Отчет по коду "Lab4.cpp"

Описание

Этот код реализует систему управления прокатом самокатов с использованием классов и перечислений в C++. Код включает в себя следующие компоненты:

- Перечисление `ScooterState` для представления состояния самоката.
- Класс `Scooter` для управления отдельным самокатом.
- Класс `ScooterPool` для управления парком самокатов.

Перечисление `ScooterState`

```
enum class ScooterState {
    FREE,
    RENTED,
    IN_REPAIR
};
```

Перечисление `ScooterState` определяет три возможных состояния самоката:

- `FREE` : Самокат свободен и готов к аренде.
- `RENTED` : Самокат арендован.
- `IN_REPAIR` : Самокат находится на ремонте.

Класс `Scooter`

Поля

- `ScooterState state` : Состояние самоката.
- `int timesRented` : Количество раз, когда самокат был арендован.
- `int timesRepaired` : Количество раз, когда самокат был отремонтирован.

Конструктор

```
Scooter() : state(ScooterState::FREE), timesRented(0), timesRepaired(0) {}
```

Инициализирует новый объект `Scooter` в состоянии `FREE` с нулевыми значениями для счетчиков аренд и ремонтов.

Методы

- `void rent()` : Изменяет состояние самоката на `RENTED` и увеличивает счетчик аренд.
- `void returnScooter(bool needsRepair)` : Возвращает самокат, устанавливая его состояние в зависимости от параметра `needsRepair` .
- `void repair()` : Ремонтирует самокат, устанавливая его состояние в `FREE` и увеличивая счетчик ремонтов.
- `ScooterState getState() const` : Возвращает текущее состояние самоката.

Класс `ScooterPool`

Поля

- `std::vector<Scooter> scooters` : Вектор для хранения объектов `Scooter` .

Конструктор

```
ScooterPool(int n) {
    scooters.resize(n);
}
```

Создает парк самокатов из `n` самокатов.

Методы

- `Scooter* getScooter()` : Ищет первый свободный самокат, переводит его в состояние `RENTED` и возвращает указатель на него. Если свободных самокатов нет, возвращает `nullptr` .

- `void returnScooter(Scooter* scooter, bool needsRepair)` : Возвращает самокат и при необходимости отправляет его на ремонт.
- `void displayStatistics()` : Выводит общую статистику по количеству арендованных и отремонтированных самокатов.

Главная функция `main`

```
int main() {
    ScooterPool pool(10); // Pool of 10 scooters

    // Simulate some operations
    Scooter* scooter = pool.getScooter(); // Rent a scooter
    if (scooter != nullptr) {
        pool.returnScooter(scooter, true); // Return and need repair
    }

    scooter = pool.getScooter(); // Rent another scooter
    if (scooter != nullptr) {
        pool.returnScooter(scooter, false); // Return and no repair needed
    }

    pool.displayStatistics(); // Display statistics
    return 0;
}
```

В главной функции создается парк из 10 самокатов, затем моделируются операции аренды и возврата самокатов, после чего выводится статистика.

Заключение

Этот код предоставляет базовую реализацию системы проката самокатов, демонстрируя использование классов, перечислений и стандартных контейнеров C++. Он показывает основные операции с объектами и вывод статистики по использованию самокатов.