Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра экономической математики, информатики и статистики (ЭМИС)

ПЕРЕГРУЗКА ФУНКЦИЙ. ШАБЛОНЫ ФУНКЦИЙ.

Отчет по лабораторной работе по дисциплине «Объектно-ориентированное программирование»

Студент группы 549  Баулин С.К.  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.  
Кандидат физико-математических наук, доцент кафедры ЭМИС \_\_\_\_\_\_\_ Шельмина Е. А. «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

\_\_\_\_\_\_

оценка

Томск 2020

Лабораторная работа №4

Разработка классов на языке С++

Цель работы

Получить навыки разработки классов на С++.

Краткий теоретический материал

Классы представляют абстрактные типы данных с открытым интерфейсом и скрытой внутренней реализацией. В классах реализованы базовые принципы объектно-ориентированного программирования (ООП):

1) абстракция данных;

2) инкапсуляция – в классах объединяются данные и методы (функции) для работы с ними, так как лишь через методы возможен доступ к сокрытым данным класса;

3) наследование – в производных классах наследуются члены базового класса;

4) полиморфизм – возможность использования одних и тех же методов для работы с различными объектами базового и порожденных им классов.

Определение простейшего класса без наследования имеет вид:

class имя\_класса {

// по умолчанию раздел private – частные члены класса

public: // открытые функции и переменные класса

};

Имя класса является новым типом данных. Понятию переменной данного типа соответствует понятие объекта класса. Список членов класса включает описание данных и функций. Функции, описания которых находятся в определении класса, называются функциями-членами класса.

Переменные и функции, объявленные в разделе класса по умолчанию или явно как private, имеют область видимости в пределах класса. Их можно сделать видимыми вне класса, если объявить в разделе public:. Обычно переменные объявляются в разделе private, а функции в разделе public.

Классами в С++ являются также структуры (struct) и объединения (union). В отличие от класса члены структуры по умолчанию являются открытыми, а не закрытыми. Кроме того, объединения не могут наследоваться и наследовать.

При реализации функциональной части класса могут быть использованы функции-члены класса, конструкторы, деструкторы, функции-операторы. Функции класса всегда объявляются внутри класса. Определение функции может находиться и внутри класса.

Такие функции называются inline-функциями. Обычно определения функций-членов класса помещаются вне класса. При этом перед именем функции указывается имя\_класса:: .

тип имя\_класса:: имя\_функции (описание аргументов)

{ /\*тело функции\*/ }

Вызов функций осуществляется одним из двух способов:

имя\_объекта.имя\_функции(аргументы);

указатель\_на\_объект -> имя\_функции(аргументы);

Обращение к данным объекта класса осуществляется с помощью функций, вызываемых из объектов. При этом функции-члену класса передается скрытый указатель this на объект, вызывающий функцию.

Функции-«друзья» класса, объявляемые в классе со спецификатором friend, указатель this не содержат. Объекты, с которыми работают такие функции, должны передаваться в качестве их аргументов. Это обычные функции языка С++, которым разрешен доступ к закрытым членам класса.

Задания

Задание 1. Разработать класс для объекта Airbus: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:

а) список рейсов для заданного пункта назначения;

б) список рейсов для заданного дня недели;

в) список рейсов для заданного дня недели, время вылета для которых больше заданного.

Ход работы

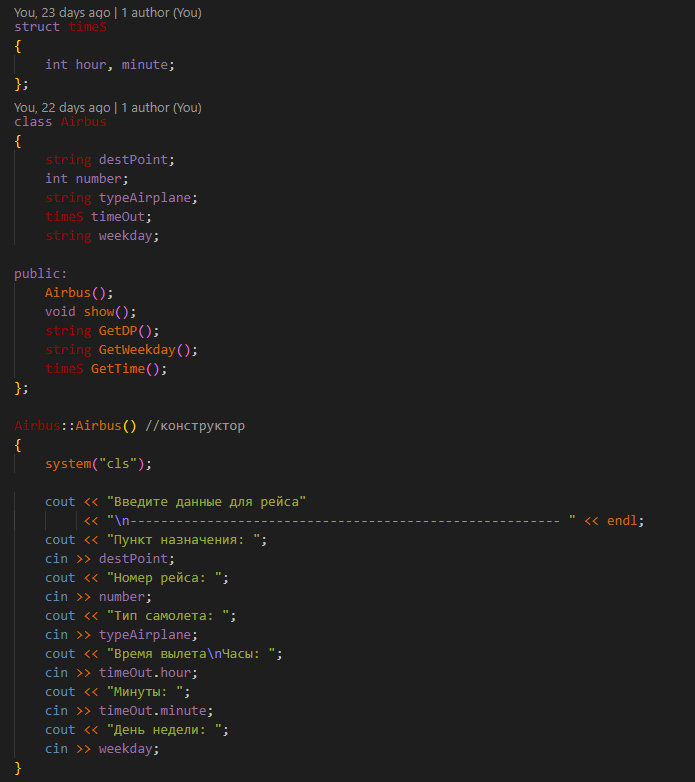
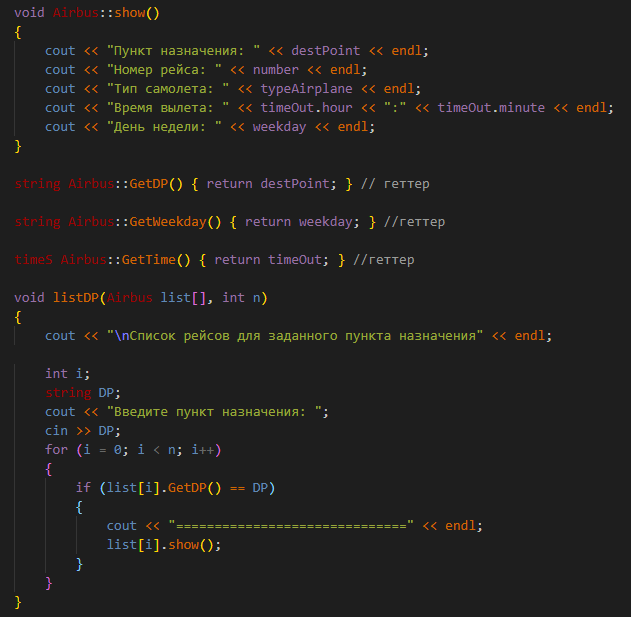
Результат работы показан на рисунках 1 – 6.

Рисунок 1 – Скриншот описания объекта класса Airbus и его конструктор

Вывод

Получены навыки разработки классов на С++.

Рисунок 2 – Скриншот геттеров класса, функции вывода рейсов, а так же вывод рейсов для заданного пункта назначения

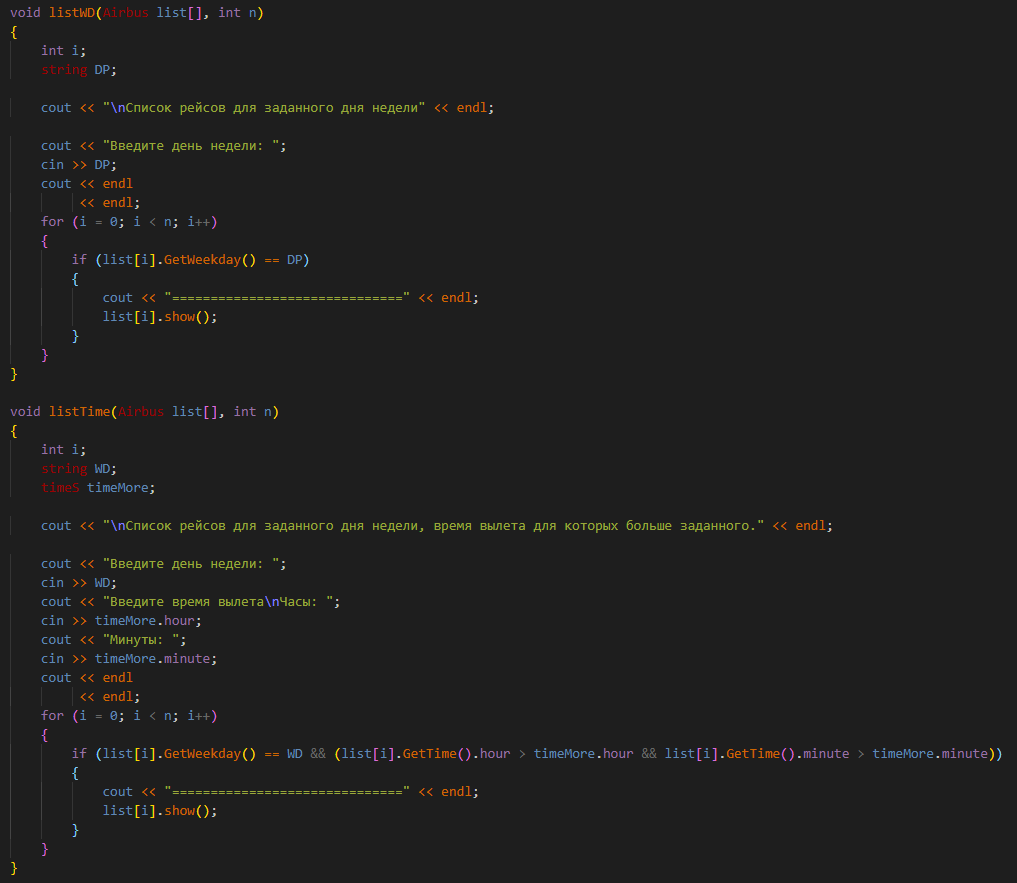
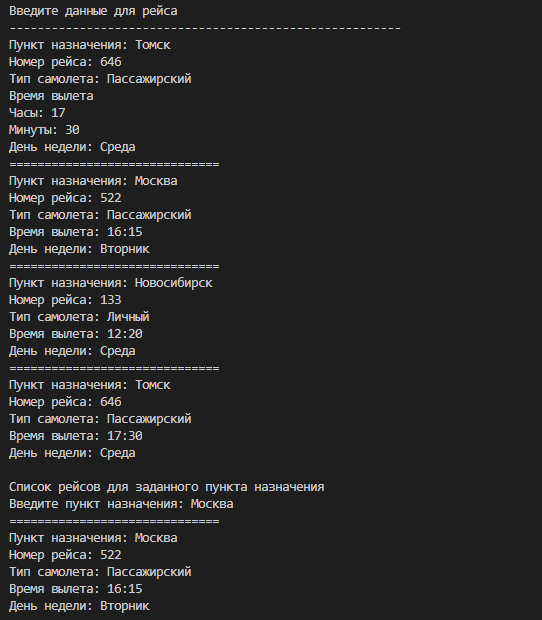
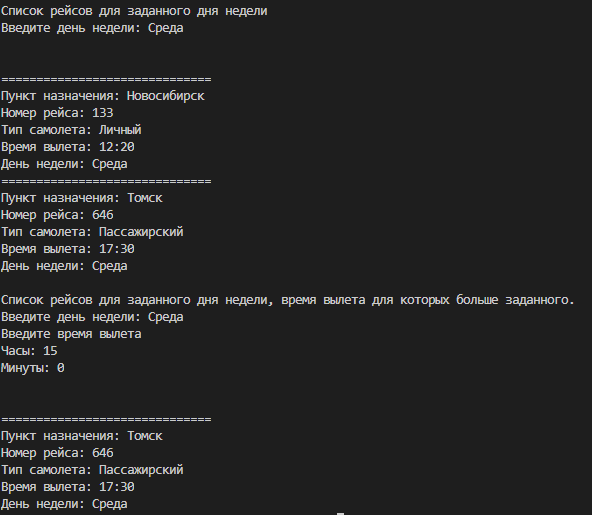
Рисунок 3 – Скриншот функций

Рисунок 4 – Скриншот функции main

Рисунок 5 – Скриншот начала результата работы программы

Рисунок 6 – Скриншот конца результата работы программы