Спроектировать объектную модель для заданной предметной области, а затем спроектировать иерархию исключительных ситуаций для спроектированной объектной модели.

1. Использовать: классы, наследование, интерфейсы, полиморфизм, инкапсуляция
2. Каждый класс, метод и переменная должны иметь исчерпывающее смысл название и информативный состав. Необходимо точно продумать, какие классы необходимы для решения задачи.
3. Наследование должно применяться только тогда, когда это имеет смысл.
4. Классы должны быть грамотно разложены по пакетам.
5. Задание представляет собой какую-то предметную область, в которой требуется выделить необходимую иерархию классов и реализовать ее с помощью ООП (используя наследование, если необходимо или реализовывая интерфейсы).
6. Выберите методы в иерархии, переопределять которые нельзя и запретите это
7. Выделите те методы, которые могут быть методами класса, а не объекта, используйте их в клиентском коде
8. В каждом классе должны быть поля и методы, которые вы посчитаете необходимыми. Программа должна создавать объекты различных классов в выделенной предметной области, объединять их в какой-то набор объектов. Как правило, задание требует выполнить поиск по заданным критериям объектов, удовлетворяющим условиям поиск из полученного набора объектов.
9. Код должен быть отформатирован и соответствовать Java Code Convention.
10. Должно быть оправданное применение Enum для соответствующих полей
11. Используйте модификатор static для метода/поля/секции инициализации, там, где это оправданно
12. Создать и использовать иерархию исключений: наследование от корректного класса, собственные специфические методы, корректное отнесение к Checked/Unchecked excpetions
13. Каждый класс, метод и переменная в иерархии исключений должны иметь исчерпывающее смысл название и информативный состав. Необходимо точно продумать, какие классы необходимы для решения задачи.
14. Наследование вашей иерархии исключений должно быть невозможно, за исключением базового класса
15. Предоставлен тестовый код (не меньше 5 разных ситуаций) в методе main, в котором во время вызовов методов объектов из объектной модели бросаются исключения, исключения обрабатываются осмысленно, блоки try/catch/finally/multicatch имеются.