Введение

Данная курсовая работа выполнена с использованием методов объектно-ориентированного программирования на языке С++.

Объектно-ориентированное программирование – методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Основными принципами ООП являются:

* Абстракция — отделение концепции от ее экземпляра;
* Полиморфизм — реализация задач одной и той же идеи разными способами;
* Наследование — способность объекта или класса базироваться на другом объекте или классе. Это главный механизм для повторного использования кода. Наследственное отношение классов четко определяет их иерархию;
* Инкапсуляция — размещение одного объекта или класса внутри другого для разграничения доступа к ним.

В ходе выполнения данной курсовой работы была реализована программа «Бронирование отеля», которая создана для выбора и бронирования номера в отеле. Также реализована опция «лояльности клиентам».

1. Техническое задание
2. Наименование программы

Наименование программы: «Бронирование отеля»

1. Цель создания и назначения программы

Целью данного проекта является приобретение практических навыков по проектированию в поэтапной разработке объектно-ориентированных программ, их отладке и документации.

«Бронирование отеля» - это специальная программа, которая позволяет найти и забронировать понравившийся номер.

Основной целью использования программы является упрощение поиска и бронирования нужного номера среди большого количества отелей и номеров.

1. Требования к программе
2. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

* Программа должна иметь интуитивно понятный интерфейс;
* У программы должна быть функция выбора отель по одному из критериев;
* У программы должна быть функция выбора критериев номер в отеле по одному из критериев;
* Программа должна уметь проверять клиента по клиентской базе;
* Программа должна уметь выводить номера и отели на экран;
* Программа должна уметь предоставлять скидку новым клиентам;

1. Требования к надежности

Надежное функционирование программы обеспечивается выполнением совокупности организационно-технических мероприятий, приведенных ниже:

* Обеспечение питания технических средств от источников бесперебойного питания;
* Использование лицензированного программного обеспечения;
* Регулярное выполнение рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 года «Об утверждении межотраслевых типовых норм и времени работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств».
* Регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытание программных средств на наличие компьютерных вирусов;

1. Условия эксплуатации
2. Требования к составу и параметрам технических средств

В состав технических средств должны входить персональный компьютер (ПЭВМ), включающий в себя:

1. Процессор Pentium-2.0Hz, не менее;
2. Оперативную память объемом не менее 1 Гигабайта;
3. Любую операционную систему;
4. Стандартный пакет C++.
5. Требования к программной документации

Состав программной документации должен включать в себя:

1. Техническое задание;
2. Код программы и ее описание;
3. Программу и методики испытаний;
4. Руководство пользователя.
5. Стадии и этапы разработки
6. Стадии разработки

Разработка должна быть проведена в три стадии:

1. Разработка технического задания;
2. Рабочее проектирование;
3. Релиз.
4. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания. На стадии рабочего проектирования и утверждения должны быть выполнены перечисленные ниже этапы работ:

1. Разработка программы;
2. Разработка программной документации;
3. Испытание программы.

На стадии релиза должен быть выполнен этап разработки, подготовки и передача программы.

1. Порядок контроля и приема.
2. Виды испытаний

Проверка ввода данных для корректной работы программы.

1. Общие требования к приемке работы

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи;
2. Определение и уточнение требований к техническим средствам;
3. Определение требований к программе;
4. Определение стадий, этапов и сроков разработки программы и документации на нее;
5. Согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. Разработка, согласование и утверждение методики испытаний;
2. Проведение приемо-сдаточных испытаний;
3. Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию.

1. Проектирование программы

Для реализации проекта будет использован язык программирования С++. Данный язык является одним из самых наиболее распространенных объектно-ориентированных языков программирования (ООП).

Целью данной работы является написание приложения для бронирования номера в отеле. Программа будет представлять из себя консольное приложение. Для удобства будет разработан пользовательский интерфейс. В начале работы программы пользователю будет представлен выбор: открыть файл «Инструкция», открыть файл «Об авторе» или начать пользоваться программой. После поиска и выбора нужного отеля и номера начинается этап бронирования. После окончания которого пользователю будет предоставлен выбор: продолжить работу программы или выйти из нее.

1. Разработка программы

Программа состоит из трех главных компонентов:

1. Класс Buyer является вспомогательным классом программы «Бронирование отеля», в котором описана и реализована вся логика программы, связанная с покупателем. На рисунке 3.1 представлена его UML диаграмма.
   1. Поля
      1. id – индикационный номер покупателя;
      2. name – имя покупателя;
      3. surname – фамилия покупателя.
   2. Методы
      1. ~Buyer – деструктор класса, вызываемый при уничтожении экземпляра класса;
      2. Buyer – конструктор класса, вызываемый при создании экземпляра класса;
      3. Метод getId возвращает индикационный номер покупателя;
      4. Метод getName возвращает имя покупателя;
      5. Метод getSurname возвращает фамилию покупателя;
      6. Метод print выводит на экран все содержимое;
      7. Метод setBuyer позволяет установить все данные о покупателе;
      8. Метод setid позволяет установить индикационный номер покупателя;
      9. Метод setname позволяет установить фамилию покупателя.

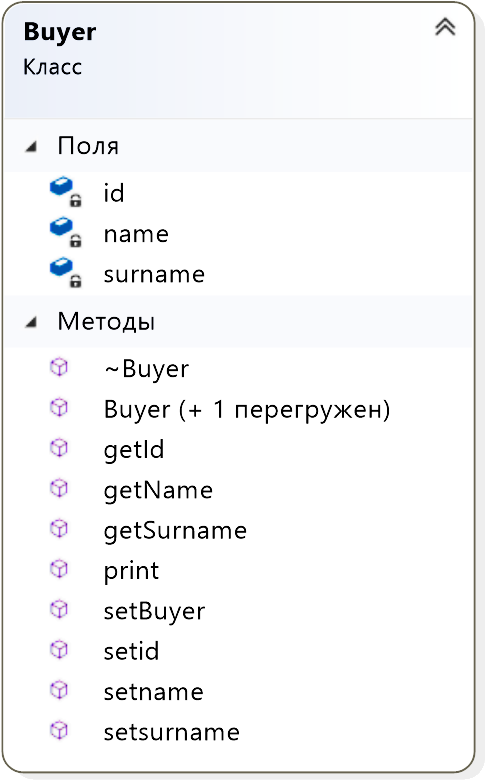


Рис. 3.1 UML диаграмма класса Buyer

1. Класс House является вспомогательным классом программы «Бронирование отеля», в котором описана и реализована вся логика программы, связанная с отелями. На рисунке 3.2 представлена его UML диаграмма.
2. Поля
3. city – название города, в котором находится отель;
4. country – название страны, в котором находится отель;
5. number\_of\_rooms – количество комнат в отеле;
6. number\_of\_star – количество звезд отеля.
7. Методы
8. ~House – деструктор класса, вызываемый при уничтожении экземпляра класса;
9. Метод getCity возвращает город отеля;
10. Метод getCountry возвращает страну отеля;
11. Метод getNumber\_of\_rooms возвращает количество комнат в отеле;
12. Метод getNumber\_of\_star возвращает количество звезд отеля;
13. House – конструктор класса, вызываемый при создании экземпляра класса;
14. Метод Print выводит на экран все содержимое;
15. Метод setcity позволяет установить город, в котором находится отель;
16. Метод setcountry позволяет установить страну, в которой находится отель;
17. Метод setHouse позволяет установить все данные об отеле;
18. Метод setnumber\_of\_rooms позволяет установить количество комнат в отеле;
19. Метод setnumber\_of\_star позволяет установить количество звезд отеля.

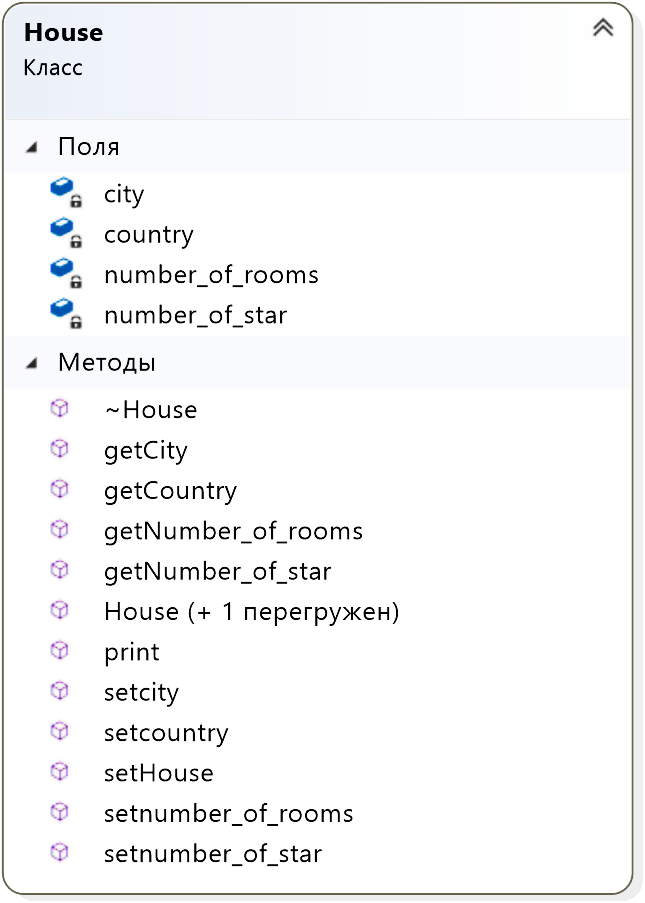


Рис. 3.2 UML диаграмма класса House

1. Класс Room является основным классом программы «Бронирование отеля», в котором описана и реализована вся логика программы, связанная с бронированием комнаты. На рисунке 3.3 представлена его UML диаграмма.
2. Поля
3. d1 – день начала брони;
4. d2 – день окончания брони;
5. floor – этаж комнаты;
6. free – свободность/занятость номера;
7. m1 – месяц начала брони;
8. m2 – месяц конца брони;
9. number – номер комнаты;
10. number\_of\_beds – количество кроватей в номере
11. price – суточная стоимость номера;
12. y1 – год начала брони;
13. y2 – год конца брони.
14. Методы
15. ~Room – деструктор класса, вызываемый при уничтожении экземпляра класса;
16. Метод datain устанавливает дату начала брони;
17. Метод dataout устанавливает дату конца брони;
18. Метод display выводит на экран все содержимое;
19. Метод getDay\_of\_check\_in возвращает день начала брони;
20. Метод getDay\_of\_check\_out возвращает день конца брони;
21. Метод getFloor возвращает этаж номера;
22. Метод getFree возвращает свободность/занятость номера;
23. Метод getMonth\_of\_check\_in возвращает месяц начала брони;
24. Метод getMonth\_of\_check\_out возвращает месяц конца брони;
25. Метод getNumber возвращает номер комнаты;
26. Метод getNumber\_of\_beds возвращает количество кроватей;
27. Метод getPrice возвращает суточную стоимость номера;
28. Метод getSale возвращает стоимость номера со скидкой;
29. Метод getYear\_of\_check\_in возвращает год начала брони;
30. Метод getYear\_of\_check\_out возвращает год конца брони;
31. Room – конструктор класса, вызываемый при создании экземпляра класса;
32. Метод setDay\_of\_check\_in устанавливает день начала брони;
33. Метод setDay\_of\_check\_out устанавливает день конца брони;
34. Метод setFloor устанавливает этаж номера;
35. Метод setFree устанавливает свободность/занятость номера;
36. Метод setiNumber\_of\_beds устанавливает количество кроватей в номере;
37. Метод setMonth\_of\_check\_in устанавливает месяц начала брони;
38. Метод setMonth\_of\_check\_out устанавливает месяц конца брони;
39. Метод setNumber устанавливает номер комнаты;
40. Метод setPrice устанавливает цену номера;
41. Метод setRoom устанавливает все данные о номере;
42. Метод setYear\_of\_check\_in устанавливает год начала брони;
43. Метод setYear\_of\_check\_out устанавливает год конца брони.

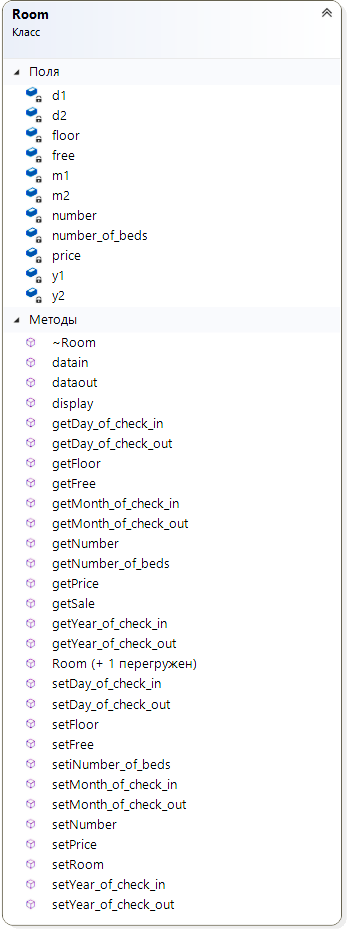


Рис. 3.3 UML диаграмма класса Room

1. Руководство пользователя

Запуск программы осуществляется при помощи исполняемого файла Bron.exe. После запуска в программе отобразятся варианты продолжения работы с программой (рис. 4.1). Особенность данной программы является интуитивно понятный интерфейс, разобраться в котором не составит труда.

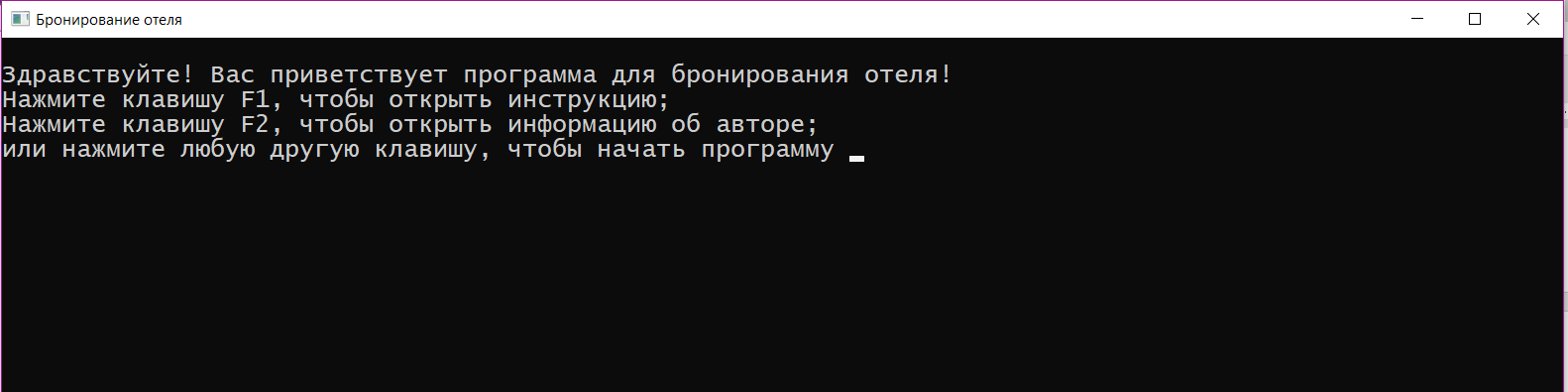


Рис. 4.1 Старт

В программе предусмотрена простейшая защита от ввода некорректных данных, но просьба четко следовать инструкциям, которые будут появляться на экране.

Далее возможны 3 исхода:

1. Пользователь открывает файл с инструкцией (рис. 4.2);

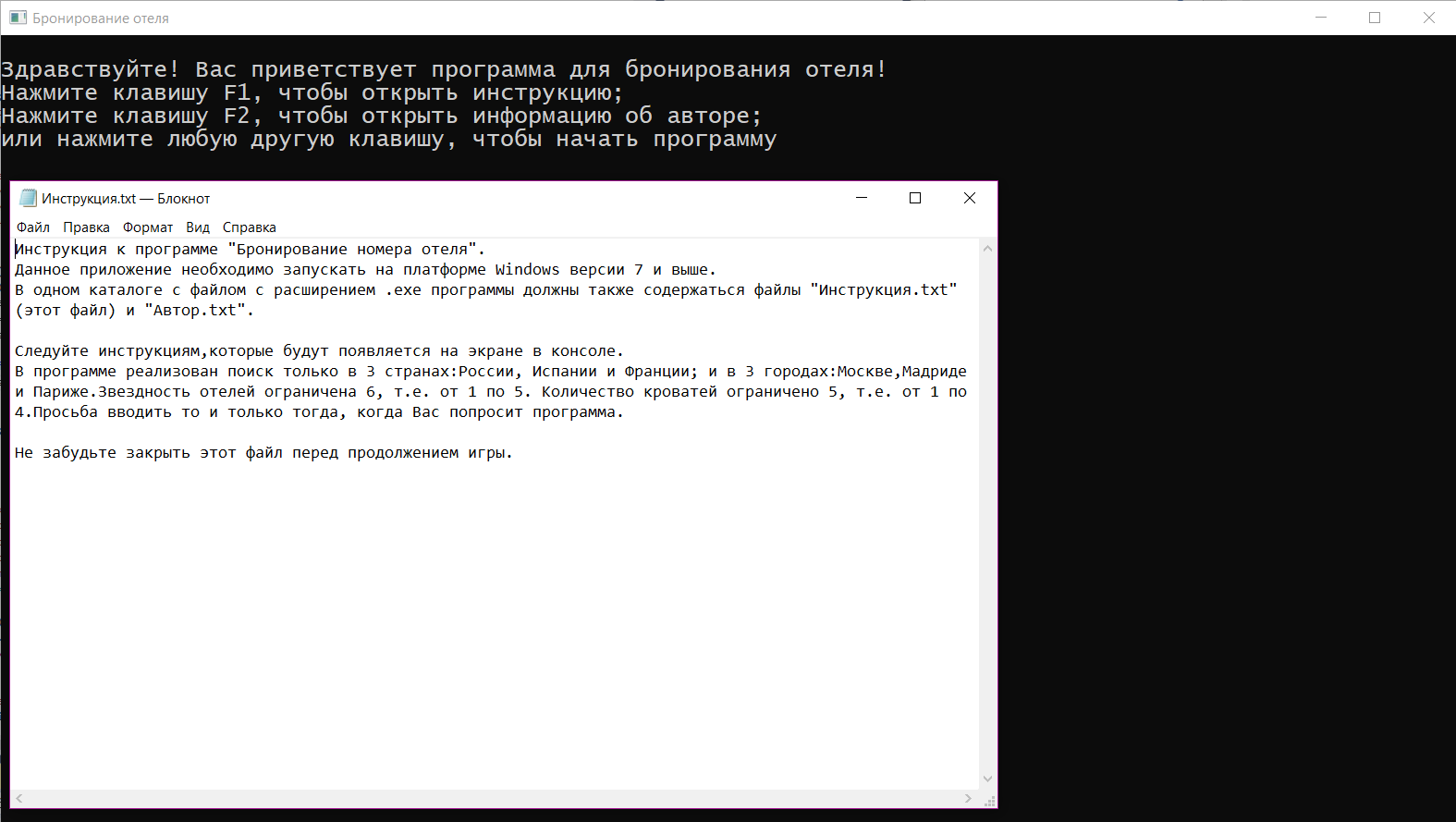


Рис. 4.2 Открытие инструкции

1. Пользователь открывает файл с информацией о авторе (рис. 4.3);

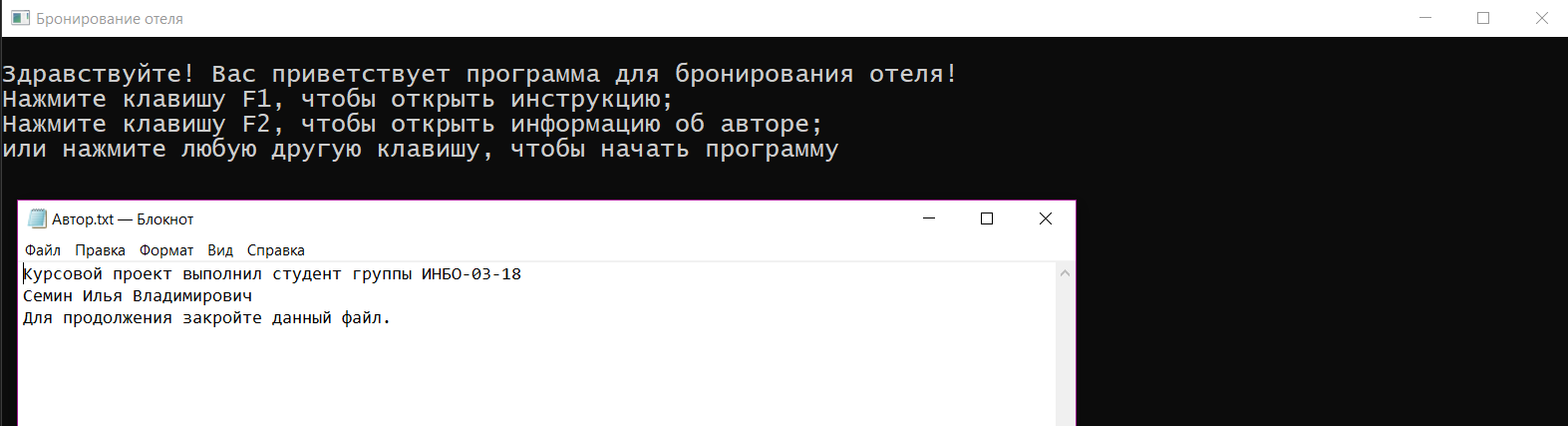


Рис. 4.3 Открытие информации об авторе

1. Пользователь начинает искать отель для бронирования (рис. 4.4).

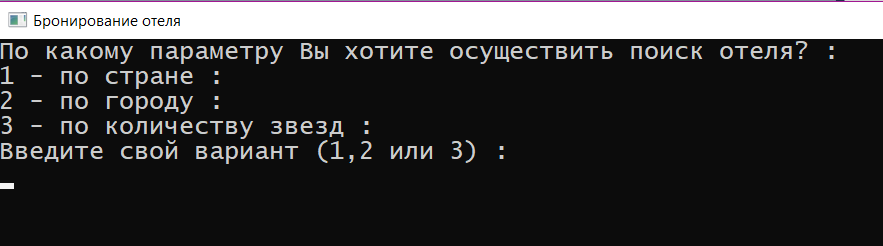


Рис. 4.4 Поиск отеля

Далее идет выбор параметра, по которому будет осуществляться поиск отеля (рис. 4.5):

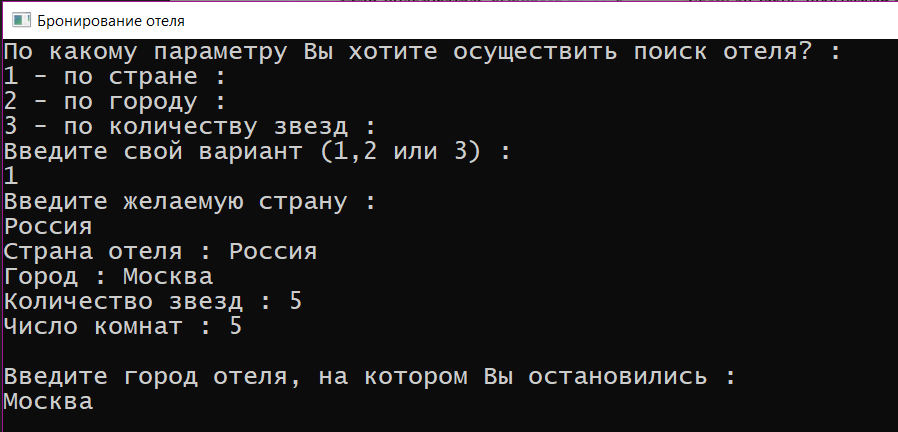


Рис. 4.5 Поиск отеля по критерию «страна»

Далее идет выбор параметра, по которому будет осуществляться поиск номера, и вывод подобранных номеров на экран (рис. 4.6):

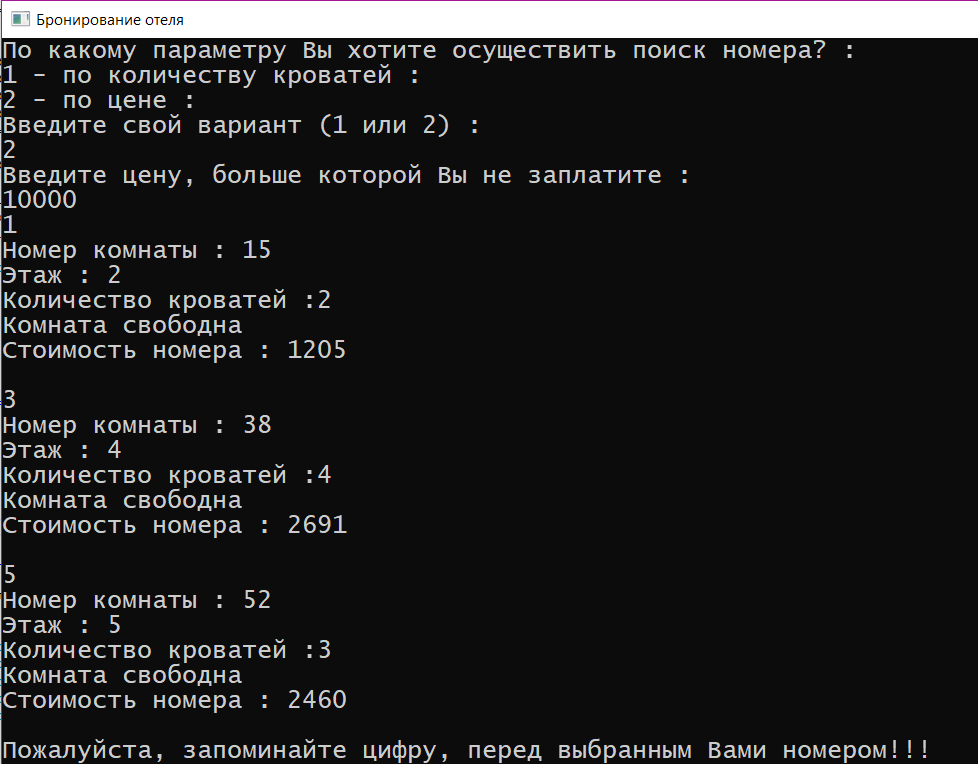


Рис. 4.6 Поиск номера по критерию «цена»

В следующей части программы предлагается ответить на вопрос. В случае положительного ответа произойдет проверка, и, если она закончится успешно, будет предоставлена скидка (рис. 4.7):

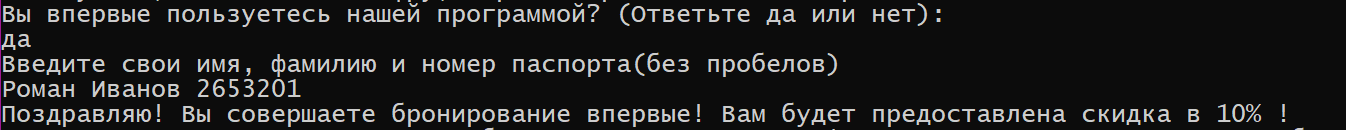


Рис. 4.7 Предоставление скидки новому клиенту

Если же вы захотите обмануть программу или ошибетесь будет выведено следующее (рис. 4.8):

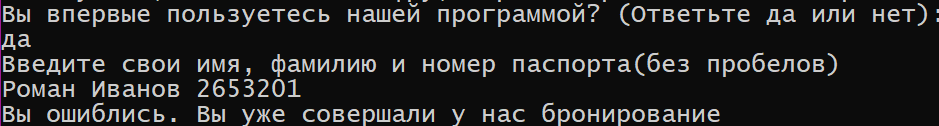


Рис. 4.8 Вывод ошибки для старого клиента

Далее программа предложит ввести цифру, выведенную перед выбранным номером (рис. 4.9):

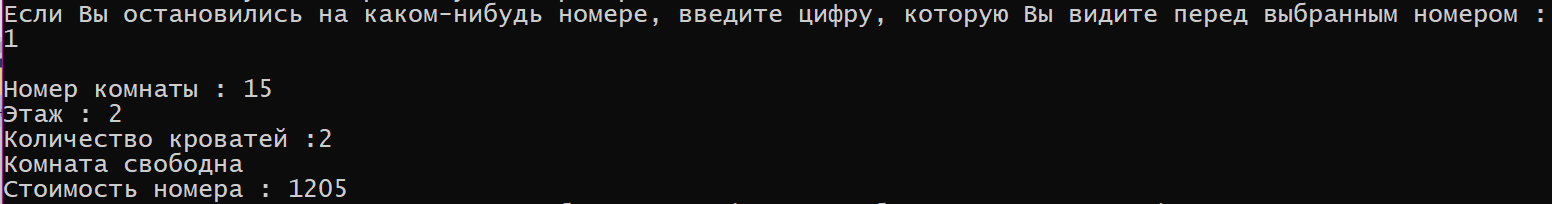


Рис. 4.9 Вывод на экран выбранного номера

После этого начинается выбор дат для бронирования номера (рис. 4.10):

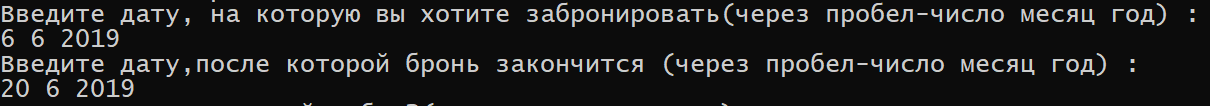


Рис. 4.10 Выбор дат бронирования

Далее следует подтверждение. Если нет подтверждения, то начнется выбор другого номера в этом отеле (рис. 4.11):



Рис. 4.11 Подтверждение выбора

Выводится сообщение об успешном бронировании. Выводится забронированный номер (рис. 4.12). И если это клиент новый, то и цену со скидкой (рис. 4.13).

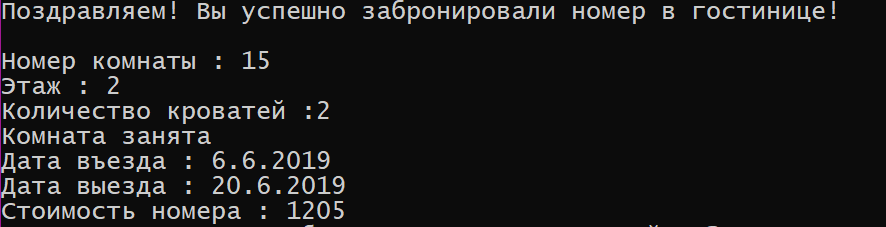


Рис. 4.11 Успешное бронирование и вывод информации о номере



Рис. 4.12 Вывод скидки для нового пользователя

Проверка (рис. 4.13):

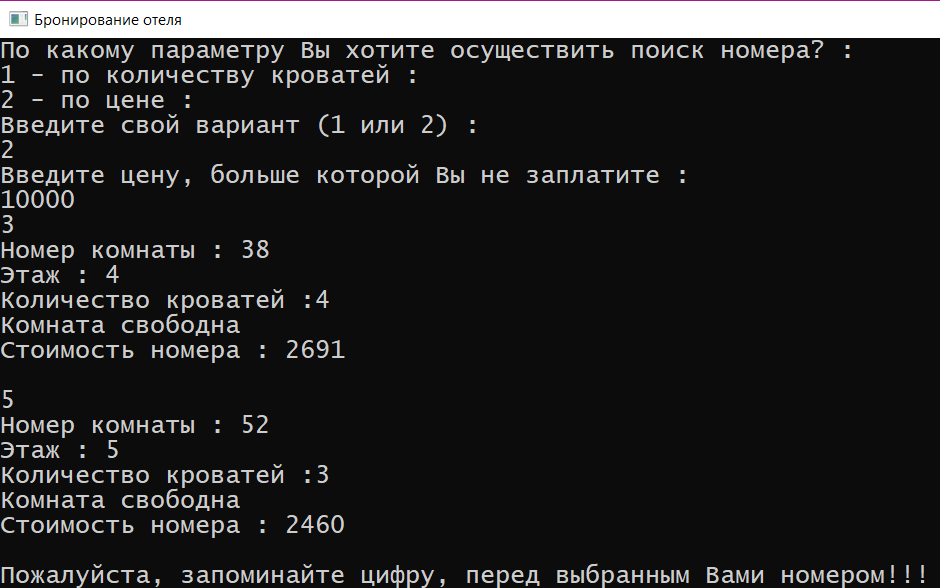


Рис. 4.13 Проверка того, что номер забронирован

В программе предусмотрена защита от некорректных данных, так что если она их встретит, то уведомит пользователя об ошибке и продолжит работу.

1. Заключение

В результате выполнения курсовой работы была реализована программа бронирования номера в отеле. Использовались основные принципы Объектно-Ориентированного Программирования. В ходе написания кода был закреплен теоретический материал, полученный на лабораторных и практических занятиях, а также отточены навыки по проектированию и поэтапной разработке объектно-ориентированных программ, их отладке и документации.