

1. Написать структуру `Box` с полями `length`, `width`, `height` (целые числа — длина, ширина и высота коробки в см), `weight` (вещественное число — масса коробки в кг) и `value` (целое число — стоимость содержимого в копейках). Добавьте конструктор по значениям полей.

Следующие функции получают на вход массив коробок (и, возможно, что-то еще).

2. Напишите функцию, которая для массива коробок вычисляет суммарную стоимость содержимого всех коробок.
3. Напишите функцию, которая проверяет, что сумма длины, ширины и высоты всех коробок не превосходит заданного значения.
4. Напишите функцию, которая определяет максимальный вес коробки, объем которых не больше параметра `maxV`.
5. В предположении, что все коробки пустые, напишите функцию, которая проверяет, что все коробки массива можно вложить друг в друга по одной штуке (т.е. самая маленькая коробка должна содержаться в коробке побольше, эта свою очередь — в еще большей и т.д.). Считаем, что одна коробка может содержать другую, если длина, ширина и высота первой коробки больше аналогичных размеров второй.
6. Напишите оператор `==` сравнения двух коробок на равенство всех параметров.
7. Напишите операторы ввода/вывода (`>>` и `<<`) для коробок в произвольные потоки.
8. Оформите тип данных `Box` и указанные функции в виде модуля. Сделайте демонстрационный модуль и подключите к нему первый. Во втором модуле напишите функцию `main`, которая использует содержимое первого модуля.
9. Переделайте структуру `Box` в класс. Добавьте геттеры и сеттеры полей.
10. Напишите класс `Container`. Контейнер содержит коробки (используйте стандартный класс `vector`), он имеет длину, ширину и высоту, а также ограничение на максимальный вес своего содержимого. Добавьте конструктор по размерам и максимальному весу (создает пустой контейнер).
11. Напишите методы добавления и удаления коробки из контейнера (по индексу). Добавьте методы:
 - количество коробок в контейнере,
 - суммарный вес содержимого контейнера,
 - суммарная стоимость содержимого,
 - получение коробки по индексу,
 - добавление коробки в контейнер. Метод получает на вход коробку и проверяет, что суммарный вес коробок внутри позволяет добавить данную (размеры коробки не проверяем). Если ограничение по весу выполнено, то коробка добавляется, метод возвращает ее индекс. В противном случае выбрасывается исключение (создайте тип для него).
 - удаление коробки по индексу.
12. Напишите операторы ввода/вывода для класса `Container`.
13. В класс `Container` добавьте метод — оператор `[]`, который позволяет получить/изменить коробку по индексу.
14. Поместите класс `Container` в отдельный модуль. Подключите его к демонстрационному модулю. Функция `main` должна использовать объекты обоих классов `Box` и `Container`.
15. Создайте пространство имен и поместите в него классы `Box` и `Container`.