

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.

оценка «\_\_» \_\_\_\_\_ 2020 г.

## Лабораторная работа №4

### Разработка классов на языке C++

#### Цель работы

Получить навыки разработки классов на C++.

#### 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 C++ 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 C++, 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.

```

You, 23 days ago | 1 author (You)
struct timeS
{
    int hour, minute;
};

You, 22 days ago | 1 author (You)
class Airbus
{
    string destPoint;
    int number;
    string typeAirplane;
    timeS timeOut;
    string weekday;

public:
    Airbus();
    void show();
    string GetDP();
    string GetWeekday();
    timeS GetTime();
};

Airbus::Airbus() //конструктор
{
    system("cls");

    cout << "Введите данные для рейса"
    | << "\n----- " << endl;
    cout << "Пункт назначения: ";
    cin >> destPoint;
    cout << "Номер рейса: ";
    cin >> number;
    cout << "Тип самолета: ";
    cin >> typeAirplane;
    cout << "Время вылета\nЧасы: ";
    cin >> timeOut.hour;
    cout << "Минуты: ";
    cin >> timeOut.minute;
    cout << "День недели: ";
    cin >> weekday;
}

```

Рисунок 1 – Скриншот описания объекта класса Airbus и его конструктор

### Вывод

Получены навыки разработки классов на C++.

```

void Airbus::show()
{
    cout << "Пункт назначения: " << destPoint << endl;
    cout << "Номер рейса: " << number << endl;
    cout << "Тип самолета: " << typeAirplane << endl;
    cout << "Время вылета: " << timeOut.hour << ":" << timeOut.minute << endl;
    cout << "День недели: " << weekday << endl;
}

string Airbus::GetDP() { return destPoint; } // геттер

string Airbus::GetWeekday() { return weekday; } //геттер

timeS Airbus::GetTime() { return timeOut; } //геттер

void listDP(Airbus list[], int n)
{
    cout << "\nСписок рейсов для заданного пункта назначения" << endl;

    int i;
    string DP;
    cout << "Введите пункт назначения: ";
    cin >> DP;
    for (i = 0; i < n; i++)
    {
        if (list[i].GetDP() == DP)
        {
            cout << "=====" << endl;
            list[i].show();
        }
    }
}

```

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

```

void listWD(Airbus list[], int n)
{
    int i;
    string DP;

    cout << "\nСписок рейсов для заданного дня недели" << endl;

    cout << "Введите день недели: ";
    cin >> DP;
    cout << endl
        << endl;
    for (i = 0; i < n; i++)
    {
        if (list[i].GetWeekday() == DP)
        {
            cout << "=====" << endl;
            list[i].show();
        }
    }
}

void listTime(Airbus list[], int n)
{
    int i;
    string WD;
    timeS timeMore;

    cout << "\nСписок рейсов для заданного дня недели, время вылета для которых больше заданного." << endl;

    cout << "Введите день недели: ";
    cin >> WD;
    cout << "Введите время вылета\nЧасы: ";
    cin >> timeMore.hour;
    cout << "Минуты: ";
    cin >> timeMore.minute;
    cout << endl
        << endl;
    for (i = 0; i < n; i++)
    {
        if (list[i].GetWeekday() == WD && (list[i].GetTime().hour > timeMore.hour && list[i].GetTime().minute > timeMore.minute))
        {
            cout << "=====" << endl;
            list[i].show();
        }
    }
}

```

Рисунок 3 – Скриншот функций

```
int main()
{
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    Airbus *list;
    int n;
    cout << "Введите кол-во рейсов: ";
    cin >> n;
    list = new Airbus[n];

    for (int i = 0; i < n; i++)
    {
        cout << "=====" << endl;
        list[i].show();
    }

    listDP(list, n);
    listWD(list, n);
    listTime(list, n);
    delete[] list;

    return 0;
}
```

Рисунок 4 – Скриншот функции main

```
Введите данные для рейса
-----
Пункт назначения: Томск
Номер рейса: 646
Тип самолета: Пассажирский
Время вылета
Часы: 17
Минуты: 30
День недели: Среда
=====
Пункт назначения: Москва
Номер рейса: 522
Тип самолета: Пассажирский
Время вылета: 16:15
День недели: Вторник
=====
Пункт назначения: Новосибирск
Номер рейса: 133
Тип самолета: Личный
Время вылета: 12:20
День недели: Среда
=====
Пункт назначения: Томск
Номер рейса: 646
Тип самолета: Пассажирский
Время вылета: 17:30
День недели: Среда

Список рейсов для заданного пункта назначения
Введите пункт назначения: Москва
=====
Пункт назначения: Москва
Номер рейса: 522
Тип самолета: Пассажирский
Время вылета: 16:15
День недели: Вторник
```

Рисунок 5 – Скриншот начала результата работы программы



```
Список рейсов для заданного дня недели
Введите день недели: Среда

=====
Пункт назначения: Новосибирск
Номер рейса: 133
Тип самолета: Личный
Время вылета: 12:20
День недели: Среда
=====
Пункт назначения: Томск
Номер рейса: 646
Тип самолета: Пассажирский
Время вылета: 17:30
День недели: Среда

Список рейсов для заданного дня недели, время вылета для которых больше заданного.
Введите день недели: Среда
Введите время вылета
Часы: 15
Минуты: 0

=====
Пункт назначения: Томск
Номер рейса: 646
Тип самолета: Пассажирский
Время вылета: 17:30
День недели: Среда
```

Рисунок 6 – Скриншот конца результата работы программы