Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**КЛАССЫ И ИНКАПСУЛЯЦИЯ**

Отчёт по лабораторной работе №3 по дисциплине

«Объектно-ориентированное программирование»

Студент группы 588-1

\_\_\_\_\_\_\_\_\_\_ / Чан Хыу Тхай

«\_» \_\_\_\_\_\_\_\_\_2022 г.

Руководитель старший научный

сотрудник, доцент каф. КСУП

\_\_\_\_\_\_\_\_\_ / Горяинов А.Е.

«\_\_\_» 2022 г.

Томск 2022

1. **Цель работы**

Цель этой лабораторной работы — ознакомиться с классами и инкапсуляцией, а также получить практические навыки решения связанных с ними задач объектно-ориентированного программирования.

1. **Теоретические основы**

**Класс** – это просто независимая подпрограмма, в которой есть свой набор переменных и функций (обязательно изучи урок про функции). В основной программе мы можем создать экземпляр класса (он же называется объект) и пользоваться теми инструментами, которые имеются в классе. Использование классов позволяет:

* Разделить сложную программу на отдельные независимые части
* Создавать удобные библиотеки
* Использовать свои “наработки” в другом проекте, не переписывая один и тот же код
* Облегчить и упростить программу, если в ней используются повторяющиеся конструкции и алгоритмы

Классы очень похожи на структуры (читай урок по структурам), как по объявлению, так и по использованию, но класс является гораздо более мощной единицей языка благодаря механизмам наследования и прочим ООП-штукам.

Класс объявляется при помощи ключевого слова class и содержит внутри себя члены класса – переменные и функции:

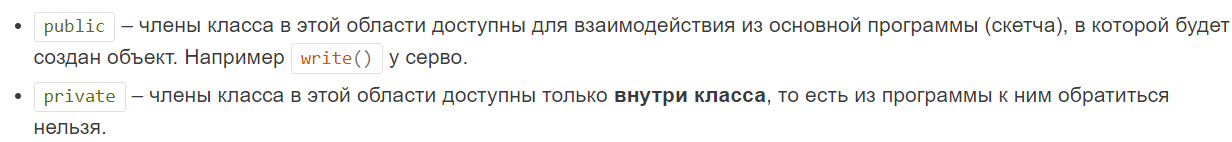
class Имя\_класса {

член1;

член2;

};

Важное отличие от структуры: содержимое класса делится на области: публичные и приватные. Они определяются при помощи ключевых слов public и private, область действует до начала следующей области или до закрывающей фигурной скобки класса:



class Имя\_класса {

public:

// список членов, доступных в программе

private:

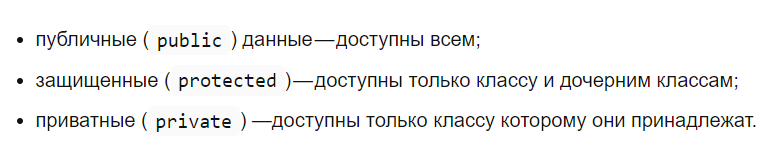
// список членов для использования внутри класса

};

**Инкапсуляция** это набор инструментов для управления доступом к данным или методам которые управляют этими данными. С детальным определением термина “инкапсуляция” можно ознакомиться в моей предыдущей публикации на Хабре по этой ссылке. Эта статья сфокусирована на примерах инкапсуляции в Си++ и Си.

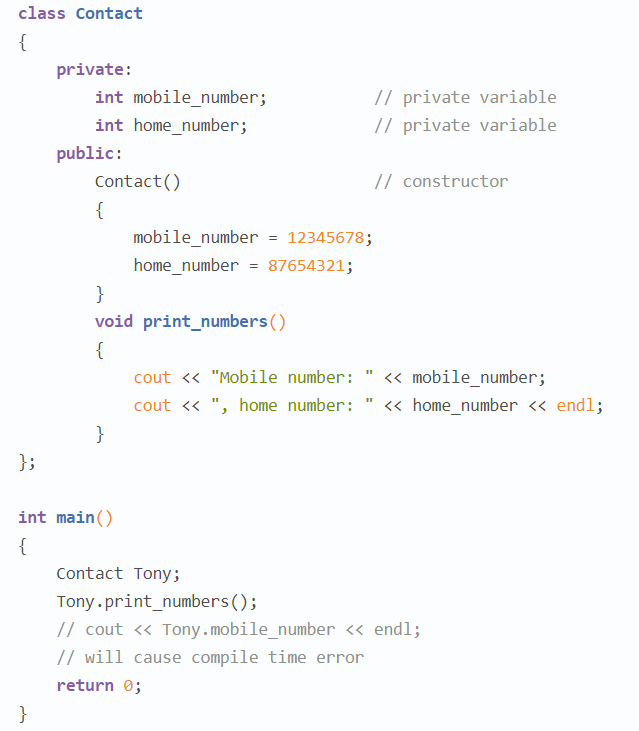
По умолчанию, в классе (class) данные и методы приватные (private); они могут быть прочитаны и изменены только классом к которому принадлежат. Уровень доступа может быть изменен при помощи соответствующих ключевых слов которые предоставляет Си++.

В Си++ доступно несколько спецификаторов, и они изменяют доступ к данным следующим образом:



**Пример инкапсуляции**

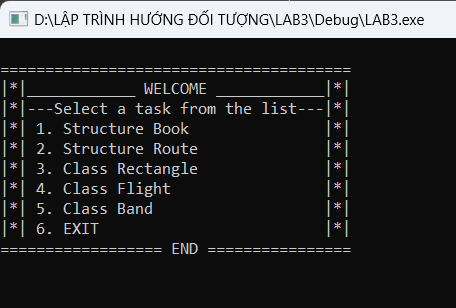
В классе Contact, публичные переменные и методы доступны из основной программы (main). Приватные переменные и методы могут прочитаны, вызваны или изменены только самим классом.



Попытка напечатать или изменить приватную переменную mobile\_number из основной программы (main) вызовет ошибку при компиляции потому как доступ к приватным данным в классе ограничен.

1. **Ход работы**

Ниже представлен интерфейс лабораторной работы:



В первой задаче нужно создать структуру книги Book. В структуре должны быть поля "Название", "Год издания", "Количество страниц", массив строк "Авторы" (не более десяти), поле "Количество авторов" (рис. 1).

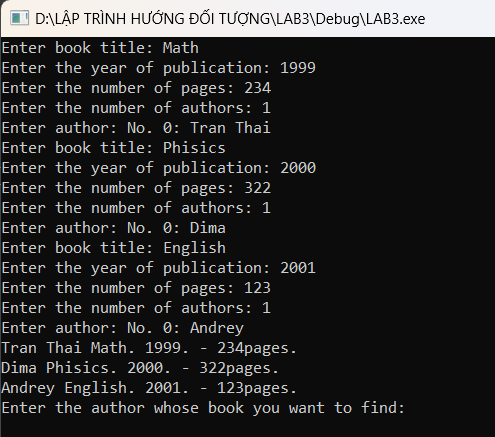


Рис. 1 – Создать структуру книги

Оттуда мы можем ввести информацию об имени автора для поиска и вывода всей информации о книге этого автора. (рис. 2).

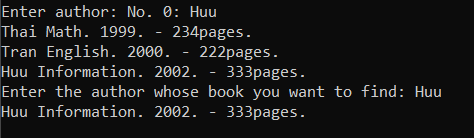


Рис. 2 – Информация о книге разыскиваемого автора

Условие проверки на количество авторов менее 10 проверено и выполнено (рис. 3).

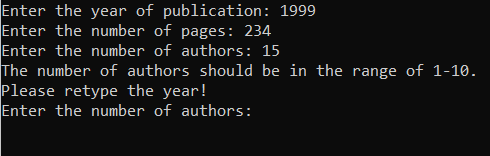


Рис. 3 – Условия проверки количества авторов

В второй задаче нужно создать функцию WriteBookToConsole(Book& book), которая выводит данные книги на экран. Аналогично вводу, вам будет нужно организовать вывод правильного количества авторов на экран. Пример работы функции (рис. 4).

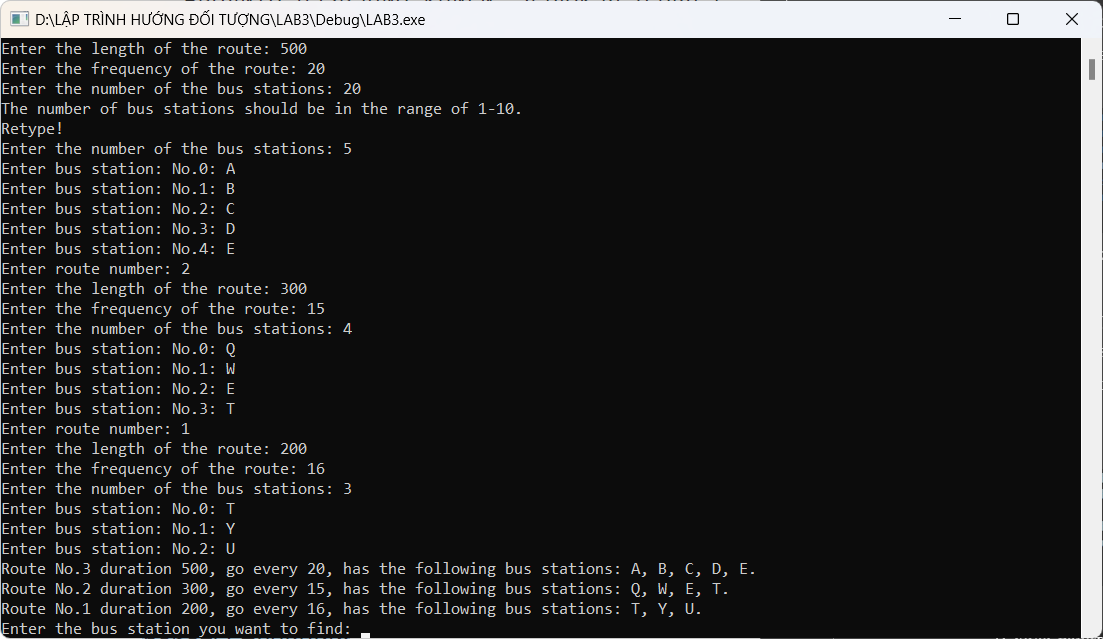


Рис. 4 – Информация о маршрутах и ​​направлениях

Условие проверки на количество остановка менее 10 проверено и выполнено (рис. 5).

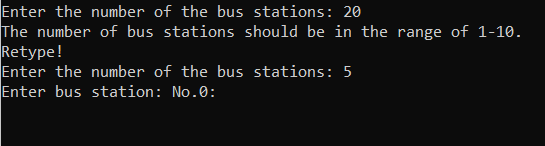


Рис. 5 – Условия проверки количества остановка

Оттуда мы можем ввести информацию о маршруте для поиска и отображения всей информации об этом маршруте. (рис. 6).

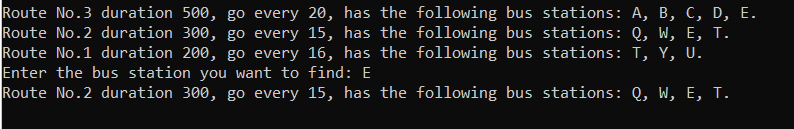


Рис. 6 – Информация о маршруте для поиска

Следующая задача — продемонстрировать функцию DemoRectangleWithPoint(). Нужно добавить код, который будет отображать данные о прямоугольниках массива (рис. 7).

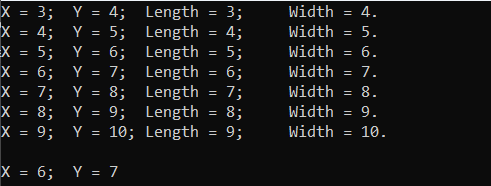


Рис. 7 – Демонстрационная функция прямоугольника

Следующая задача — продемонстрировать функцию DemoRectangleWithPoint(). Нужно добавить код, который будет вычислять среднее значение всех центров прямоугольника и отображать его на экране. Пример вывода (расчетные значения для показанных выше координат) (рис. 8).

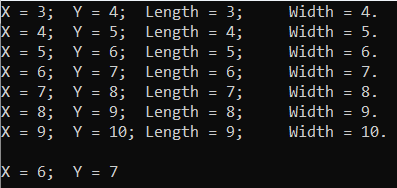


Рис. 8 – Результат вычисления среднего значения всех центров прямоугольников

Для следующей задачи необходимо создать структуру Flight, в которой хранятся поля номера рейса, пункт отправления, пункт назначения, время отправления и время прибытия (компоненты структуры Time). Для структур необходимо создать функции конструктора и сеттера с соответствующими валидациями данных (время прихода не может быть раньше времени отправления) (рис. 9).

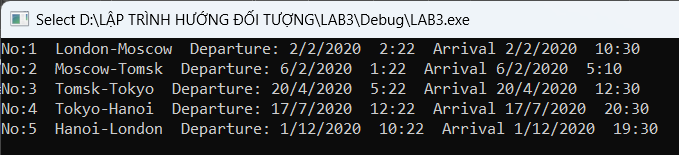


Рис. 9 – Информация о полете

Необходимо написать функцию GetFlightTimeMinutes(Flights & Flights), которая принимает на вход объект полета и возвращает время полета в минутах (или в новом объекте Time).

И нужно продемонстрировать, как работает функция, вызвав функцию для каждого элемента массива в функции DemoFlightWithTime() (рис. 10).

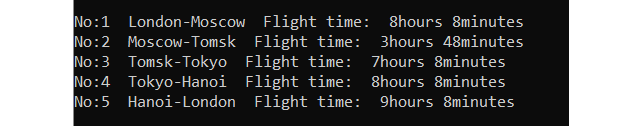


Рис. 10 – Информация о время полете

Следующей задачей необходимо создать функцию Demoband() для вывода информации о музыкальных продуктах. (рис. 11).

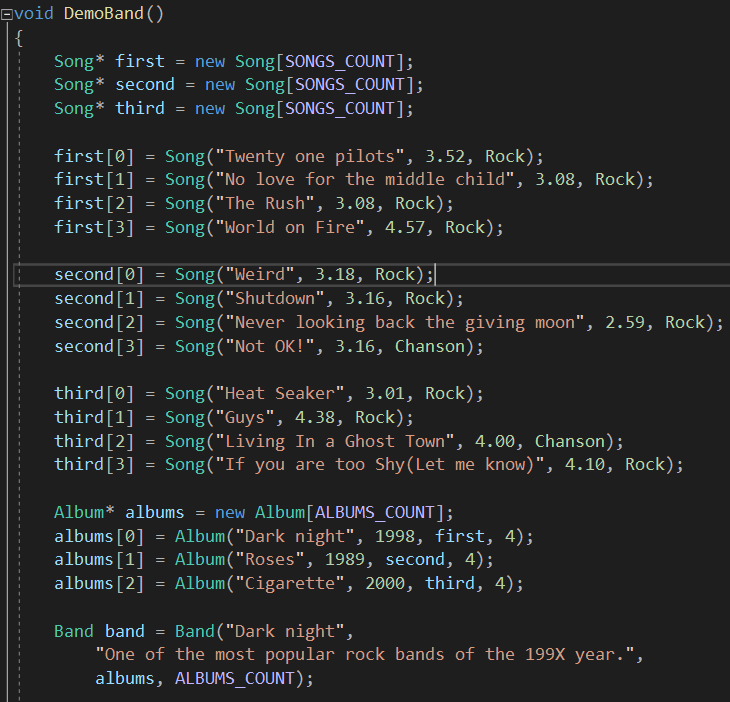


Рис. 11 – Функция Demoband() для вывода информации о музыкальных продуктах

И при поиске возвращаются результаты для искомой песни (рис. 12).

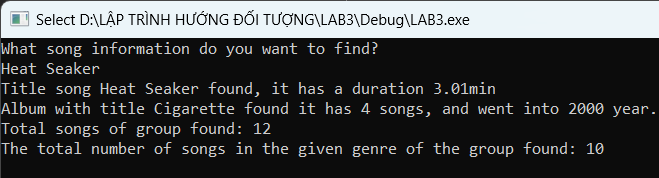


Рис. 12 – Результаты возвращены для искомой песни

1. **Вывод**

В ходе работы в лаборатории были выполнены все задачи, разработана природа «Классы и инкапсуляция» в объектно-ориентированном программировании и набор функций для работы с ней.