/\*III - 3. Разработать контейнерный класс «Депо», в котором список вагонов упорядочен по возрастанию

значений поля Номер. Проверить, есть ли в депо пассажирские вагоны типа купе с количеством

мест менее 36? Если есть, то распечатать информацию о вагоне с наименьшим номером.

Использовать STL контейнер list\*/

#pragma once

#include <iostream>

#include <fstream>

#include <string>

using std::cout;

using std::ifstream;

using std::string;

class Carriage

{

protected:

int number;

int release\_year;

public:

Carriage(int number = 0, int release\_year = 0) :

number(number), release\_year(release\_year) {}

void set\_number(int input) { number = input; }

void set\_release\_year(int input) { release\_year = input; }

int get\_number() { return number; }

int get\_release\_year() { return release\_year; }

virtual void print();

int compare(Carriage obj);

};

class Passenger\_Carriage : public Carriage

{

public:

enum passenger\_carriage\_type { NoType, WithSeats, Coupe, SecondClass };

private:

passenger\_carriage\_type type;

int number\_of\_seats;

bool presence\_of\_a\_shower\_module;

public:

Passenger\_Carriage(int number\_of\_seats, bool presence\_of\_a\_shower\_module, passenger\_carriage\_type type = NoType, int number = 0, int release\_year = 0) :

type(type), number\_of\_seats(number\_of\_seats), presence\_of\_a\_shower\_module(presence\_of\_a\_shower\_module), Carriage(number, release\_year) {}

Passenger\_Carriage(ifstream& file);

void set\_type(passenger\_carriage\_type input) { type = input; }

void set\_number\_of\_seats(int input) { number\_of\_seats = input; }

void set\_presence\_of\_a\_shower\_module(bool input) { presence\_of\_a\_shower\_module = input; }

passenger\_carriage\_type get\_type() { return type; }

int get\_number\_of\_seats() { return number\_of\_seats; }

bool get\_presence\_of\_a\_shower\_module() { return presence\_of\_a\_shower\_module; }

void print();

};

class Freight\_Carriage : public Carriage

{

public:

enum freight\_carriage\_type { NoType, tank, platform, gondola, refrigerator };//gondola - полувагон

private:

freight\_carriage\_type type;

int tonnage;

public:

Freight\_Carriage(int tonnage, freight\_carriage\_type type = NoType, int number = 0, int release\_year = 0) :

type(type), tonnage(tonnage), Carriage(number, release\_year) {}

Freight\_Carriage(ifstream& file);

void set\_type(freight\_carriage\_type input) { type = input; }

void set\_tonnage(int input) { tonnage = input; }

freight\_carriage\_type get\_type() { return type; }

int get\_tonnage() { return tonnage; }

void print();

};