Список 01 – Лабораторная практика

Повторение

Предмед: Алгоритмизация и программирование

Преподаватель: Хольгер Эспинола Ривера

1. Вектор объектов

Рассмотрим некую веб-службу, реализованную в финансовой компании **traders.org**, которая имеет **кластеры** и **серверы баз данных**. Каждый **сервер баз данных** имеет следующие характеристики:

- **server name** (имя сервера): строка с максимум 20 буквами

- **database engine** (ядро базы данных): перечисление, вызываемое с 3 возможными вариантами строк (PostgreSQL, MySQL и Oracle)

- host name (имя хоста): строка, объединяющая имя компании «**traders.org**» + некоторую дополнительную информацию о строке, определенную пользователем.

- **IP address** (IP-адрес): имеет формат: «xxx.xxx.x.x» (пример: 192.168.1.1). Первые 3 группы чисел точно такие же, как у кластера (скопируйте IP-адрес соответствующего кластера и рассмотрите только первые группы чисел, используя `copy[strlen(copy)-4] = '\0';`), и добавьте к строке последнее число, которое определяется пользователем.

- **port** (порт): целочисленное значение

Действительно, у компании есть 2 **кластера**: **первичный кластер (Primary cluster)** и **кластер-реплика (Replica cluster)**. Каждый **кластер** имеет следующие характеристики:

- **cluster name** (имя кластера): строка, содержащая максимум 20 букв

- **total number of nodes** (общее количество узлов): целочисленное значение количества зарегистрированных серверов баз данных.

- **maximum capacity** (максимальная емкость): целочисленное значение максимального количества серверов баз данных.

- **IP address** (IP-адрес): имеет формат: «xxx.xxx.x.0/24» (пример: 192.168.1.0/24). Последняя цифра адреса всегда равна 0, а 24 указывает максимальную емкость.

- **vector of servers** (вектор серверов): содержит вектор серверов баз данных.

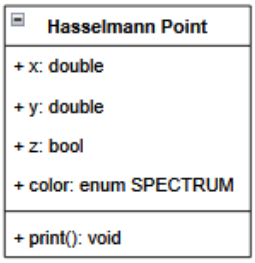
Следуйте следующим инструкциям:

[1]. Создайте классы **Cluster** и **Database Server** и нарисуйте диаграмму UML. Определите конструктор по умолчанию, конструктор по параметрам, конструктор по копии объекта, методы **setters** и **getters**. Все атрибуты для обоих классов имеют **private** доступ. Каждый кластер (первичный кластер и кластер реплики) являются объектами класса Cluster. В main.cpp необходимо определить эти 2 объекта с помощью конструктора кластера.

[2]. Определите в классе **Cluster** методы **set_vec_servers()** и **print_cluster()**. Первый метод требует вставки вектора объектов сервера базы данных в некоторый кластер. Второй метод требует вывода имени кластера и всех серверов базы данных с их атрибутами в формате, похожем на таблицу. Действительно, надо печать общего количества серверов.

2. Перегрузка операций

Для построения некоторой системы оптических точек в C++ определен тип **enum SPECTRUM** с тремя возможными значениями: **RED** и **BLUE**. Определен класс, называемый Hasselmann Point, определенный на диаграмме UML ниже:



[1]. Построить класс **Hasselmann Point**, конструктор по умолчанию, по параметрам, по копированию объекта, методы **setters** и **getters**. Учтите, что все атрибуты имеют привилегию доступа **private**. Определить функцию **print()** для отображения значений каждого атрибута.

[2]. Создайте 2 объекта класса **Hasselmann Point**. Сделайте перегрузку операции **+ и *.** Реализуйте глобальную функцию с привилегированным **friend**, вызываемым по значению (рассмотрите объекты **hp1** и **hp2** класса **Hasselmann Point**):

a) определить операцию + определяется как $result = hp1 + hp2$:

$result.x = \sqrt{(hp1.x)^2 + (hp2.x)^2}$; $result.y = \sqrt{(hp1.y)^2 + (hp2.y)^2}$;

$result.z = hp1.z \,||\, hp2.z$; $result.color = hp1.color$

b) определить операцию * определяется как $result = hp1 * hp2$:

$result.x = (hp1.x)(hp2.x)$; $result.y = (h1.y)(h2.y)$

$result.z = hp1.z \,\&\&\, hp2.z$; $result.color = hp2.color$

c) Распечатать атрибуты результата операции **+ и *