# Вариант 2 Гостиница

Разработать приложение, позволяющее организовать работу по учёту занятости номеров гостиницы. Информация о номере хранится в некотором описателе номера.

Описатель номера “*люкс*” содержит следующую информацию: свободен или занят (0 или 1); количество комнат (не меньше двух) и количество проживающих (один или два гостя); дата регистрации и число дней проживания; суточный тариф проживания в номере.

Описатель *одноместного* номера содержит следующую информацию: свободен или занят (0 или 1); дата регистрации и число дней проживания; суточный тариф проживания в номере.

Описатель *многоместного* номера содержит следующую информацию: свободен или занят (0 или 1); количество мест и сколько занято (два числа); для каждого проживающего (от одного до четырёх гостей) дата регистрации и число дней проживания; суточный тариф проживания в номере для одного гостя.

Информация обо всех номерах гостиницы сведена в таблицу. Каждый элемент таблицы содержит “указатель номера” и указатель на его описатель. Элементы упорядочены по возрастанию значений “указателей номера”.

Обеспечить выполнение следующих операций.

* Для таблицы:
  + включить новый элемент, не нарушая упорядоченности;
  + найти элемент по заданному “указателю номера”;
  + удалить элемент, заданный “указателем номера”;
  + показать содержимое таблицы.
* Для любого номера:
  + вывести информацию о номере;
  + вернуть в качестве результата тип номера;
  + вернуть в качестве результата состояние номера (свободен или занят); вернуть в качестве результата число гостей в номере;
  + занять номер (с регистрацией гостя); освободить номер.
* Для многоместного номера:
  + занять или освободить часть номера (дополнить или сократить список проживающих).
* Для приложения:
  + оформить регистрацию гостя в гостинице (при наличии свободного номера модифицировать состояние номера);
  + оформить освобождение номера (изменить состояние номера и вернуть в качестве результата сумму оплаты за проживание в номере);
  + вывести информацию о свободных номерах гостиницы;
  + (\*) определить занятость гостиницы (отношение числа занятых мест в номерах к общему числу мест в номерах), используя класс-итератор.

# Порядок выполнения работы

1. На основе описания задачи определить состав классов.
2. Разработать иерархию классов и схему их взаимодействия.
3. Для каждого класса определить его состояние и необходимые методы.
4. Разработать и отладить все классы, используя стандартную библиотеку STL. Приложение реализовать в виде простой диалоговой программы.
5. (\*) Разработать и отладить класс, реализующий работу приложения. Предусмотреть в классе возможность сохранения таблицы в файле и ее восстановления.
6. Разработать и отладить класс-итератор, с помощью которого реализовать дополнительные операции.
7. Повторить разработку контейнерного класса (вместе с необходимыми итераторами), используя собственные шаблоны классов.
8. (\*) Реализовать приложение с использование средств оконного интерфейса.

Примечание: пункты задания, отмеченные (\*), являются дополнительными.