# Проект “Хотел”

**На Даниел Петров Трополинов, студент от 1 група, 1 поток, 1 курс, специалност “Компютърни науки”, ФН: 2MI0800096**

## 1. Увод

### 1.1. Описание и идея на проекта

Проектът представлява конзолно приложение, което реализира информационна система, обслужваща хотел. Идеята е за хотела да се записва и обработва информация като настанявания, затваряния на стаи и т.н., която може да бъде извличана и обработвана по различни начини. Програмата съхранява и обработва данните за стаите в хотела в двоичен файл като информацията за валидните номера на стаи в хотела се зарежда предварително във файла.

### 1.2. Цел и задачи на разработката

Целта на проекта е да бъде изградено конзолно приложение, което поддържа въвеждане на потребителски команди в диалогов режим.
Поддържаните операции са следните:

| Операция | Описание |
|---|---|
| Регистриране на гост | Регистриране на гости в стая с даден номер за въведен период, може да се въвежда и бележка за резервацията и се добавя името на госта (само едно име на стая). |
| Свободни стаи | Извежда списък на свободните стаи за дадена дата. |
| Освобождаване на стая | Освобождаване на заета стая с даден номер. |
| Справка за заетост | Извежда справка за използването на стаи в даден период и я записва в текстов файл с име report-YYYY-MM-DD.txt, където YYYY-MM-DD е началната дата на периода. Извежда се списък, в който за всяка стая, използвана в дадения период, се извежда и броя на дните, в които е била използвана. |
| Търсене на стая | Намиране на подходяща свободна стая с въведен минимален брой легла в даден период. При наличие на повече свободни стаи се предпочитат такива с по-малко на брой легла. |
| Затваряне на стая | Обявява стаята с даден номер за временно недостъпна за даден период от време (например за ремонт или основно почистване), като се добавя бележка. В стаята няма регистриран гост, но никой не може да бъде настанен в нея. |

За да бъдат разгледани всички възможни случаи, въведените потребителски данни се валидират. Ако има проблем с въведените данни, на потребителя се дава възможност (ако съществува такава) да коригира данните или да направи избор как да продължи операцията.

# 2. Проектиране

## 2.1. Допълнителни класове, използвани за реализацията на проекта:

- **Клас String**
- **Клас Vector<>** - динамичен саморазширяващ се масив от произволен тип данни

## 2.2. Обща архитектура – ООП дизайн

- **Клас Date –** поддържа валидни дати след 01.01.1950 г. и преди 01.01.10000 г.
- **Клас Time_Period –** двойка от начална и крайна дата
- **Клас Accommodation –** пази и обработва информация за настаняване. Настаняването се характеризира с:
  - uint32_t room_number – номер на стаята за конкретното настаняване
  - uint32_t number_of_guests – брой на гостите
  - Time_Period period – период на настаняването
  - String guest_name – име на госта
  - String note – бележка към настаняването (незадължителна)
- **Клас Closure** - пази и обработва информация за затваряне на стая. Затварянето се характеризира с:
  - uint32_t room_number - номер на стаята за конкретното затваряне
  - Time_Period period - период на затварянето
  - String note – бележка към затварянето на стаята
- **Клас Room –** реализира стая, характеризираща се с:
  - uint32_t room_number – номер на стаята
  - uint32_t number_of_beds – брой на леглата в стаята
  - Vector<int32_t> accommodations_indexes_in_file_list – списък на индексите в двоичния файл, на които са запазени всички настанявания в конкретната стая
  - Vector<int32_t> closures_indexes_in_file_list – списък на индексите в двоичния файл, на които са запазени всички затваряния на конкретната стая
- **Клас Hotel** – пази и обработва информация за хотела. Съдържа:
  - Vector<Room> rooms – списък на стаите в хотела
  - std::fstream file – двоичният файл, в който се съхранява цялата информация за хотела

# 3. Подход и реализация, тестване

## 3.1. Подход и реализация на функционалностите

- **Регистриране на гост –** въвеждат се необходимите данни за резервацията:
  - номер на стая
  - брой на гостите

- начална дата
- крайна дата
- име на гост
- бележка(незадължително)

Проверява се дали нововъведената резервация не се припокрива с вече запазена резервация или затваряне на стаята. Ако няма колизии, резервацията се записва във файла и началната позиция на записа във файла се запазва в списъка на индексите за настаняванията за конкретната стая. Ако настаняването не е възможно поради някаква причина се дават част от следните опции на потребителя (които са възможни):

- настаняване на госта в друга подходяща стая (ако такава съществува) – директно се използва функционалност 5 – търсене на стая
- преместване на гостите, чиято резервация в момента е записана за тази стая, в подходяща стая (ако съществува единствена такава резервация, припокриваща се с нововъведената)
- отказ на нововъведената резервация

- **Свободни стаи** – въвежда се дата. Преминава се през списъкът от стаи в хотела и за всяка се проверява дали някой период на настаняване или затваряне на конкретната стая не съдържа въведената дата. Ако няма такъв период – стаята е свободна и се извежда.
- **Освобождаване на стая** – въвеждат се номер на стая и дата на напускането на гостите. Ако не съществува стая с въведения номер или на конкретната дата няма никой настанен – потребителят бива известен. В противен случай в записът на настаняването, съдържащо въведената дата, се променя крайната дата на нововъведената дата.
- **Справка за заетост** – въвежда се период. Преминава се през списъкът от стаи в хотела и за всяка се проверява за колко дни от въведения период е имало настанени хора в стаята. Информацията се записва в текстов файл с име report-YYYY-MM-DD.txt
- **Търсене на стая** – въвежда се брой на хората и период. Стаите в хотела предварително са сортирани в нарастващ ред по отношение на леглата в тях. Преминава се последователно през всички стаи и се връща номерът на първата, която има достатъчно легла и за която никое настаняване в нея не се припокрива с въведения период.
- **Затваряне на стая** – въвежда се номер на стая, период на затварянето и причина за затварянето. Проверява се дали нововъведеното затваряне не се припокрива с вече запазена резервация или затваряне на стаята. Ако няма колизии, затварянето се записва във файла и началната позиция на записа във файла се запазва в списъка на индексите за затварянията за конкретната стая. Ако затварянето не е възможно поради някаква причина, потребителят бива известен.

**Бележка:** Всички гореизброени проверки за припокриване на интервали се извършва като първо нужните интервали се извлекат от двоичния файл, пазещ цялата информация.

## 3.2. Тестване

Всички функции на класовете и реализираните функционалности са тествани по време на изграждането на проекта. Приложен е файл с примерна интеракция с програмата (example.pdf) заедно с двоичния файл, пазещ информацията след интеракцията (hotel_example3.bin). Проектът е

реализиран така, че след като един файл бъде зареден, то освен стаите записани в него, се четат настанявания и затваряния от предишни взаимодействия с програмата. Тоест горният пример може да бъде продължен ако заредим файла hotel_example3.bin при стартиране на програмата. Допълнително са предоставени още два файла (hotel_example1.bin и hotel_example2.bin), в които са записани само номера на стаи и техните капацитети, както и сорсът, който ги е генерирал (saveroomstofile.cpp).

# 4. Заключение и бъдещи подобрения

Успешно бе създадена програма, реализираща информационна система, обслужваща хотел.

Неща, които биха могли да се подобрят:

- обработката на данните може да стане по-бърза ако те се пазят сортирани и се използва двоично търсене или други подходящи алгоритми за търсене.

**Използвана литература:**


- Идея за оператор >> и getline функция за класа String - Link
- Идея за клас Date от семинарите на 6 група - Link


**Връзка към хранилище, съдържащо реализацията на проекта:** GitHub