# Elokuvateatterin varausjärjestelmä

Työn aiheena on elokuvateatterin varausjärjestelmä, jossa:

1. Ylläpitäjä pystyy luomaan/muokkaamaan/poistamaan saleja, elokuvia ja näytöksiä
2. Käyttäjä pystyy selaamaan näytöksiä ja tekemään varauksia niihin, jos vapaita paikkoja on
3. Tiedot tallennetaan JSON-formaattiin, josta ne ladataan käyttöön sovelluksen käynnistyessä

Käytössä ovat seuraavat luokat (jotka ovat omissa tiedostoissaan):

1. Application (itse sovellus, hallitsee lähinnä kirjautumista)
2. Theater (Pääosa varsinaisesta ohjelmasta, joka hyödyntää muita luokkia)
3. CinemaHall (Salien luokka)
4. Films sekä Film (Elokuvien luokat. Films on luokka, joka sisältää luokan Film objekteja)
5. Settings (Yhteen paikkaan kerättynä ohjelman komennot ja pari muuta juttua. Tämän toteutuksen järkevyydestä en ole aivan varma)

Lisäksi löytyvät seuraavat tiedostot:

1. Input\_validation (Lähinnä käyttäjän syötteiden pyytämiseen ja käsittelyyn)
2. Salit.json (tiedontallennusta)
3. Films.json (tiedontallennusta)

# Funktioita

# Application-luokka

def select\_user(self):

Käytetään kirjautumisessa joko asiakkaana tai omistajana.

def \_\_admin\_login\_passcheck(self):

Vain oikealla salasanalla (123) pääsee sisään omistajana. Tämä ei hyödynnä mitään salauksia tai muuta monimutkaisempaa.

def exit\_admin\_mode(self):

Määrittää admin-tilan epätodeksi, jolloin silmukka palaa takaisin kirjautumiskohtaan.

stop\_execution(self):

Tämän jälkeen komentoja ei enää kysellä, vaan ohjelman suoritus palaa Applicationin sisältä takaisin main-funktioon, eli käytännössä ohjelma sulkeutuu.

def \_\_execute\_commands(self):

Riippuen millä käyttäjällä kirjauduttiin sisään, tämä funktio lähettää käyttäjän silmukkaan, jossa suoritetaan käyttäjäkohtaisia komentoja (admin/customer).

def execute\_customer\_commands(self):

Tämä silmukka suorittaa asiakaskomentoja.

def execute\_admin\_commands(self):

Tämä silmukka suorittaa omistajan komentoja.

# Theater-luokka

def \_\_load\_cinema\_halls(self, path: str):

Ladataan elokuvasalien tiedot JSON-tiedostosta (Tiedoston puuttuessa se luodaan tyhjänä).

def \_\_load\_json\_data\_from\_file(self, path: str):

Avustaa yllä olevaa (\_\_load\_cinema\_halls) funktiota.

def save\_file(self):

Tallentaa elokuvasalit tiedostoon.

def cinema\_halls\_editing\_menu(self):

Silmukka salien muokkaamiskomentoja varten.

def add\_cinema\_hall(self, manual: bool = True, name: str = "Ei nimeä", seats: int = 0, shows: list = [], loading\_from\_file = False):

Lisää joko suoraan annetuilla arvoilla (esim. tiedostosta ladatessa) uuden salin. Tai vaihtoehtoisesti sali luodaan pyytämällä käyttäjältä arvot.

def edit\_cinema\_hall(self):

Pyytää muokattavan salin numeron ja tarjoaa sille muokkausoperaatioita.

remove\_cinema\_hall(self):

Pyytää poistettavan salin numeron ja poistaa sen.

def confirm\_removal\_of\_hall\_with\_shows(self) -> bool:

Jos poistettavaksi annetussa salissa on näytöksiä, kysytään varmistus poistolle.

def \_\_choose\_cinema\_hall\_number(self) -> int:

Käytetään esim. yllä olevissa funktioissa määrittämään muokattava tai poistettava sali.

def print\_cinema\_halls(self):

Tulostaa kaikki salit.

def print\_film\_selection(self):

Tulostaa kaikki elokuvat.

def films\_editing\_menu(self):

Siirtyy elokuvien käsittelysilmukkaan.

def shows\_editing\_menu(self):

Silmukka, jolla hallitaan näytöksiä.

def print\_all\_shows(self):

Tulostetaan kaikki näytökset.

def print\_halls\_and\_shows\_in\_readable\_format(self, indexed\_list: list):

Data on tallessa tiedostoissa (ja muuttujissa) hieman käyttäjäepäystävällisessä muodossa, joten tällä ja parilla apufunktiolla tulostetaan selkokielisesti olennaiset tiedot näytökselle.

def edit\_shows\_in\_a\_hall(self):

Muokkaa tietyn salin näytöksiä.

def browse\_shows(self):

Silmukka asiakkaan tarpeisiin, näytösten selaamista varten.

def seat\_reservation(self, indexed\_list, index):

Varaa paikkoja näytöksestä.

def reserve\_seats\_in\_show(self, show\_info: dict):

Silmukka, jossa selvitetään asiakkaan varaustarpeet ja suoritetaan ne, jos vapaiden paikkojen määrä sen mahdollistaa.

def get\_and\_print\_indexed\_list\_and\_chosen\_index(self, show\_list: list) -> list:

Ehkä turhan sekava funktio, joka apufunktioita hyödyntäen käsittelee annetun listan niin, että siinä olevat saavat indeksit (joita hyödynnetään komentojen kohdistamisessa) sekä kysyy käyttäjältä numeron (näistä edellämainituista indekseistä), jonka perusteella toiminnot suoritetaan tietylle näytökselle.

def choose\_show\_by\_date(self):

Käytetään näytösten etsimiseen päivämäärän perusteella.

def choose\_from\_todays\_shows(self):

Tulostaa kuluvan päivän näytökset ja pyytää valitsemaan niistä esim. yllä olevan get\_and\_print\_indexed\_list\_jnejnejne –funktion avulla.

def choose\_from\_next\_7\_days\_shows(self):

Sama kuin yllä, mutta tulostaa kuluvan päivän + viikon eteenpäin näytökset.

def get\_user\_show\_choice(self, largest\_index: int) -> int:

Kysyy käyttäjältä näytöksen, johon hän haluaa varata paikkoja.

def add\_indices\_to\_a\_list\_of\_shows(self, list: list) -> list:

Käytetään näytöslistojen indeksointiin muissa funktioissa.

def get\_shows\_from\_all\_halls(self):

Palauttaa listan kaikkien salien kaikista näytöksistä.

def sort\_list\_of\_shows(self, show\_list):

Palauttaa alkamispäivämäärän (ja ajan) mukaan lajitellun kopion annetusta näytöslistasta.

def get\_sorted\_list\_limited\_by\_days(self, days: int) -> list:

Aikaisempien funktioiden käyttämä apufunktio tulostettavien näytösten rajaamiseen nykyhetkestä x päivää tulevaisuuteen. Palauttaa listan

def get\_list\_of\_shows\_for\_given\_date(self, date: datetime.date):

Sama kuin yllä, mutta näytökset rajattu tiettyyn päivään.

def limit\_list\_of\_shows\_to\_chosen\_date(self, shows: list, chosen\_date: datetime.date) -> list:

Apufunktio yllä olevalle. Tämä tekee sen varsinaisen työn näytösten rajaamisessa päivämäärään.

def limit\_list\_of\_shows\_to\_number\_of\_days\_from\_today(self, shows: list, max\_days: int):

Sama kuin yllä, mutta päivämäärän sijaan rajataan nykyhetkestä annettu määrä päiviä eteenpäin.

# CinemaHall-luokka

def form\_dictionary\_from\_self(self): # For JSON format saving in a file

Muodostaa CinemaHall objektista sanakirjan, jotta sen saa dumpattua JSON-tiedostoon helposti.

def create\_new\_hall\_from\_input(self):

Luodaan uusi sali käyttäjän antamista arvoista.

def change\_cinema\_hall\_name(self):

Muuttaa salin nimen.

def change\_cinema\_hall\_seats(self):

Muuttaa salin istumapaikkojen määrää.

add\_show(self):

Lisää näytöksen saliin käyttäjän antamien arvojen mukaan.

def generate\_unique\_show\_id(self):

Luo näytökselle uniikin ID:n (salikohtaisesti uniikin).

def reserve\_seats\_in\_show(self, number\_of\_seats: int, show\_id: int):

Varaa paikkoja näytöksestä.

def edit\_shows(self):

Silmukka näytösten muokkaamiseen.

def remove\_show(self):

Poistaa näytöksen.

def get\_shows(self):

Palauttaa salin näytökset.

def print\_shows(self):

Tulostaa salin näytökset.

def get\_number\_of\_available\_seats(self, show\_reservations: int):

Palauttaa näytöksessä olevien vapaiden paikkojen määrän.

def get\_name(self):

Palauttaa salin nimen.

def get\_number\_of\_seats(self):

Palauttaa salin istuinpaikkojen määrän.

# Films-luokka

def \_\_load\_films(self, path: str):

Lataa elokuvat tiedostosta.

def save\_file(self):

Tallentaa elokuvat tiedostoon.

def add\_film(self, manual: bool = True, name:str = "Ei nimeä", director: str = "Ei ohjaajaa", runtime: int = 0, required\_age: int = 0, id: int = 0, loading\_from\_file = False):

Lisää elokuvan joko käyttäjän antamilla arvoilla tai sitten funktiokutsun mukana tulevilla arvoilla (esim. tiedostosta ladatessa)

def edit\_films(self):

Elokuvien muokkaussilmukka.

def edit\_film(self):

Silmukka yksittäisen elokuvan muokkaamiseen.

def remove\_film(self):

Poistaa elokuvan.

def get\_all\_film\_ids(self) -> list:

Palauttaa listan kaikista elokuvien ID:istä.

def get\_film\_name\_from\_id(self, id: int) -> str:

Palauttaa annetulla ID:llä olevan elokuvan nimen.

def film\_index\_exists(self, film\_index) -> bool:

Tarkistaa onko elokuva annetulla indeksillä olemassa (esim. poistamista ja muokkausta varten)

def print\_films(self):

Tulostaa kaikki elokuvat.

def select\_film\_for\_show(self):

Apufunktio elokuvan lisäämiseksi näytökselle.

# Film-luokka

def get\_age\_limit(self):

Palauttaa ikärajan.

def get\_name(self):

Palauttaa elokuvan nimen.

def get\_id(self):

Palauttaa elokuvan ID:n

def create\_unique\_id(self):

Luo uniikin ID:n elokuvalle, pohjautuen jo olemassa oleviin.

def change\_name(self):

Muuttaa elokuvan nimeä.

def change\_director(self):

Muuttaa ohjaajan nimeä.

def change\_runtime(self):

Muuttaa elokuvan kestoa.

def change\_required\_age(self):

Muuttaa ikärajaa.

def get\_runtime(self):

Palauttaa elokuvan keston.

def edit\_film(self):

Silmukka elokuvan muokkaamiseen.

def form\_dictionary\_from\_self(self): # For JSON

Jälleen muutetaan datan muotoa JSON-yhteensopivaksi.

# Settings-luokka

Tämä tiedosto sisältää lähinnä admin-salasanan, päiväämäärille annettavan maksimivuoden, tiedostopolut JSONeille sekä tulosteita käyttäjälle ja sanakirjat, joissa numero ja funktion nimi.

Esimerkkinä:

self.\_\_customer\_commands = {

                                "0": "self.stop\_execution",

                                "1": "self.theater.print\_cinema\_halls",

                                "2": "self.theater.browse\_shows",

                                "9": "self.select\_user"

        }

Kun käyttäjä antaa Application-luokan asiakaskomentojen silmukassa arvon 0, kutsutaan funktiota self.stop\_execution(), joka muuttaa Application-objektin self.\_\_execution\_on muuttujan Falseksi, joka taasen katkaisee kyseisen while-silmukan seuraavan kierroksen alussa.

# Input\_validation.py

Tämä tiedosto sisältää apufunktioita käyttäjältä arvojen pyytämiseksi ja niiden sisällön tarkistamiseksi sekä muutaman muun funktion, joita käytetään matkan varrella, esim. Datetime-objektien muuntamiseksi merkkijonoiksi ja toisin päin.

En usko, että olen onnistunut näillä funktioilla estämään kaikkia ohjelman kaatavia virheellisiä arvoja, joita käyttäjä voi tuputtaa sisään, mutta edes osan.

# Kirjastot

Ohjelma käyttää kirjastoja datetime ja json, jotka kuuluvat Pythonin oletuskirjastoon eli ulkoisia riippuvaisuuksia ei sen suuremmin ole.