans, tant la classe GetFilesHTTPOperation com la classe GetFilesSFTPOperation, a l'hora de descarregar informació de la web, es fa a través de l'estructura try-except. Aquesta estructura dona la possibilitat de recuperar l'error que ha sorgit en l'execució del codi i poder-lo enviar a altres llocs. Fins ara aquest error era passat al controller mitjançant el mètode operation_error de la classe OperationController. Per a evitar l'ús d'aquest, el que es podria fer seria que la funció retornés el si la sortida ha sigut exitosa o no i que cada classe que cridés les funcions download_from_web o download_from_sftp s'encarreguessin de rebre i gestionar la resposta.

La meva proposta de solució és la següent:

```
def download_from_web(self) -> None:
    # Build url
    url = '/'.join([self.base_url, self.file_name])
    try:
        # Get file
        print('Downloading file from %s...' % url)
        request.urlretrieve(url, os.path.join(
            self.local_path, self.file_name))

        # Send success message
        return 'Success downloading file from %s completed' % url

    except (IOError, ValueError, AttributeError):

        # Send error message
        message = 'Error downloading the file %s\n:%s' % (
            self.file_name, traceback.format_exc())
        return message
```

```
def download_from_sftp(self) -> None:
    # Download and delete all the files from remote SFTP
    sftp = None
    try:
        print('Downloading input files from SFTP')
        cnopts = pysftp.CnOpts()
        cnopts.hostkeys = None
        sftp = pysftp.Connection(
            host=self.host, username=self.user_name, password=self.password, cnopts=cnopts)
        if self.remote_path:
            sftp.cwd(self.remote_path)
        remote_files = sftp.listdir()

        for file_name in remote_files:
            if not sftp.isfile(file_name):
                continue
            # Save the file
            file_path = os.path.join(self.local_path, file_name)
            print('Downloading %s...' % file_name)
            sftp.get(file_name, file_path)

        sftp.close()

        # Send success message
        return 'Success downloading files'
    except Exception:
        if sftp:
            sftp.close()

        # Send error message
        message = 'Error downloading files: \n%s' % traceback.format_exc()
        return message
```

```python
def execute_next_operation(self) -> None:

    if self.current_operation_index < len(self.operations):
        op = Operation(self.operations[self.current_operation_index])

        # Show operation info
        op.operation_info()

        # Execute operation
        response = op.execute()

        # Handle response
        if response == "Unknown operation":
            self.operation_error(op, "Error unknown operation")
        elif response[0:7] != "Success":
            self.operation_error(op, response)
        else:
            print("Operation completed")
            self.operation_completed(op)
    else:
        self.operations_completed()
```

Amb aquest codi, ja no es necessari importar la classe OperationController a la classe GetFilesHTTPOperation ni a la GetFilesSFTPOperation ja que la comunicació amb ell es fa mitjançant el valor de retorn de la funció. D'aquesta manera, ja no és imprescindible fer servir un controlador per utilitzar aquestes classes.

Pregunta 6

"Hem pensat que no només sincronitzem fitxers a app2U, és molt normal que també sincronitzem dades entre diverses fonts. És per això que hem pensat desenvolupar un nou tipus d'operació. Els seus requeriments són:

- Ha de descarregar les dades del següent endpoint: https://gorest.co.in/public-api/users (podeu obtenir informació de com interactuar-hi a: https://gorest.co.in/)

- Ha de guardar les dades en un fitxer CSV separat per comes, on cada registre descarregat sigui una línia del fitxer"

Per a solucionar això, he creat una nova classe amb la mateixa estructura que la dels altres tipus d'operació anomenada GetFilesEndpointOperation i he afegit l'operació a realitzar a la funció create_operations() del main.

Al constructor d'aquesta nova classe se li envia l'adreça local on es vol guardar el fitxer, la url d'on s'agafaran les dades i el nom del fitxer on es guardaran les dades.

Amb el mètode download_from_endpoint() descarrego les dades de la url i les guardo al fitxer de text que he indicat al constructor. Aquest mètode segueix la mateixa estructura de gestió d'errors que els altres.

```python
def download_from_endpoint(self) -> None:
    try:
        # Get file
        print('Downloading file from %s...' % self.url)
        request.urlretrieve(self.url, os.path.join(
            self.local_path, self.data_file_name))

        # Send success message
        return 'Success downloading file from %s completed' % self.url

    except (IOError, ValueError, AttributeError):

        # Send error message
        message = 'Error downloading the file %s\n:%s' % (
            self.data_file_name, traceback.format_exc())
        return message
```

Un cop tinc les dades descarregades, si l'extracció ha sigut un èxit, converteixo aquest fitxer de text amb un csv.

```python
def data_to_csv(self, new_filename):
    # Read info from file
    with open(self.data_file_name, 'r') as file:
        data = file.read()

    data = eval(data)  # Convert string to dict
    keys = data['data'][0].keys()

    with open(new_filename, 'w', newline='') as file:
        writer = csv.DictWriter(file, fieldnames=keys)
        writer.writeheader()
        writer.writerows(data['data'])
```

Fitxer final:

users_data.csv

```
id,name,email,gender,status
1731065,Deependra Varrier,deependra_varrier@runolfsson.example,female,active
1731064,Aatreya Varman,aatreya_varman@reichert.example,female,inactive
1731063,Chandi Chopra,chandi_chopra@abernathy-gusikowski.test,female,active
1731061,Sukanya Banerjee,sukanya_banerjee@maggio.example,male,inactive
1731060,Sameer Chopra,sameer_chopra@bins.example,female,inactive
1731059,Ramesh Butt,ramesh_butt@white.example,male,active
1731058,Ananda Desai,ananda_desai@senger-leuschke.example,male,inactive
1731056,Dharmaketu Mehra,dharmaketu_mehra@glover.test,female,active
1731054,Kanak Saini,saini_kanak@bartoletti.example,male,active
1731053,Krishnadasa Kaniyar PhD,krishnadasa_kaniyar_phd@cole.test,female,inactive
```