Технически университет – Варна,

Факултет по Изчислителна Техника и Автоматизация,

Катедра: „ Софтуерни и Интернет технологии“,

специалност „ Софтуерни и Интернет технологии“,

тема: Хотел,

Борис Недков Луков, №20621544

II курс, 2а) група.

1. **Увод**
   1. Идеята на проекта е да реализира програма, която има за цел да управлява стаите в даден хотел. Като има възможност за промяна на данните и използване на специфични методи, които позволяват да се извършват дадени промени и актуализации на стаите.
   2. Целта на програмата е да използва манипулиране на файлове, като така може да създава нови, да чете вече създадени, да записва и променя съдържанието. С това се улеснява работата върху структурата на задачата и позволява по-големи и четими данни да бъдат запазени. Задачата е да се създават методи и да се използва меню, към което са свързани за да може да се използват за промяната на някой от данните, които се въвеждат от потребителя. За целта е нужно да се използват дати и да се форматират типовете данни от прграмата. Това е и една от задачите, които програмата трябва да изплни. Друг важе
   3. Документацията съдържа увод, в който има кратко описание за програмата и какви са целите и на работа. Преглед на проблемната област, където се разглеждат проблеми и как да бъдарт решени. Представено е и проектирането и какви пакети са използвани. Включени са реализации и примерни данни към кода с цел тестване и изобразяване на данните. Представено е и заключение и използвана литература.
2. **Преглед на предметната област**.
   1. Използвани са алгоритми за сортиране

rooms.sort(new Comparator<RoomDetails>() {  
 public int compare(RoomDetails o1, RoomDetails o2) {  
 if (o1.getBeds() == o2.getBeds())  
 return 0;  
 return o1.getBeds() < o2.getBeds() ? -1 : 1;  
 }  
});

* 1. Проблем при реализирането на програмата беше превръщането и предаването на типът Date на данни взети от конзолата, въведени от потребителя, които усложняват работата и степента на сложност на кода. Сложността не висока, но има определени места, където е възможно възникване на усложнения поради непредвиден проблем при изработването на задачата.
  2. Поддръжката на програмата не е от висока степен на сложност, но скаларуемостта на проекта може да принуди смяна на основни методи и логики на класове, което ще допринесе до висока степен на сложност и удължаване на времевият период, за който може да бъде променена.

1. **Поектиране**

3.1Проектът съдържа три пакета за разделяне на класовете и пет класа за реализация.

public void checkin() throws ParseException {  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
 System.*out*.println("Enter room number.");  
 int op = scanner.nextInt();  
 System.*out*.println("Enter date from: ");  
 String op1 = scanner.next();  
 System.*out*.println("Enter date to: ");  
 String op2 = scanner.next();  
 System.*out*.println("Enter number of guests: ");  
 int op3 = scanner.nextInt();  
 for (RoomDetails roomDetails : *roomOperations*) {  
 if (roomDetails.getRoomNumber() == op && !roomDetails.getNote().equals("Unavailable")) {  
 roomDetails.setNote("Taken");  
 roomDetails.setDateFrom(sdf.parse(op1));  
 roomDetails.setDateTo(sdf.parse(op2));  
 roomDetails.setNumGuests(op3);  
 }  
 }  
 System.*out*.println(*roomOperations*);  
  
}

Метод от класът RoomOperations, който показва употребата на scanner за вземане на данни въведени от клавиатурата. Обхождане на масиви с цикли както и проверки за параметри на даден обект.

Доста от логиките биват използвани и в други подобни методи, като се добавят нови условия и алгоритми. Като например алгоритъм за сортиране.

public void find() throws ParseException {  
  
 ArrayList<RoomDetails> rooms = new ArrayList();  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
 System.*out*.println("Enter date from: ");  
 String op = scanner.next();  
 System.*out*.println("Enter date to: ");  
 String op2 = scanner.next();  
 System.*out*.println("Enter number of beds: ");  
 int op3 = scanner.nextInt();  
 for (RoomDetails roomDetails : *roomOperations*) {  
 if (roomDetails.getBeds() >= op3) {  
 rooms.add(roomDetails);  
 }  
 }  
  
 rooms.sort(new Comparator<RoomDetails>() {  
 public int compare(RoomDetails o1, RoomDetails o2) {  
 if (o1.getBeds() == o2.getBeds())  
 return 0;  
 return o1.getBeds() < o2.getBeds() ? -1 : 1;  
 }  
 });  
 System.*out*.println("Enter number of guests: ");  
 rooms.get(0).setNumGuests(scanner.nextInt());  
 rooms.get(0).setNote("Taken");  
 rooms.get(0).setDateFrom(sdf.parse(op));  
 rooms.get(0).setDateTo(sdf.parse(op2));  
 }

Метод от RoomOperations, който използва проверки във цикъл на арейлист, които позволяват намиране на най-подходяща стая и задаването на параметрите и спрямо въведени от клавиатурата дата и номер на гости. Алгоритъмът проверява броят на леглата, които са предварително зададени за стаите като сравнява дали има брой легла въведени от клавиатурата или повече свободни с цел ако няма стая с поне n(брой легла въведени от клавиатурата) легла да вземе следваща стая с минимален брой легла със стойност повече от n.

1. **Реализация и тестове**
   1. Реализирането на кода става чрез въвеждане на команда в конзолата

Text

Description automatically generated

Примерна команда в конзолата.

Както за цел има опция help, която представя всички възможни команди и какво им е предназначението.

Text

Description automatically generated

Команда help.

Text

Description automatically generated

Примерно въвеждане на данни в команда checkin.

1. **Заключение** 
   1. Възможно е развитие на проектва в уголемяване на броя на стаите и въвеждане на повече характеристики. Възможно е и подобряване и развитие на методите.

**Литература**

https://www.delftstack.com/howto/java/java-read-xml/

https://www.tutorialspoint.com/java\_xml/java\_dom\_create\_document.htm

https://stackoverflow.com/questions/21125998/how-to-write-an-arraylist-to-an-xml-file?fbclid=IwAR3HGsnnYpJitAornbXpqRwgvkyzW258bW4tjBxs5DYz53SHw2MG6gESQ3Q

https://dirask.com/posts/Java-sort-ArrayList-based-on-Object-field-przXBp

https://stackabuse.com/how-to-get-the-number-of-days-between-dates-in-java/