ФМИ проект №11 – Хотел

Документация

Мартин Йорданов Пенев ФН: 81990 Група 2

Специалност: Компютърни науки

1) Уводна част:

Проектът представлява реализирана система, чрез която е възможно създаване на резервации в хотел, извеждане на информация за направените резервации, намиране на подходяща стая с определени изисквания и т. н. Към програмата е приложен файл, който съдържа определена информация, която приложението може да съхранява и обработва, чрез различни функции.

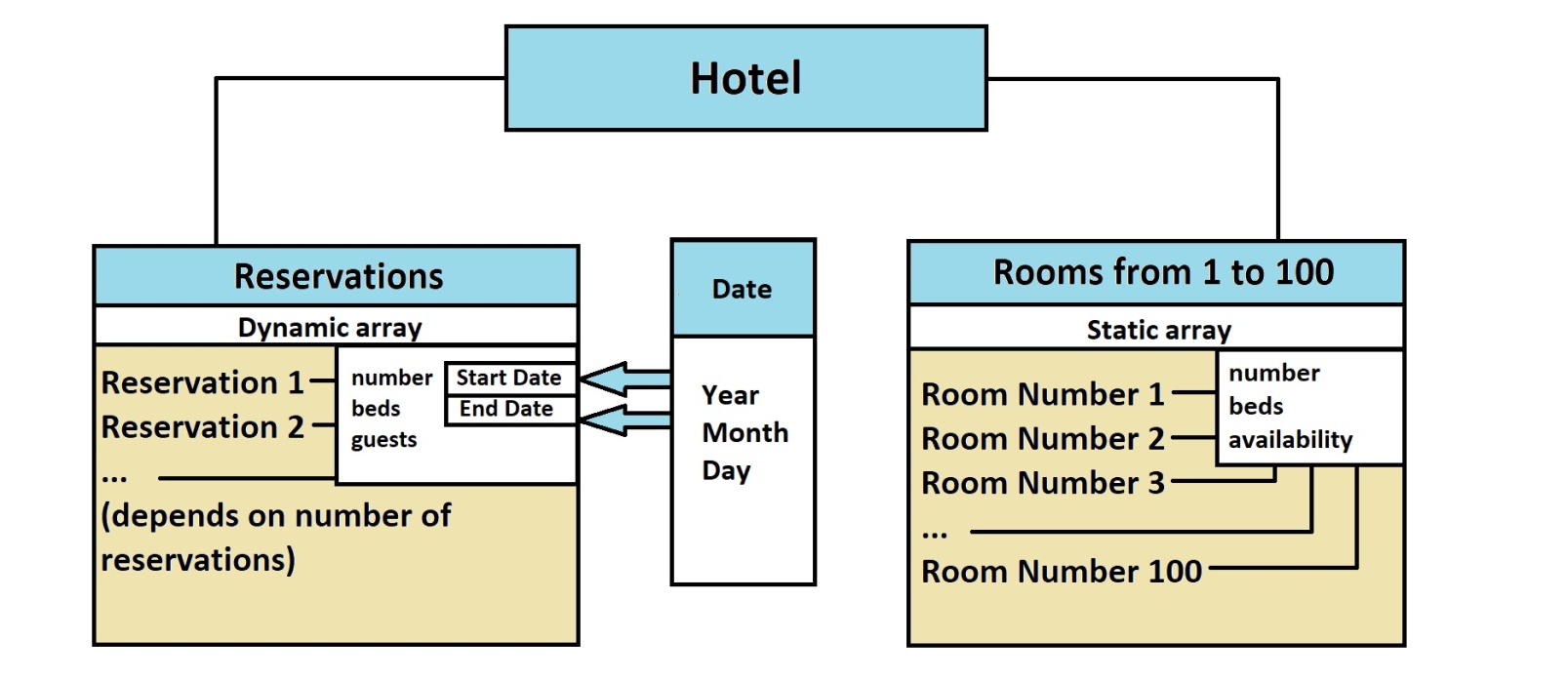
Основната цел на проекта е успешно да прочита информация от файл и да я съхранява по определен начин в паметта на програмата, след което успешно да променя съдържанието на приложения ***.txt*** документ и да го запазва (по желание на потребителя) на същото място или да предлага функция, която създава нов файл и запазва съответно нужната информация в него. Нужно е да се поддържат и определени команди, които позволяват на работещия със системата бързо и лесно да направи своята резервация, да отмени вече направена такава без да се налага консултация с физическо лице като рецепциониста на съответния хотел. Програмата би улеснила работата на всеки, който се занимава с хотелиерство, разполага с база със стаи, които желае да дава под наем. Информационната *booking* система позволява на клиентите на даден хотел да правят своите резервации денонощно, без да се налага някакво работно време.

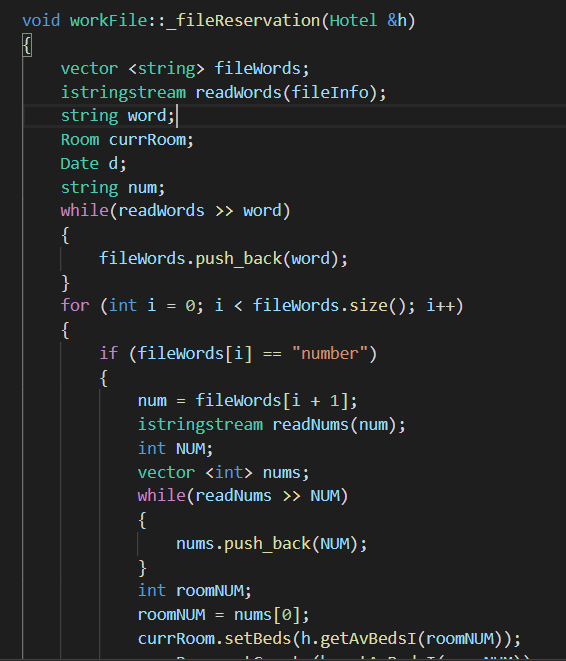
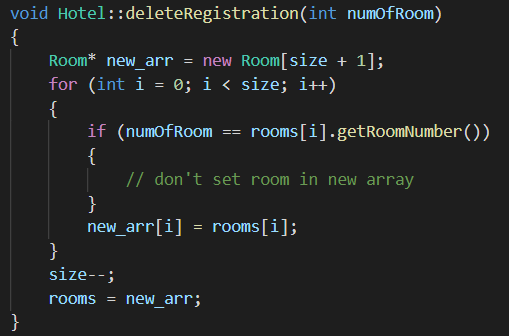
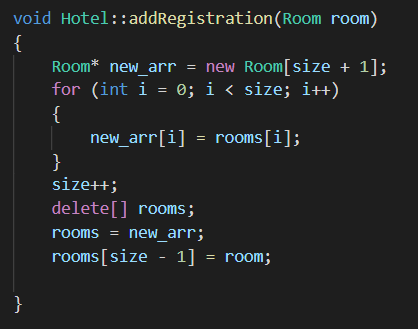
Структурата на документацията ще бъде следната:  
(1) Уводна част, в която се описва основната идея на проекта, с какво той би улеснил работата на хората и по какъв начин. Съдържа също така кратко описание на основните функции и кратка информация за начина, по който приложението обработва своите данни и работи с файлове. (2) Следва преглед на предметната област, в който се разглеждат подходите и методите при реализацията на съответната система, алгоритми и концепции. В третата част е разгледана ООП архитектурата (3) на базата, на която е изградено приложението. Накрая на документацията се показват определени тестове и реализация и е направено заключение, в което се коментира възможност за бъдещо развитие на проекта и по какъв начин той може да бъде оптимизиран.

2) Преглед на предметната област:

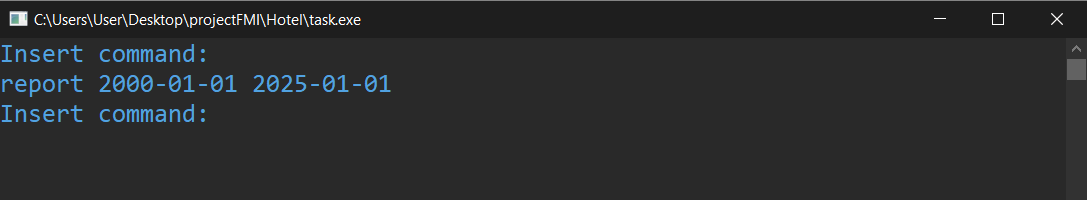
Основният проблем, който трябва да бъде решен при реализацията на съответното приложение е то успешно да работи с дати и периоди във формат ISO както е посочено в заданието. Тъй като повечето функции в приложението работят чрез подаване на някаква дата. При пускане на програмата тя автоматично генерира хотел със 100 стаи и всяка стая съответно има произволен брой легла от 1 до 5. Следните команди се поддържат и работят коректно: ***checkin*** – подава се валидиран номер на стая (не може да бъде по-голям от броя на стаите в хотела) в хотела, начална и крайна дата на престой, които също са валидирани с помощта на ***isValid*** в класа ***Date****,* след това се подава някаква бележка, която най-често представлява на чие име се запазва стаята.  
Ако не бъде посочен брой гости, стаята се запазва за толкова гости колкото са леглата в нея. Командата ***availability*** (с подадена дата или без) – извежда всички свободни стаи от първата до последната на подадената от потребителя дата. Ако потребителят не подаде дата автоматично чрез взаимстван алгоритъм посочен с коментар в кода на програмата се подава текуща дата и се извеждат свободните стаи на тази дата. За това се използва помощната библиотека *<****ctime****>.* Командата ***report***, чрез подаване на две дати извежда списък на направените резервации в периода между тях. Програмата работи чрез два масива – статичен и динамичен. Статичният масив съдържа всички стаи от първата до последната, а динамичният съдържа всички резервации, които се правят в хотела. ***Report***  извежда резервациите и работи с динамичния масив като извежда само, тези стаи, които са резервирани в посочения период, докато ***availability*** ще изкара свободните стаи от статичния масив като първо проверява резервациите и техните дати, ако подадената дата попада в периода на резервацията, съответният номер на стаята се засича и командата няма да изведе заетите стаи. Това става с помощта на булева стойност, която всяка стая притежава. По този начин двата масива си взаимодействат. Следващата команда ***find*** – чрез подаване на брой легла и период, извежда стая с точно толкова или повече легла, която е свободна в посочения период. За целта пробягва по статичния масив и отпечатва една стая по зададените изисквания. Командата ***unavailable*** – чрез подаден номер на стая, период и бележка, прави стая невалидна в определения период, съответно ако една стая е обявена за невалидна, потребителят може да я провери чрез ***availability*** или чрез ***report***. ***Checkout*** чрез подаване на номер – изтрива вече направена резервация, съответно резервацията изчезва от динамичния масив.

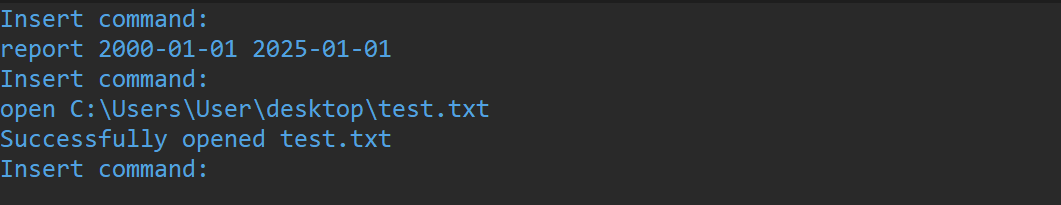
Програмата поддържа и команди, които работят с .txt файлове, които са по определен модел. Това са следните команди:  
***open*** – отваря файл на посочен адрес в компютъра, прочита информацията от него и записва в динамичния масив всички направени резервации.   
***close*** – затваря вече отворен файл.  
***save*** – запазва всички резервации направени от приложението в текстовия документ.  
***saveas*** – (с подаване на нов адрес) създава нов текстов документ и запазва цялата информация за резервациите направени от приложението или от друг файл.  
***help*** – извежда информация за всички поддържани команди.  
***exit*** – затваря отворените файлове и приключва стартираната програма.   
3) ООП архитектура и дизайн:

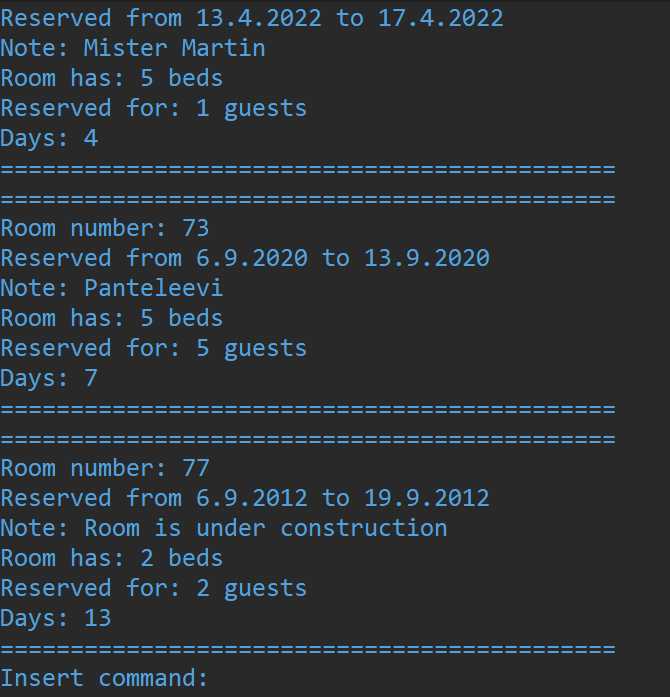


  
  
 На схемата е изобразен начинът, по който   
класовете в програмата работят заедно. Всеки хотел  
има списък с резервации и определен масив от стаи   
- в случая 100 на брой. На всяка стая могат да бъдат  
зададени съответно номер, брой легла (генерират се   
автоматично), брой гости, начална дата на престой  
и крайна дата, както и булева стойност. Класът ***Date***и неговите оператори играят ключова роля в реализацията.  
Операторите за сравняване позволяват множество  
проверки, както и операторът „ - ”, който позволява да  
се изчисли периодът, в който една стая е била заета.  
 На извадката отдясно е показана част от функцията  
\_***fileReservation*** намираща се в класа, поддържащ   
командния ред и функциите за работа с файлове.  
 С помощта на показаната функция се обработва   
информацията от файл с резервации и се записва   
в масива с резервации. Съответно с ***report*** можем да   
видим резултата от направените резервации.   
 Методите ***addRegistration*** и ***deleteRegistration***  
са едни от най-важните по отношение на функционалността  
на програмата. Те добавят и изваждат резервации от   
динамичния масив съобразено с подадени от потребителя команди.

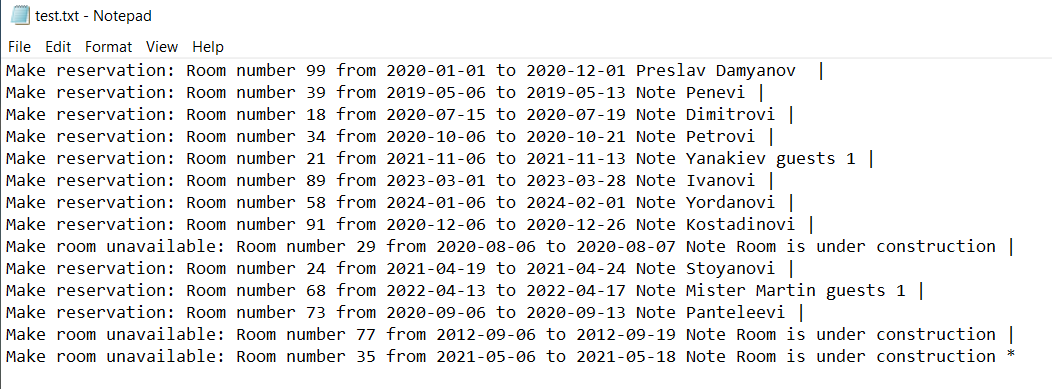
4) Реализация, тестване:  
  
 Класовете, чрез които програмата е реализирана са ***Room***, ***Hotel***, ***Date*** и ***workFile***. Като последният е за командите, които работят с файловете, прочитат информацията от тях и я презаписват коректно в паметта на програмата. В следният снимков материал е представена кратък тест на програмата и показва как програмата възприема данните от текстовия документ и разполага с тях:

1. Стартираме ***.exe*** файла на проекта и проверяваме какви резервации са направени в системата (съответно все още никакви). Правим примерна справка от дата 1 януари 2000г.  
до 1 януари 2025г.

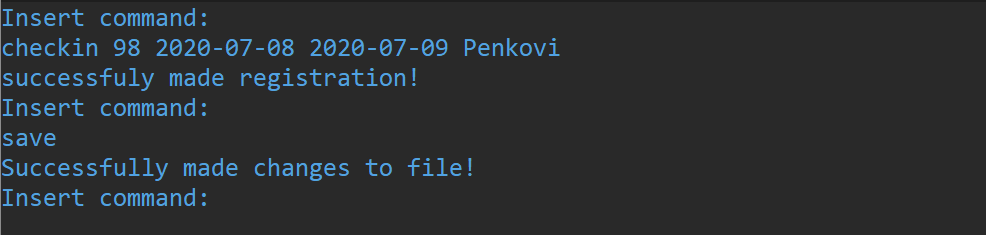
2. Отваряме примерния текстов документ през приложението и прочитаме информацията от него:

3. Правим отново справка чрез ***report***:

Появява се списък с всички направени резервации и актуализирани статуси на стаи в ремонт и т. н.

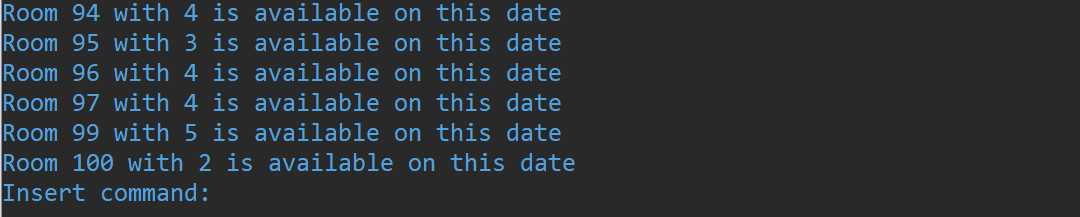
  
Съдържание на текстов документ:

4. Правим нова резервация през приложението и запазваме:

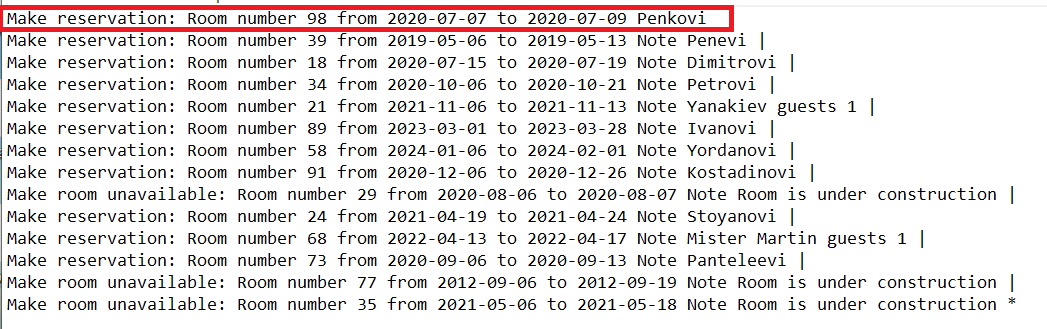


5. Проверяваме свободните стаи на дата 2020-07-08:

…



* Стая 98 липсва от списъка със свободни стаи.

6. Проверяваме текстовия документ за обновления:  


- Новата резервация вече е във файла.

7. Затваряме файла и приключваме програмата…

Приложението, представляващо информационна букинг система за хотел би имало добро практическо приложение и би било от полза на всеки, който се занимава с отдаване на стаи под наем. Като бъдеща разработка може да се направи така, че приложението да работи за различни по големина хотели с различен брой стаи, също така може да се разработят зали в хотела, които предлагат различни услуги като СПА център, ресторант и т. н. С правене на резервация потребителите да могат да заявяват   
дали ще се ползват от допълнителните услуги към съответния хотел.

Линк към проекта в гитхъб: <https://github.com/mjpenev/Hotel>