**Наследяване**

Наследяване – създаване на нов клас (наречен клас- наследник) на базата на вече съществуващ клас, наречен родителски.

Класът-наследник наследява всички данни и методи на родителския клас, но може да промени някои от тях, както и да добави нови.

Множеството от всички родителски класове и техните наследници се нарича йерархия от класове.

class Tochka { public: float x, y; }; class Krug: public Tochka { public: float r; float Obikolka(); float Lice(); }; class Prusten: public Krug { public: float r\_otvor; float Obikolka(); float Lice(); };

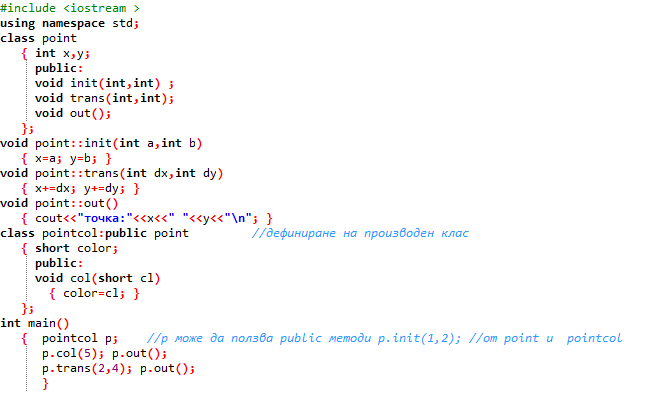
**Видимост на елементите на класа**

• Ако клас наследява public даден родител, то методите и данните на родителския клас са със същата видимост и в класа наследник.

• Ако клас наследява private даден родител, то методите и данните на родителския клас са private в класа наследник.

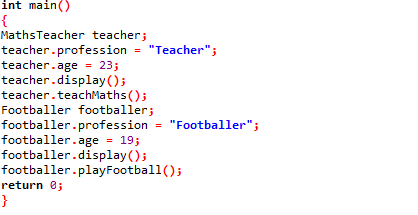
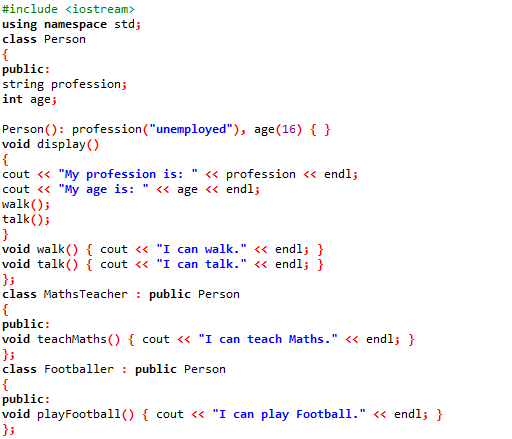
• Методи и данни, дефинирани като protected в класа, са видими само в методите на класа и класовете, които са негови наследници, но не и извън тях.

**Пример 1**  изясняващ същността на наследяването



В производния клас **pointcol** могат да се ползват само **public** компоненти, но не и private.

**Пример 2**



Задача 1

Създайте клас Parent с наследник клас Child, Father и Mother и изведете годините на детето и родителите.

Задача 2

Да се напише програма, която дефинира клас People, определящ човек по име и единен граждански номер (ЕГН), а също производен клас Student на класа People, който определя понятието студент като човек, който има факултетен номер и среден успех. Да се дефинира обект от клас Student и се изведе дефинираният обект.

Задача 3

Съставете базов клас триъгълник с четири елемента дължина на съответните страни и конструктори, който инициализират и валидират входните данни. Направете функция на базовия клас за намиране на лицето на триъгълника по хереоновата хормула. Създайте:  
– Производен клас „равнобедрен триъгълник“ и предефинирайте конструктора.  
– Производен клас „равностранен триъгълник“ и предефинирайте конструктора. Напишете втора функция за изчисляване на лицето на триъгълник, която използва директно формулата за лице на равностранен триъгълник: a\*a\*sqrt(3)/4.

Пример 2

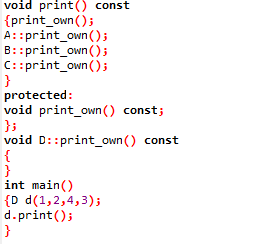
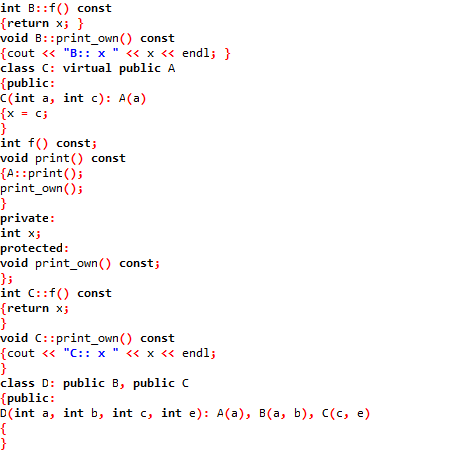
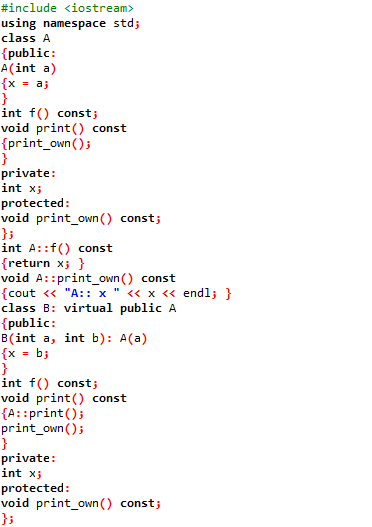
Да се създаде клас Point описващ точка в равнината. Задаването на стойност на координатите да става с подходящи методи (член-функции). Да се създаде производен клас Square описващ квадрат, зададена с център пресечна точка на диагоналите и дължина на диагонала (число с плаваща запетая). Задаването на параметрите на обект от клас Square става по време на конструиране (като параметри на конструктора). Да се създаде член-функция на класа за изчисляване на периметъра на квадрата.



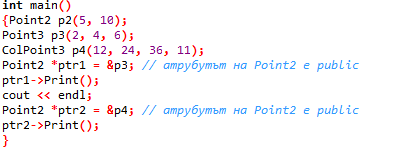
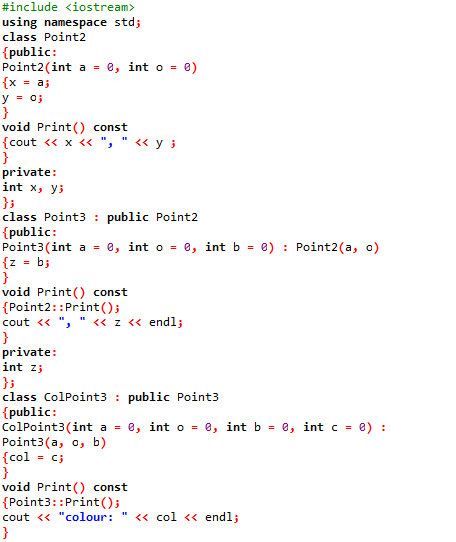
**Виртуални класове**

Многократното наследяване на клас води от една страна до затруднен достъп до многократно наследените членове, а от друга до поддържане на множество копия на член-данните на многократно наследения клас, което не е ефективно. Преодоляването на недостатъците на многократното наследяване на клас се осъществява чрез използването на т.н. виртуални основни класове. Чрез тях се дава възможност да се “поделят” основни класове. Когато един клас е виртуален, независимо от участието му в няколко списъка на основни класове, се създава само едно негово копие.

Декларацията на основен клас като виртуален се осъществява като в декларацията на производния клас заедно с името и атрибута за област на основния клас се укаже и запазената дума virtual.



Така класът А е обявен за виртуален. Кaзва се, че B и C наследяват класа A виртуално. Виртуалното наследяване на класа A от класовете B и C оказва влияние само на производните на B и C класове. То не променя поведението на самите класове B и C.



И в трите класа е дефинирана функция Print() без параметри и от тип void. В главната функция са дефинирани три обекта: p2, p3 и p4 от класове Point2, Point3 и ColPoint3 съответно. Освен това са дефинирани указатели и ptr1 и ptr2 към класа Point2. Указателят ptr1 е инициализиран с адреса на обекта p3 от класа Point3, а ptr2 – с адреса на обекта p4 от класа ColPoint3. Тъй като атрибутът за област на Point2 в Point3 е public и атрибутът за област на Point3 в ColPoint3 също е public, преобразуванията са допустими. Обръщението: ptr1->Print(); извежда първите две координати на точката p3, а ptr2- >Print(); - първите две координати на точката с цвят p4, т.е. изпълнява се Print() на класа Point2 и в двата случая. Още по време на компилация член-функцията Print() на Point2 е определена като функция на обръщенията ptr1->Print() и ptr2->Print(). Определянето става от типа Point2 на указателите ptr1 и ptr2. Вр