**Main.py**

**Import re** – за регулярните изрази

**Import json** – за отварянето и пазенето на информацията във файловете / файл с потребителите и с книгите /

**from book import \*** - файл с проверките за книгите

**from user import \*** - файлове с проверките за потребители

-------------------------------------------------------------------------------------------------------------------------------------------

Class Book – основния клас за книгите

* Init
* Приема като параметри автора на книгата, заглавие, жарн, годината и на издаване, ISBN номер, брой страници, дали е поредица, кой номер е в поредицата (ако не участва в поредица се приема като 0), id номер
* add\_to\_file() – функция, която служи за добавянето на книгата във файла;
* не приема нищо като параметри
* в променливата data се записват данните на книгата;
* после отваряме файла в режим за четене (“r”) за да заредим данните от файла – това се случва в променливата file\_data
* след като сме заредили вече данните затваряме файла и го отваряме наново, обаче в режим за писане (‘’w’’), защото искаме да запишем към файла новата книга – при отварянето на файл в режим за писане, цялата информация от файла бива изтривана и писането започва от началото на файла, точно зареди това няколко реда по-нагоре отворихме файла в режим за четене и заредихме данните от файла;
* след като вече сме отворили файла в режим за писане, прибавяме към речника file\_data (данните от файла) новата книга чрез функцията .append(), после записваме вече актуализирана информация наново във файла с функцията json.dump() на която подаваме като параметри какво искаме да запишем – тоест името на променливата, в която се държат данните (в нашия случай това е file\_data) ; и също къде искаме да я запишем – тоест файла (file)
* print\_book\_info() – функция, която служи за принтирането на кратка информация за книгата
* принтира автора, заглавието и id номера на книгата

-------------------------------------------------------------------------------------------------------------------------------------------

Class User – основния клас за потребителите

* init
* Приема като параметри username, имейл, парола, име, фамилия, пол, дата на раждане
* add\_to\_file() – функция, която служи за добавянето на потребител във файла;
* същата функция като от класа Book, но вместо да добавя книга към файла, в който се пазят книгите, добавя потребител към файла с потребителите; функциолността е една и съща

-------------------------------------------------------------------------------------------------------------------------------------------

add\_book() – функция за добавяне на нова книга

1. Потребителя трябва да въведе от конзолата заглавието на книгата, ако книгата е вече вписана във файла, програмата ще пита потребителя докато не въведе ново заглавие

* проверката става чрез функцията check\_if\_book\_is\_registered();

1. След това потребителя трябва да въведе автор и жарн на книгата
2. Въвеждане на годината на публикуване

* Тук за проверка използваме exception, ако потребителя въведе нещо различно от int, ще се изпринтира грешка и потребителя ще трябва да въведе наново годината, ако потребителя си въведе int - няма да влезем в този блок и ще влезем в else-ь където става проверката дали годината наистина е валидна (чрез функцията ckeck\_if\_published\_year\_is\_correct()), като потребителя отново ще бъде питан да въведе наново година, докато не въведе валидна година

1. Въвеждане на ISBN, поредица, номер в поредицата, брой страници

* Имаме отново проверка дали данните са валидни, като са на същата база като останалите – потребителя ще бъде питан отново и отново докато не въведе валидни данни

1. Определянето на id номера

* Пак ползваме exception, ако във файла имаме нещо различно от int ще бъде изпринтирана грешка! , ако всичко е наред създаваме променлива book\_id в която да отворим файла в режим четене, защото искаме да прочетем последното записано id от файла, създаваме променлива id\_ в която да запишем прочетеното от файла (и тъй като когато четем от файла ние четем информацията като string, е нужно да превърнем първо променливата към int и добавяме към нея едно, защото искаем да променим номера на id-то с едно !тук е моментът в която ако има грешка с id-то ще влезем в except!)

-------------------------------------------------------------------------------------------------------------------------------------------

sign\_up() – функция за регистриране на нов потребител

1. Потребителя трябва да въведе от конзолата потребителско име (username), ако username-а е зает или е неправилен, програмата ще пита потребителя да въведе докато не въведе валиден username

* проверката става чрез функцията check\_if\_book\_is\_registered();

1. По същата логика потребителя трябва да въведе imeil, парола, пол и рождена дата

* Имаме отново проверка дали данните са валидни, като са на същата база като останалите – потребителя ще бъде питан отново и отново докато не въведе валидни данни

1. След това създаваме обект (u) от класа User като като параметри му даваме данните, които по-горе беше питан да въведе, и след това го записваме във файла чрез метода u.add\_to\_file() на класа User
2. Функцията return-ва вече създадения потребител

-------------------------------------------------------------------------------------------------------------------------------------------

sign\_in() – функция за влизане на вече създаден потребител

1. Създаваме променлива rev\_counter, която ще използваме по-нататък; задаваме и стойност 3
2. След това питаме потребителя дали ще иска да влезе с имейл или с потребителско име и съответно проверяваме какво е избрал – ако е избрал с имейл го питаме да въведе и имейл и проверяваме дали е въведен валиден имейл, ако не - го питаме да въведе отново докато имейла не е валиден; след като вече е въведен валиден имейл, потребителя е питан да въведе парола (за която съответно е извършена проверка дали съответства на имейла – отваряме json файла с потребителите и зареждаме данните, след това въртим for цикъл, който да премине през всички потребители и когато намери съответствие на имейлите прави проверка дали паролата съответства, ако не - програмата пита потребителя да въведе нова парола като има само 3 опита – тук изпозваме по-горе създадената rev\_counter променлива ) след успешно въведена парола програмата return-ва потребителя
3. Ако потребителя е избрал опцията да влезе в профила си с потребителско име – правим същите проверки, но вместо с имейл с потребителско име

-------------------------------------------------------------------------------------------------------------------------------------------

print\_all\_books() – функция, която да принтира всички книги записани във файла

* Отваряме json файла с книгите, зареждаме информацията от него; създаваме for цикэл който да премине през всички книги; когато сме на дадената книга създаваме обект от класа книга, на който подаваме информацията за книгата от файла, за да може да ползваме метода .print\_book\_info() на класа книга

-------------------------------------------------------------------------------------------------------------------------------------------

print\_book\_with\_id – функция, която да принтира пълната информация за книга, чрез подадено id

1. Създаваме flag променлива, която в началото да седи на 0
2. След това зарежда данните от json файла с книгите
3. Правим for цикъл, който да премине през всяка една книга и когато (ако) срещнем съответстващо id на подаденото – принтираме пълната информация на книгата
4. След това проверяваме дали книгата има rate (това става като проверяваме дали големината на book['rate'] е по-голяма от 0), ако има създаваме променлива sum\_ в която да трупаме сумата от рейтингите на книгата; с for цикъл преминаваме през всички рейтинги и ги трупаме в сумата (int(list(dict.values(rate))[0]))– чрез този код достъпваме рейтингите на книгата, понеже те са записани под формата на речник във файла и чрез функцията dict.values можем да вземем стойността на ключа без да знаем самият ключ), след като сме преминали през всички рейтинги принтираме средно аритметичното на рейтингите като делим сумата на броя рейтинги; ако книгата няма никакъв рейтинг принтираме че няма
5. След това проверяваме дали книгата име коментари по същия начин и ако има ги принтираме, ако не принтираме празния лист
6. След това променяме flag на едно
7. Накрая на функцията проверяваме дали flag-а е още 0, като ако е още 0 това значи че не сме открили книга с такова id и принтира съобщение за това

-------------------------------------------------------------------------------------------------------------------------------------------

add\_to\_list() – функция за добавяне на книга към някой от листите; приема като параметри номера на листа към който иска да добави книги, id-то на книгата и обект от класа потребител(тоест профила на потребителя от който сме влезли )

1. Създаваме три празни листа
2. След това отваряме файла в режим за четене, в който пазим потребителите и правим проверка кой е user-а който сме подали на функцията, когато намерим съответствие при потребителските имена , присвояваме на трите празни листа , които създадохме по-нагоре, информацията от файла
3. След това проверяваме към кой лист искаме да добавим книгата и правим проверка дали книгата не е вече добавена към този лист и ако е вече добавена изписваме съобщение, а ако не – я добавяме към листта
4. След това отваряме наново файла но този път в режим за писане и презаписваме вече променените данни отново във файла

-------------------------------------------------------------------------------------------------------------------------------------------

see\_list - функция за преглеждане на някой от листите; приема като параметри номера на листа, който иска да види и обект от клас апотребител(тоест профила на потребителя от който сме влязли )

1. Създаваме три празни листа
2. След това отваряме файла в режим за четене, в който пазим потребителите и правим проверка кой е user-а който сме подали на функцията, когато намерим съответствие при потребителските имена присвояваме на трите празни листа , които създадохме по-нагоре, информацията от файла
3. След това проверяваме кой лист искаме да видим и принтираме съответния лист

-------------------------------------------------------------------------------------------------------------------------------------------

remove\_from\_list() - функция за премахване на книга от някой от листите; приема като параметри номера на листа от който иска да премахне книги, id-то на книгата и обект от класа потребител(тоест профила на потребителя от който сме влезли )

1. Създаваме три празни листа
2. След това отваряме файла в режим за четене, в който пазим потребителите и правим проверка кой е user-а който сме подали на функцията, когато намерим съответствие при потребителските имена , присвояваме на трите празни листа , които създадохме по-нагоре, информацията от файла
3. След това проверяваме от кой лист искаме да премахнем книга и правим проверка дали книгата е в лист и ако я има – я премахваме от листта
4. След това отваряме наново файла но този път в режим за писане и презаписваме вече променените данни отново във файла

-------------------------------------------------------------------------------------------------------------------------------------------

rate\_book() – функция за добавяне на оценка на някоя книга; приема като параметри id номера на книгата, оценката и обект от класа потребител(тоест профила на потребителя от който сме влезли )

* Отваряме json файла, в който пазим книгите, в режим четене+ и зареждаме данните от файла; след това преминаваме през всички книги (чрез for цикъл) и когато намерим съответствие между подадения от нас id номер и id-ата на книгите от файла - добавяме нашата оценка към речника с оценките; след това отваряме наново файла но този път в режим за писане и презаписваме вече променените данни отново във файла

-------------------------------------------------------------------------------------------------------------------------------------------

comment\_book - функция за добавяне на коментар на някоя книга; приема като параметри id номера на книгата, оценката и обект от класа потребител (тоест профила на потребителя от който сме влезли)

* Отваряме json файла, в който пазим книгите, в режим четене+ и зареждаме данните от файла; след това преминаваме през всички книги (чрез for цикъл) и когато намерим съответствие между подадения от нас id номер и id-ата на книгите от файла - добавяме нашия коментар към речника с коментарите; след това отваряме наново файла но този път в режим за писане и презаписваме вече променените данни отново във файла

-------------------------------------------------------------------------------------------------------------------------------------------

Menu() – функция за основно меню, на която не подаваме нищо като параметър, а само я извикваме

1. Първо питаме потребителя какво иска да прави – да се регистрира или да си влезе в профила; след като избере се извиква съответната функция (sign\_up() или sign\_in)
2. След това имаме while True цикъл, в който ни е същинското меню
3. Потребителя е питан какво иска да прави и съответно като въведе команда се извиква съответната функция
4. Потребителя излиза от while като въведе 9

**Book.py** – използвам го за проверките за книгите

**Import re** – за регулярните изрази

**Import json** – за отварянето и пазенето на информацията във файловете / файл с потребителите и с книгите /

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_book\_is\_registered()

* функция която проверява дали подадена книгата е вече вписана във файла чрез заглавието и

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_id\_is\_valid()

- функция която проверява дали подаденото id съществува

-------------------------------------------------------------------------------------------------------------------------------------------

ckeck\_if\_published\_year\_is\_correct()

– функция която проверява дали въведената година е вярна

-------------------------------------------------------------------------------------------------------------------------------------------

ckeck\_if\_ISBN\_is\_correct()

– функция която проверява дали въведения ISBN е верен

-------------------------------------------------------------------------------------------------------------------------------------------

ckeck\_if\_pages\_are\_correct()

– функция която проверява дали въведения брой страници е коректен

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_num\_in\_series\_is\_correct()

– функция която проверява дали въведения номер е коректен

-------------------------------------------------------------------------------------------------------------------------------------------

ckeck\_if\_series\_are\_correct()

– функция която проверява дали книгата е в поредица или не

-------------------------------------------------------------------------------------------------------------------------------------------

**User.py** - използвам го за проверките за потребителите

**Import re** – за регулярните изрази

**Import json** – за отварянето и пазенето на информацията във файловете / файл с потребителите и с книгите /

------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_username\_is\_taken()

* Функция която проверяваме дали подадения username e правилен и зает

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_imeil\_is\_taken()

* Функция която проверяваме дали подадения имейл e правилен и зает

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_username\_is\_registered()

* Функция която проверяваме дали подадения username e вече регистриран

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_imeil\_is\_registered()

* Функция която проверяваме дали подадения имейл е вече регистриран

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_password\_is\_correct()

* Функция която проверяваме дали подадената парола е вярна (проверката става чрез регулярен израз)

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_sex\_is\_correct

* Функция която проверяваме дали подадения пол е валиден

-------------------------------------------------------------------------------------------------------------------------------------------

check\_if\_birthday\_is\_correct()

* Функция която проверяваме дали подадената рождена дата е валидна (проверката става чрез регулярен израз)

-------------------------------------------------------------------------------------------------------------------------------------------

**Id.txt**

* в този файл се пази id на последната книга

-------------------------------------------------------------------------------------------------------------------------------------------

**Book.json**

* Json файл, в който се пазят всички записани книги

-------------------------------------------------------------------------------------------------------------------------------------------

**User.json**

* Json файл, в който се пазят данните на потребителите

-------------------------------------------------------------------------------------------------------------------------------------------