

7. ПРОИЗВОДНИ КЛАСОВЕ, НАСЛЕДЯВАНЕ И ПОЛИМОРФИЗЪМ

7.1. ДЕКЛАРИРАНЕ НА КЛАС КАТО ПРОИЗВОДЕН

Всеки клас може да бъде деклариран като производен клас (клас наследник) на други вече декларирани класове, които, с това, стават негови базови класове. Понятията базов и производен клас са относителни, тъй като всеки производен клас може да бъде базов за други класове.

Даден клас D се декларира като производен на класа B както следва:

```
class D : [public|private|protected] B {
```

```
};
```

Пред името на базовия клас може да се постави една от ключовите думи **public**, **private** или

protected

(

private

е подразбираща се, а

protected

тук е равнозначна на

private

). Тяхното значение

ще бъде изяснено по-надолу.

С декларирането на даден клас като производен, той ще получи достъп до компонентите на базовия клас, регламентиран от правила, които са разгледани по-нататък.

Програма 7.1. е примерна програма за дефиниране и използване на производен клас. В нея е дефиниран един производен клас с име **TheCar**, който има един базов клас с име **Car**

Програма 7.1. Базов клас Car и производен клас TheCar

```
#include
```

```
#include
```

```
#include
```

```
// Декларация на клас Car, който е базов за класа TheCar
```

```
class Car {
```

```
public:
```

```
void makeCar(char *,int);
```

```
void printCar();
```

private:

char *mark; //Марка

int year; //Година на производство

};

// Дефиниция на метод makeCar на класа Car

void Car::makeCar(char *a,int b)

{

mark=new char[strlen(a)+1];

strcpy(mark,a);

year=b;

}

// Дефиниция на метод printCar на класа Car

```
void Car::printCar()
```

```
{
```

```
cout
```