

1

ВИРТУАЛНИ КЛАСОВЕ. ВИРТУАЛНИ ФУНКЦИИ

1. Виртуални базови класове
2. Виртуални функции
3. Абстрактни базови класове
4. Полиморфизъм
5. Пример

Виртуални класове. Виртуални функции

05.04.2023г.

2

1. Виртуални базови класове

- ❖ Един клас може да се използва като базов за няколко други класове.
- ❖ В резултат на множественото наследяване могат да възникнат ситуации, при които някои производни класове наследяват многократно едни и същи компоненти.
- ❖ Многократното наследяване на едни и същи компоненти може да бъде предотвратено чрез използване на **виртуални класове**.

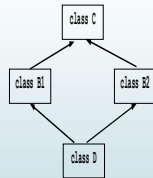
Виртуални класове. Виртуални функции

05.04.2023г.

3

1. Виртуални базови класове

```
class C
{ ..... };
class B1 : virtual public C
{ ..... };
class B2 : virtual public C
{ ..... };
class D : public B1, public B2
{ ..... };
```



Виртуални класове. Виртуални функции

05.04.2023г.

4

1. Виртуални базови класове

```
class C
{
public:
    int x;
    C (int a)
    { x=a; }
    void displayC ( void )
    { cout << "C - X = " << x << endl; }
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

5

1. Виртуални базови класове

```
class B1 : virtual public C
{
public:
    B1 ( int );
    void displayB1 ( void )
    { cout << "B1 - X = " << x << endl; }
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

6

1. Виртуални базови класове

```
class B2 : virtual public C
{
public:
    B2 ( int );
    void displayB2 ( void )
    { cout << "B2 - X = " << x << endl; }
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

7

1. Виртуални базови класове

```
class D : public B1, public B2
{
public:
    D ( int );
    void displayD ( void )
    { cout << "D - X = " << x << endl; }
};
```

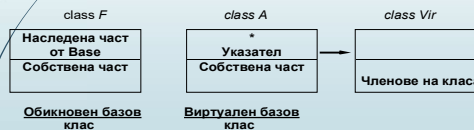
Виртуални класове, Виртуални функции

05.04.2023г.

8

1. Виртуални базови класове

- Виртуалният базов клас е общ за всички производни от него класове.
- Всеки производен клас съдържа две части: наследена от базовия му клас и собствена.
- При виртуалните класове наследената част е единствена и обща за всички производни класове, наследили производния клас.

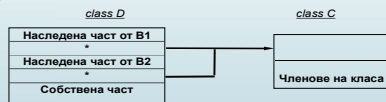
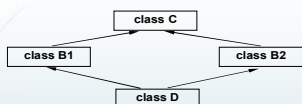


Виртуални класове, Виртуални функции

05.04.2023г.

9

1. Виртуални базови класове



Виртуални класове, Виртуални функции

05.04.2023г.

10

2. Виртуални функции

- В един производен клас и в негови базови класове могат да се дефинират методи с еднакви прототипи.
- Дали да се изпълни методът на производния клас или този на базовия клас зависи от типа на обекта или от типа на указателя (ако се използва указател към обект), чрез който се извиква методът.

Виртуални класове, Виртуални функции

05.04.2023г.

11

2.1. Механизъм на динамичното свързване

- На указател към обект от базов клас може да се присвои указател към обект от производен клас;
- Формален параметър на функция, който е указател към обект от базов клас, може да се замени с фактически параметър, който е указател към обект от производен клас;
- Виртуалните функции се извикват чрез указател или алтернативно име към обект от базовия клас.
- Указателят, чрез който се активира виртуална функция, може да приема стойности на указатели към обекти на производни класове.

Виртуални класове, Виртуални функции

05.04.2023г.

12

2.1. Механизъм на динамичното свързване

```
class Base
{
public:
    virtual void mess1( void )
    { cout << "Base - message1"<<endl; }
    virtual void mess2( void )
    { cout << "Base - message2"<<endl; }
    void mess3( void )
    { cout << "Base - message3"<<endl; }
};
```

Виртуални класове, Виртуални функции

05.04.2023г.

2.1. Механизъм на динамичното свързване

13

```
class Derived : public Base
{
public:
    virtual void mess1( void )
    { cout << "Derived-message1"<<endl; }
    void mess2( int ); // Не е виртуален
    void mess3( void )
    { cout << "Derived-message3"<<endl; }
};
```

Виртуални класове, Виртуални функции

05.04.2023г.

2.1. Механизъм на динамичното свързване

14

- Предимството на виртуалните функции е, че детайлите на реализацията са скрити за програмиста.
- При промени в наследствената йерархия е необходимо само да се добавят декларациите за новия клас.
- Класът ще бъде включен автоматично в механизма на динамичното свързване при компилиране на програмата.

Виртуални класове, Виртуални функции

05.04.2023г.

2.2. Дефиниране на виртуални функции

15

Виртуалните методи имат следните особености:

- Виртуален метод се декларира чрез ключовата дума **virtual**, поставена пред неговото име;
- Статичните методи не могат да бъдат виртуални;
- Виртуални методи не могат да се декларират като приятелски на други класове;
- Ако в даден клас бъде дефиниран виртуален метод и след това в производен на дадения клас бъде дефиниран метод със същия прототип, то методът в производния клас ще бъде също виртуален, дори ако модификаторът **virtual** бъде пропуснат.

Виртуални класове, Виртуални функции

05.04.2023г.

2.2. Дефиниране на виртуални функции

16

- Виртуална функция може да бъде само член-функция на клас.
- Виртуалните функции се извикват чрез **алтернативно име** или **указател към обект** от базовия клас.
- Базовият клас трябва да бъде от вида **public**.
- Производен клас наследява виртуалните функции на базовия клас и може да декларира свои собствени виртуални функции.
- Ако производен клас декларира свой вариант на виртуална функция на базовия клас, трябва да има съвпадение в прототипите.

Виртуални класове, Виртуални функции

05.04.2023г.

2.2. Дефиниране на виртуални функции

17

```
class First
{
public:
    virtual void fun( void )
    { cout << "Base class"; }
};
```

Виртуални класове, Виртуални функции

05.04.2023г.

2.2. Дефиниране на виртуални функции

18

```
class Second : public First
{
public:
    virtual void fun( void )
    { cout << "Derived class"; }
};
```

Виртуални класове, Виртуални функции

05.04.2023г.

19

2.2. Дефиниране на виртуални функции

Чрез използване на виртуални методи се постигат следните ефекти:

1. Ако даден виртуален метод не е дефиниран в един произведен клас, но бъде извикан, то автоматично се извиква виртуалният метод от неговия базов клас.

Виртуални класове. Виртуални функции

05.04.2023г.

20

2.2. Дефиниране на виртуални функции

Чрез използване на виртуални методи се постигат следните ефекти:

2. Ако в базовия клас също не е дефиниран такъв метод, търсенето продължава в следващото ниво на йерархията на класовете, докато бъде намерен извиканият метод или докато се изчерпят всички класове в йерархията.

Виртуални класове. Виртуални функции

05.04.2023г.

21

2.2. Дефиниране на виртуални функции

Чрез използване на виртуални методи се постигат следните ефекти:

3. Извикването на виртуален метод чрез указателя води до изпълнение на този метод, който е дефиниран за обекта (по-точно за класа на обекта), към който сочи указателят в момента на извикването, а не до изпълнение на метода, дефиниран за класа, който е тип на указателя.

Виртуални класове. Виртуални функции

05.04.2023г.

22

2.3. Виртуални деструктори

```
class First
{
public:
    virtual void fun( void )
        { cout <<"Base class"; }
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

23

2.3. Виртуални деструктори

```
class Second : public First
{
public:
    virtual void fun( void )
        { cout << "Derived class"; }
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

24

2.3. Виртуални деструктори

- Ако класът има виртуална функция и деструктор, най-добре е да се обяви деструкторът за виртуален.
- Това спестява проблемите по определяне на подходящия деструктор.

Виртуални класове. Виртуални функции

05.04.2023г.

25

3. Абстрактни базови класове

Неопределен виртуален метод е този, който има само декларация, но няма дефиниция.

```
class Abstr
{
public:
    Abstr( void );
    virtual void fun1( void ) = 0;
    virtual void fun2( void );
};
```

Клас, в който е деклариран поне един неопределен виртуален метод, се нарича абстрактен клас.

Виртуални класове, Виртуални функции

05.04.2023г.

26

3. Абстрактни базови класове

- ❖ Обекти на абстрактни класове не могат да бъдат създавани;
- ❖ Абстрактните класове могат да се използват само като базови за други класове;
- ❖ Абстрактните класове не могат да се използват като тип на параметри и тип на върнатата стойност на функциите;
- ❖ Методите на абстрактните класове могат да се извикват от техните конструктори, но извикване на чиста виртуална функция не е възможно и води до грешка по време на изпълнение на програмите;
- ❖ Допустимо е дефиниране и използване на указатели към абстрактни класове и псевдоними на абстрактни класове.

Виртуални класове, Виртуални функции

05.04.2023г.

27

4. Полиморфизъм

- Едни и същи действия (в общ смисъл) се реализират по различен начин в зависимост от обектите, върху които се прилагат, т.е. **действията са полиморфични**.
- В C++ полиморфизмът се реализира чрез виртуални методи.
- За да се реализира едно полиморфично действие, класовете на обектите, върху които то ще се прилага, трябва да имат общ корен, т.е. да бъдат производни на един и същ клас.

Виртуални класове, Виртуални функции

05.04.2023г.

28

4. Полиморфизъм

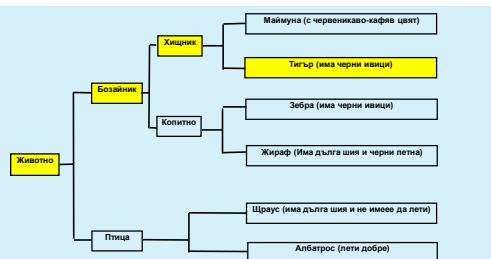
- В класа трябва да бъде дефиниран виртуален метод, съответстващ на полиморфичното действие.
- Във всеки от производните класове този метод може да бъде предефиниран съобразно особеностите на конкретния клас.
- Активирането на полиморфичното действие трябва да става чрез указател към базовия клас, на който могат да се присвояват адресите на обекти на който и да е клас от йерархията.

Виртуални класове, Виртуални функции

05.04.2023г.

29

5. Пример



Виртуални класове, Виртуални функции

05.04.2023г.

30

5. Пример

- Да се състави йерархия от класове, отразяваща представената класификация;
- Да се дефинира виртуална функция, която извежда характеристиката на животното за съответния клас от йерархията;
- Да се напише главна програма, която изгражда списък от обекти от всички класове в йерархията;
- Да се разработи функция, която обхожда списъка и извежда информация за признаците на включените в него обекти.

Виртуални класове, Виртуални функции

05.04.2023г.

31

КОРЕКЦИИ, КОИТО ТРЯБВА ДА БЪДАТ НАПРАВЕНИ В ТЕКСТА НА ПРОГРАМАТА, ПУБЛИКУВАН В ЛЕКЦИЯТА

```
#include <iostream.h>      #include <iostream>
#include <conio.h>
#include <string.h>        using namespace std;
```

Навсякъде, където в дефиницията на функцията
print() има

getch(); да се замени с cin.get() - продължение
след натискане на <Enter>

Виртуални класове. Виртуални функции

05.04.2023г.

32

Декларации на класовете

```
class Animal
{
public:
    Animal( char *k = " Животно " );
    virtual ~Animal( )
    { delete kind; }
    virtual void print( void );
    void link( Animal * );
    void printlist( Animal * );
protected:
    char *kind;
    Animal *next;
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

33

Декларации на класовете

```
class Boza : public Animal
{
public:
    Boza( char *r = " Бозайник " );
    ~Boza( )
    { delete rod; }
    virtual void print( void );
private:
    char *rod;
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

34

Декларации на класовете

```
class Hisht : public Boza
{
public:
    Hisht( char *v = " Хищник " );
    ~Hisht( )
    { delete vid; }
    virtual void print( void );
private:
    char *vid;
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

35

Декларации на класовете

```
class Tigar : public Hisht
{
public:
    Tigar( char *n = " Тигър " );
    ~Tigar( )
    { delete name; }
    virtual void print( void );
private:
    char *name;
};
```

Виртуални класове. Виртуални функции

05.04.2023г.

36

Дефиниции на методите

```
Animal :: Animal( char *k )
{
    next=0;
    kind = new char[strlen(k)+1];
    strcpy( kind, k );
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

37

Дефиниции на методите

```
void Animal :: link( Animal *pt )
{
    pt->next=next;
    next = pt;
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

38

Дефиниции на методите

```
Animal :: printlist( Animal *begin )
{
    for(Animal *np=begin; np; np=np->next)
        np->print();
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

39

Дефиниции на методите

```
Boza :: Boza( char *r )
{
    rod = new char[strlen(r)+1];
    strcpy( rod, r );
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

40

Дефиниции на методите

```
Hisht :: Hisht( char *r )
{
    vid = new char[strlen(r)+1];
    strcpy( vid, r );
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

41

Дефиниции на методите

```
void Hisht :: print( void )
{
    Boza :: print();
    cout << " от вид " << vid << "\n";
    getch();
}
```

Виртуални класове. Виртуални функции

05.04.2023г.

42

Дефиниции на методите

```
void main(void ) {
    Animal *p, *pl;
    Animal *a = new Animal;
    Animal *b = new Boza;
    Animal *m = new Mamuna;
    .....
    p = a;
    Animal *list = p;
    pl = b;
    p->link(pl);
    p = pl;
    pl = m;
    p -> link(pl);
    .....
    list->printlist(list);
    delete a;
}
```

Виртуални класове. Виртуални функции

05.04.2023г.