**ЛАБОРАТОРНАЯ РАБОТА №5**

Выделять классы на основе требований к ИС. Абстрактных классов нет. Генерировать код по классам на С++, все атрибуты соответствующего типа.

Для базы данных морского порта "Причал", вам потребуются следующие таблицы, поля, связи, атрибуты и операции:

**Таблицы и их поля:**

1. **Суда**
   * ID судна
   * Название
   * Тип судна
   * Водоизмещение
   * Флаг
2. **Прибытие\_Отправление**
   * ID регистрации
   * ID судна (связь с таблицей Суда)
   * Дата и время прибытия
   * Дата и время отправления
   * Причал
3. **Грузы**
   * ID груза
   * Описание
   * Вес
   * Тип груза
   * ID судна (связь с таблицей Суда)
4. **Склады**
   * ID склада
   * Местоположение
   * Вместимость
   * Тип хранения
5. **Персонал**
   * ID работника
   * ФИО
   * Должность
   * Отдел
6. **Финансы**
   * ID транзакции
   * Сумма
   * Дата
   * Тип операции
7. **Операции**
   * ID операции
   * Описание
   * ID работника (связь с таблицей Персонал)
   * ID судна (связь с таблицей Суда)
8. **Инфраструктура**
   * ID объекта
   * Название
   * Тип объекта
   * Состояние
9. **Безопасность**
   * ID инцидента
   * Описание
   * Дата и время
   * Меры реагирования

**Связи:**

* Суда связаны с Прибытие\_Отправление, Грузы и Операции.
* Персонал связан с Операциями.

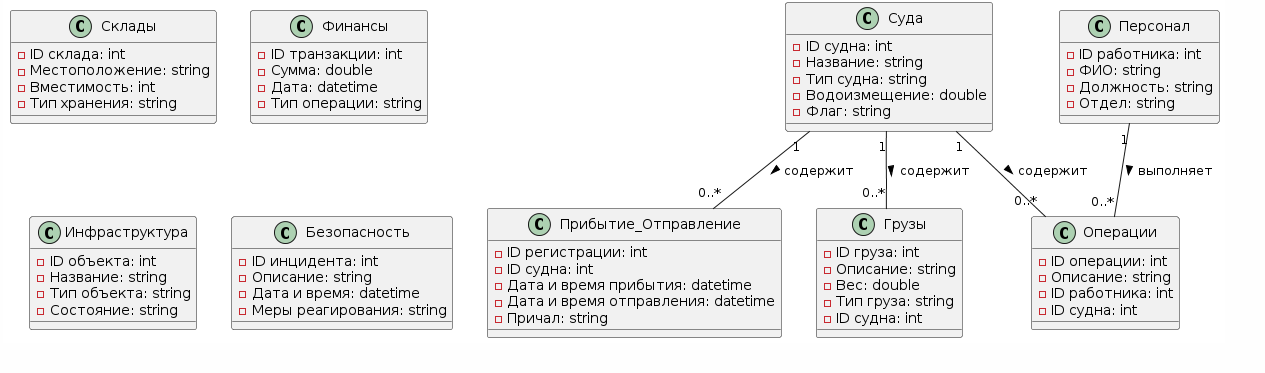
**Атрибуты:**

* Все ID должны быть уникальными и автоинкрементными.
* Даты и время должны быть в формате, соответствующем стандартам SQL.

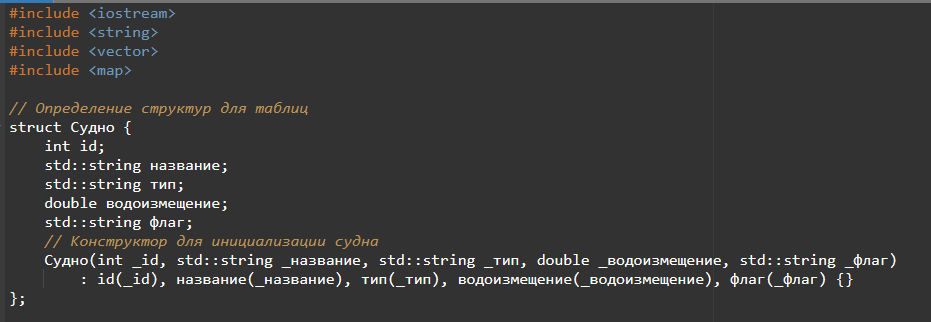
**Операции:**

* Регистрация прибытия/отправления судов в таблице Прибытие\_Отправление.
* Управление грузами и складами через таблицы Грузы и Склады.
* Финансовый менеджмент через таблицу Финансы.
* Управление персоналом через таблицу Персонал.
* Отчетность и анализ через запросы к соответствующим таблицам.
* Координация деятельности по погрузке и разгрузке грузов через таблицу Операции.
* Безопасность перемещения грузов через таблицу Безопасность.
* Навигационные услуги и управление инфраструктурой порта через таблицу Инфраструктура.

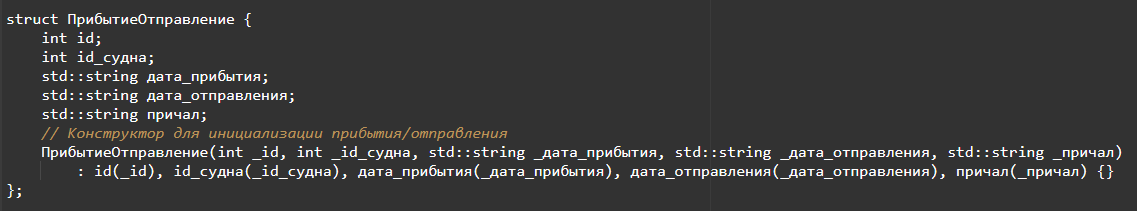
**Диаграмма классов**:



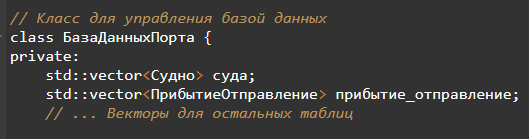
**Сгенерировали** программный код на С++ на основе диаграммы классов.



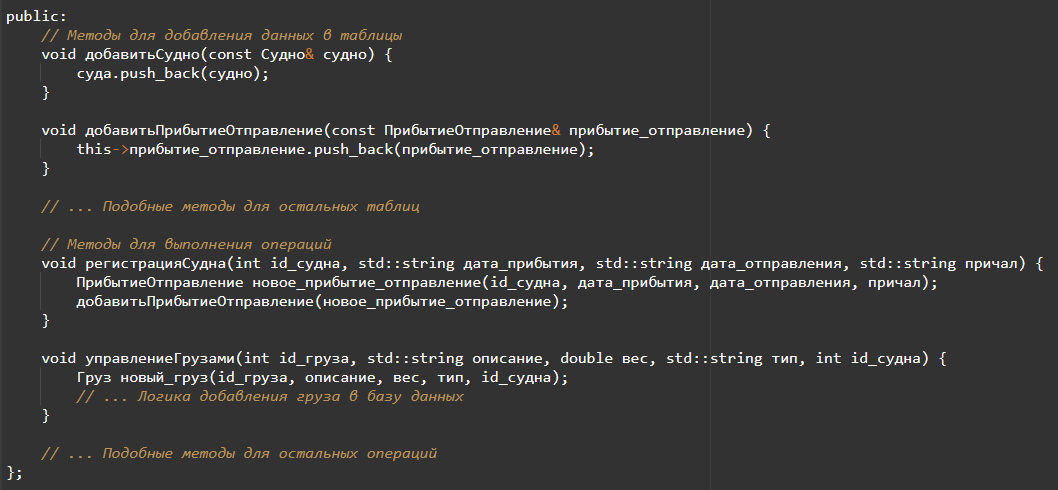
Структура судно

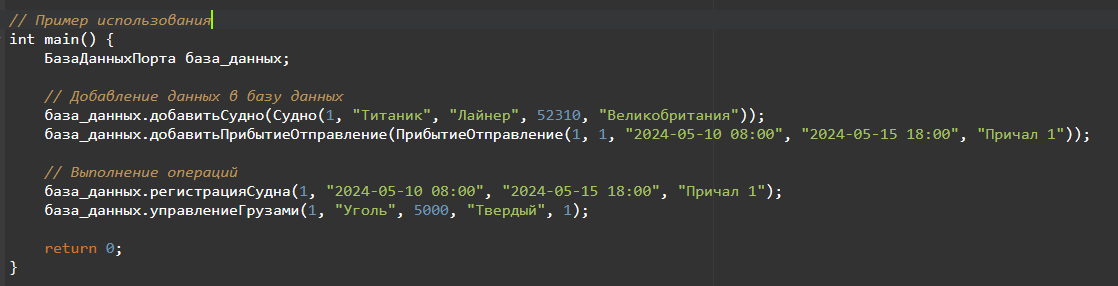


Структура Прибытия-отправления



Класс управления базы данных





Main

#include <iostream>

#include <string>

#include <vector>

#include <map>

// Определение структур для таблиц

struct Судно {

int id;

std::string название;

std::string тип;

double водоизмещение;

std::string флаг;

// Конструктор для инициализации судна

Судно(int \_id, std::string \_название, std::string \_тип, double \_водоизмещение, std::string \_флаг)

: id(\_id), название(\_название), тип(\_тип), водоизмещение(\_водоизмещение), флаг(\_флаг) {}

};

struct ПрибытиеОтправление {

int id;

int id\_судна;

std::string дата\_прибытия;

std::string дата\_отправления;

std::string причал;

// Конструктор для инициализации прибытия/отправления

ПрибытиеОтправление(int \_id, int \_id\_судна, std::string \_дата\_прибытия, std::string \_дата\_отправления, std::string \_причал)

: id(\_id), id\_судна(\_id\_судна), дата\_прибытия(\_дата\_прибытия), дата\_отправления(\_дата\_отправления), причал(\_причал) {}

};

// ... Подобные структуры для остальных таблиц

// Класс для управления базой данных

class БазаДанныхПорта {

private:

std::vector<Судно> суда;

std::vector<ПрибытиеОтправление> прибытие\_отправление;

// ... Векторы для остальных таблиц

public:

// Методы для добавления данных в таблицы

void добавитьСудно(const Судно& судно) {

суда.push\_back(судно);

}

void добавитьПрибытиеОтправление(const ПрибытиеОтправление& прибытие\_отправление) {

this->прибытие\_отправление.push\_back(прибытие\_отправление);

}

// ... Подобные методы для остальных таблиц

// Методы для выполнения операций

void регистрацияСудна(int id\_судна, std::string дата\_прибытия, std::string дата\_отправления, std::string причал) {

ПрибытиеОтправление новое\_прибытие\_отправление(id\_судна, дата\_прибытия, дата\_отправления, причал);

добавитьПрибытиеОтправление(новое\_прибытие\_отправление);

}

void управлениеГрузами(int id\_груза, std::string описание, double вес, std::string тип, int id\_судна) {

Груз новый\_груз(id\_груза, описание, вес, тип, id\_судна);

// ... Логика добавления груза в базу данных

}

// ... Подобные методы для остальных операций

};

// Пример использования

int main() {

БазаДанныхПорта база\_данных;

// Добавление данных в базу данных

база\_данных.добавитьСудно(Судно(1, "Титаник", "Лайнер", 52310, "Великобритания"));

база\_данных.добавитьПрибытиеОтправление(ПрибытиеОтправление(1, 1, "2024-05-10 08:00", "2024-05-15 18:00", "Причал 1"));

// Выполнение операций

база\_данных.регистрацияСудна(1, "2024-05-10 08:00", "2024-05-15 18:00", "Причал 1");

база\_данных.управлениеГрузами(1, "Уголь", 5000, "Твердый", 1);

return 0;

}