

# РАЗРАБОТВАНЕ НА МНОГОМОДУЛНИ ПРОГРАМИ

## 1. Цел на упражнението

Усвояване на технологията за разработване на многомодулни програми.

## 2. Многомодулни програми

Голяма част от програмите, разработвани на C++ се състоят от повече от един модул. Всеки модул е оформен като отделен .CPP или .OBJ файл. Някои от модулите могат да се намират в създадена вече библиотека - .LIB файл. Използването на заглавни файлове (**header**) е основен принцип в програмирането на C++. Заглавният файл включва декларации на външни променливи, прототипи на функции, дефиниции на класове. Файловете, които използват тези декларации, трябва само да включат съответния заглавен файл чрез директивата **#include** за препроцесора. За да се избегне многократното включване на един заглавен файл, се използват директивата **#define** и условните директиви **#ifdef** и **#ifndef**, чрез които може да се провери дали даден идентификатор вече е дефиниран.

## 3. Структура на многомодулна програма

```
.....
                                VEHICLE.H
.....

#ifndef VEHICLE_H
#define VEHICLE_H

class Vehicle
{
    // Декларации на класа
}; // end class Vehicle

#endif

.....
                                VEHICLE.CPP
.....

#include "vehicle.h"

// дефиниции на методите на клас Vehicle
void Vehicle :: initialize( int inWheels, float inWeight)
{ ..... }
int Vehicle :: getWheels( void )
{ ..... }
float Vehicle :: getWeight( void )
{ ..... }
float Vehicle :: wheelLoading( void )
{ ..... }
```

```

.....
CAR.H
.....

```

```

#ifndef CAR_H
#define CAR_H

class Car : public Vehicle
{
    // Декларации на класа
}; // end class Car

#endif

```

```

.....
CAR.CPP
.....

```

```

#include "vehicle.h"
#include "car.h"

// дефиниции на методите на клас Car
void Car :: initialize(int inWheels, float inWeight, int people)
{ ..... }

int Car :: passengers(void)
{ ..... }

```

```

.....
TRUCK.H
.....

```

```

#ifndef TRUCK_H
#define TRUCK_H

class Truck : public Vehicle
{
    // Декларации на класа
}; // end class Truck

#endif

```

```

.....
TRUCK.CPP
.....

```

```

#include "vehicle.h"
#include "truck.h"

// дефиниции на методите на клас Tuck
void Truck :: initTruck( int howMany, float maxLoad)
{ ..... }
float Truck :: efficiency( void )
{ ..... }
int Truck :: passengers( void )
{ ..... }

```

```

.....
                                MAIN.CPP
.....

#include <iostream>

using namespace std;

#include "vehicle.h"
#include "car.h"
#include "truck.h"

void main( void )
{
    Vehicle tricycle;
    .....

    Car sedan;
    .....

    Truck zil;
    .....

} // end main

```

#### 4. Задачи за изпълнение

##### Задача 1:

Декларациите и дефинициите на методите на класовете **Vehicle**, **Car** и **Truck** са представени във файла H:\SKELET\EX8\_1.CPP.

Да се създаде проект с име EX81.

От файла EX8\_1.CPP да се създаде:

- заглавен файл **vehicle.h**, съдържащ декларациите на клас **Vehicle** (структурата на файла е представена по-горе);
- заглавен файл **car.h**, съдържащ декларациите на клас **Car**;
- заглавен файл **truck.h**, съдържащ декларациите на клас **Truck**;
- файл **vehicle.cpp**, съдържащ дефинициите на клас **Vehicle** (структурата на файла е представена по-горе);
- файл **car.cpp**, съдържащ дефинициите на клас **Car**;
- файл **truck.cpp**, съдържащ дефинициите на клас **Truck**;
- файл **main.cpp**, съдържащ главната програма.

Във всички файлове, съдържащи дефинициите на класовете и главната програма, чрез директивата **#include** да се включат съответните заглавни файлове, съдържащи декларациите на класовете.

Създадените файлове да се добавят към проекта EX81.

Проектът да се компилира и изпълни.

### **Задача 2:**

Декларациите и дефинициите на методите на класовете **Parent**, **Child** и **GrandChild** са представени във файла `H:\SKELET\EX8_2.CPP`.

Да се създаде проект с име `EX82`.

От файла `EX8_2.CPP` да се създаде:

- заглавен файл `parent.h`, съдържащ декларациите на клас **Parent**;
- заглавен файл `child.h`, съдържащ декларациите на клас **Child**;
- заглавен файл `grchild.h`, съдържащ декларациите на клас **GrandChild**;
- файл `parent.cpp`, съдържащ дефинициите на клас **Parent**;
- файл `child.cpp`, съдържащ дефинициите на клас **Child**;
- файл `grchild.cpp`, съдържащ дефинициите на клас **GrandChild**;
- файл `mainpr.cpp`, съдържащ главната програма.

Във всички файлове, съдържащи дефинициите на класовете и главната програма, чрез директивата `#include` да се включат съответните заглавни файлове, съдържащи декларациите на класовете.

Създадените файлове да се добавят към проекта `EX82`.

Проектът да се компилира и изпълни.

### **Задача 3:**

Декларациите и дефинициите на методите на класовете **Point**, **Circle** и **Cylinder** са представени във файла `H:\SKELET\EX8_3.CPP`.

Да се създаде проект с име `EX83`.

От файла `EX8_3.CPP` да се създадат заглавни файлове, съдържащи декларациите на отделните класове. Дефинициите на методите на класовете да бъдат отделени в съответни `.cpp` файлове.

Във всички файлове, съдържащи дефинициите на класовете и главната програма, чрез директивата `#include` да се включат съответните заглавни файлове, съдържащи декларациите на класовете.

Създадените файлове да се добавят към проекта `EX83`.

Проектът да се компилира и изпълни.

### **Обща схема на шаблона за описание на класове:**

ИМЕ НА КЛАСА:

НАСЛЕДЯВА ОТ:

ПРИЯТЕЛСКИ НА:

ОПИСАНИЕ ХАРАКТЕРНИТЕ ОСОБЕНОСТИ НА КЛАСА:

ТИП НА КЛАСА:

БАЗОВ ЗА:

ИМА ПРИЯТЕЛСКИ:

ЧЛЕНОВЕ ДАННИ:

**име:**

**тип:**

**съдържание:**

ФУНКЦИИ-ЧЛЕНОВЕ (МЕТОДИ):

**<прототип>**

**<общо описание действието на функцията>**

**<структура и тип на параметрите>**

**<резултати>**

**<особености при използването>**