

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа № 5

по дисциплине: «Объектно-ориентированное
программирование»

Выполнил: ст. группы ПВ-211
Медведев Дмитрий Сергеевич

Проверил:
Буханов Дмитрий Геннадьевич
Харитонов Сергей Дмитриевич

Белгород 2023 г.

Классы, виды отношений.

Наследование.

Вариант 8

Цель работы: получение теоретических знаний в области разработки классов, получение практических навыков реализаций классов и отношений между ними.

Задание к работе: В соответствии с вариантом $((\text{номер по списку} + 5) \% 10) + 1$ выполнить построение объектной модели (использовать не менее 5 объектов) заданной предметной области (задание 1), разработать диаграмму классов для описанной объектной модели (не менее 7 классов), и реализовать предложенные классы (задание 2).

Вариант 4 $((8+5)\%10)+1$

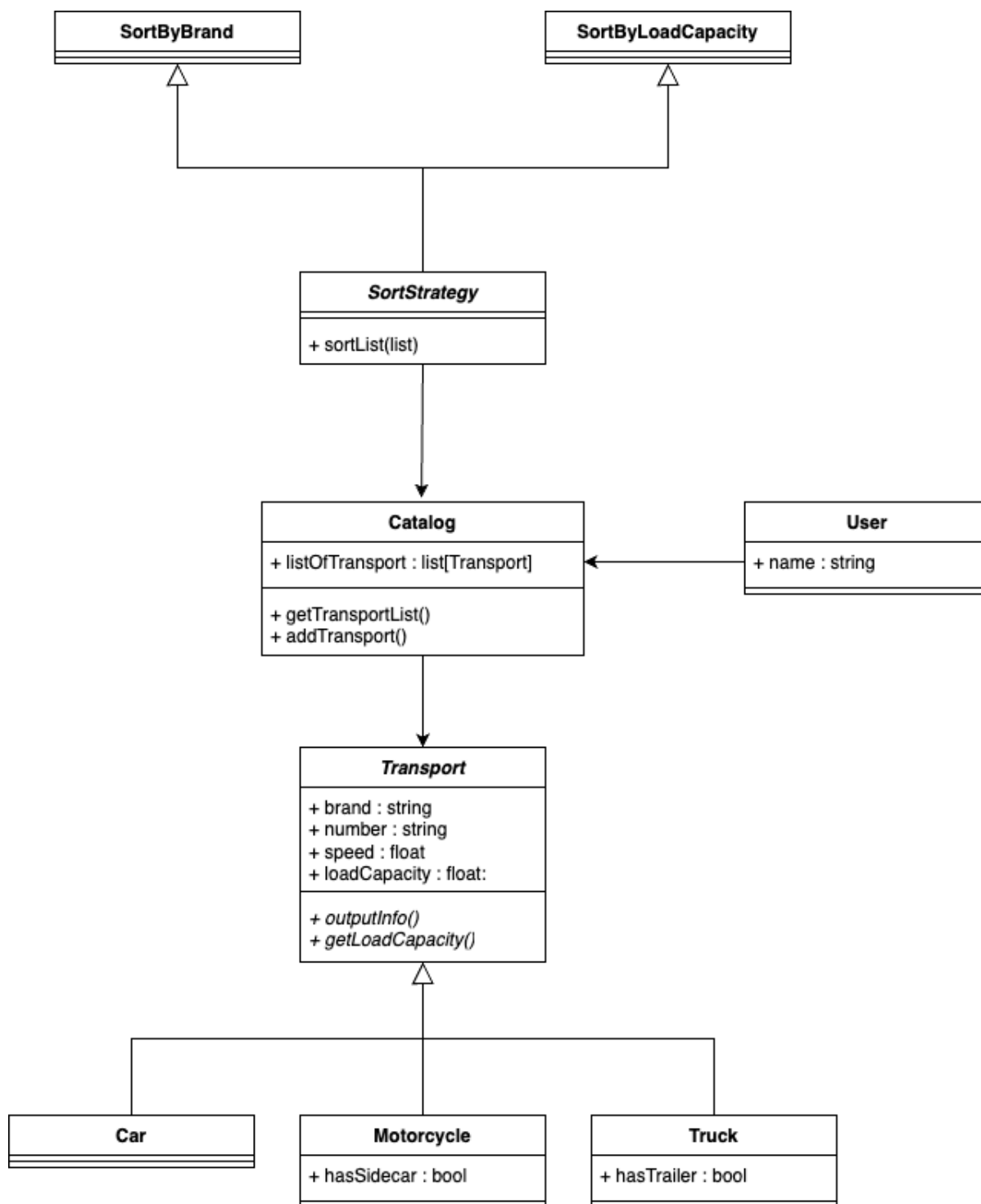
Задание 1

Программа учета парка автотранспорта.

Диаграмма объектов



Диаграмма классов



Задание 2

1. Создать абстрактный класс `Trans` с методами, позволяющими вывести на экран информацию о транспортном средстве, а также определить грузоподъёмность транспортного средства.

```
class Transport {  
protected:
```

```

    string brand;
    string number;
    float speed;
    float loadCapacity;

public:
    virtual void outputInfo() {
        cout << "Brand: " << brand;
        cout << "\nNumber: " << number;
        cout << "\nSpeed: " << speed;
        cout << "\nLoad capacity: " << loadCapacity;
    }

    virtual float getLoadCapacity() = 0;
};

```

2. Создать производные классы: Легковая_машина (марка, номер, скорость, грузоподъёмность), Мотоцикл (марка, номер, скорость, грузоподъёмность, наличие коляски, при этом если коляска отсутствует, то грузоподъёмность равна 0), Грузовик (марка, номер, скорость, грузоподъёмность, наличие прицепа, при этом если есть прицеп, то грузоподъёмность увеличивается в два раза) со своими методами вывода информации на экран, и определения грузоподъёмности.

```

class Motorcycle : public Transport {
private:
    bool hasSidecar;
public:
    Motorcycle(string brand, string number, float speed, float
loadCapacity, bool hasSidecar) {
        this->brand = brand;
        this->number = number;
        this->speed = speed;
        this->loadCapacity = loadCapacity;

        this->hasSidecar = hasSidecar;
        if (!hasSidecar) {
            loadCapacity = 0;
        }
    }

    void outputInfo() override {
        Transport::outputInfo();
        cout << "\nHas sidecar: ";
        if (hasSidecar)
            cout << "Yes";
        else
            cout << "No";

        cout << '\n';
    }
};

```

```

    }

    float getLoadCapacity() override {
        return loadCapacity;
    }
};

class Truck : public Transport {
    bool hasTrailer;
public:
    Truck(string brand, string number, float speed, float loadCapacity,
    bool hasTrailer) {
        this->brand = brand;
        this->number = number;
        this->speed = speed;
        this->loadCapacity = loadCapacity;

        this->hasTrailer = hasTrailer;
        if (hasTrailer) {
            loadCapacity *= 2;
        }
    }

    void outputInfo() override {
        Transport::outputInfo();
        cout << "\nHas trailer: ";
        if (hasTrailer)
            cout << "Yes";
        else
            cout << "No";

        cout << '\n';
    }

    float getLoadCapacity() override {
        return loadCapacity;
    }
};

```

3. Создать базу (массив) из n машин, вывести полную информацию из базы на экран, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

```

int main() {
    Car car1("VAZ", "T532TA031", 140, 1400);
    Car car2("Ford", "M385BK031", 180, 2200);
    Motorcycle m1("BMW", "A001AA031", 220, 800, false);
    Motorcycle m2("BMW", "A002AA031", 180, 800, true);
    Truck t1("Kamaz", "B001BB031", 160, 10000, false);
    Truck t2("Kamaz", "B002BB031", 160, 20000, true);

    Transport *data[10];
    data[0] = &car1;
}

```

```

data[1] = &car2;
data[2] = &m1;
data[3] = &m2;
data[4] = &t1;
data[5] = &t2;

for (int i = 0; i <= 5; i++) {
    data[i]->outputInfo();
    cout << '\n';
}

cout << "Удовлетворяют требованиям грузоподъемности:\n";
for (int i = 0; i <= 5; i++) {
    if (data[i]->getLoadCapacity() > 1500) {
        data[i]->outputInfo();
        cout << '\n';
    }
}
}

```

Вывод: в ходе лабораторной работы мы получили теоретические знания в области разработки классов, получили практические навыков реализации классов и отношений между ними.