

НАСЛЕДЯВАНЕ НА КЛАСОВЕ

1. Механизъм на наследяване
2. Деклариране на производни класове (просто наследяване)
3. Достъп до наследени компоненти на производни класове
4. Конструктори и деструктори на производни класове
5. Методи с еднакви имена в производни и базови класове
6. Множествено наследяване
7. Достъп до наследени компоненти, които имат еднакви имена

24.3.2020 г.

Наследяване на класове

1

1. Механизъм на наследяване

- Класовете, които наследяват от друг клас, се наричат **производни** класове.
- Класът, от който се наследява, се нарича **базов**.

24.3.2020 г.

Наследяване на класове

2

1. Механизъм на наследяване

Процесът на наследяване по отношение на производните класове се изразява в следното:

- наследява се структурата на данните на базовия клас и неговите методи;
- получава се достъп до наследените членове на базовия клас;
- производният клас познава реализацията само на базовия клас, от който произлиза;
- производен клас може да изпълнява функцията на *базов* за други класове.

24.3.2020 г.

Наследяване на класове

3

1. Механизъм на наследяване

Производният клас може да дефинира допълнително:

- **свои собствени данни;**
- **методи, аналогични на тези на базовия клас, но видоизменени;**
- **нови методи.**

24.3.2020 г.

Наследяване на класове

4

1. Механизъм на наследяване

- ✓ Производният клас не копира конкретни данни от базовия клас.
- ✓ Той наследява само модела на базовия клас, като може и да го доразвие.
- ✓ Обект от производния клас може да използва конкретни данни от обект на базовия клас.

24.3.2020 г.

Наследяване на класове

5

1. Механизъм на наследяване

Предимства при използване на производни класове:

1. Декларирането на производни класове е еквивалентно на конструиране на йерархии от класове;

24.3.2020 г.

Наследяване на класове

6

1. Механизъм на наследяване

Предимства при използване на производни класове:

2. Ако множество от класове имат общи елементи (данни и методи), тези общи части могат да се обособяват като базови класове, а всеки от множеството класове да бъде дефиниран като производен на съответните базови класове;

24.3.2020 г.

Наследяване на класове

7

1. Механизъм на наследяване

Предимства при използване на производни класове:

3. При създаването на производни класове е достатъчно да се разполага само с **обектните модули** на базовите класове, а не с техния програмен текст.

24.3.2020 г.

Наследяване на класове

8

2. Деклариране на производни класове

➤ Под **просто наследяване** се разбира, че производният клас наследява от единствен базов клас.

➤ Декларация на производния клас:

```
class <произв_клас> : <атрибут> <базов_клас>
{
    <тяло на производния клас>
};
```

<атрибут_за_област> - **public** или **private**.

24.3.2020 г.

Наследяване на класове

9

2. Деклариране на производни класове

➤ Производният клас автоматично наследява като дефиниции **всичките членове-данни** на базовия клас.

➤ В тялото на производния клас автоматично се копират моделите на локалната секция **private** и глобалната секция **public** на базовия клас.

24.3.2020 г.

Наследяване на класове

10

2. Деклариране на производни класове

➤ Производният клас **получава възможност да използва всичките нелокални методи на базовия клас.**

➤ Методите могат да се извикват чрез обекти и на базовия, и на производния клас, използва се общ код на функциите.

24.3.2020 г.

Наследяване на класове

11

2. Деклариране на производни класове

```
class Vehicle
{
public:
    void initialize( int, float);
    int getWheels( void );
    float getWeight( void );
    float wheelLoading( void );
    int access( void ) { return k; }

protected:
    int wheels;           // брой колела
    float weight;         // тегло

private:
    int k;
};
```

24.3.2020 г.

Наследяване на класове

12

2. Деклариране на производни класове

```
class Car : public Vehicle
{
public:
    void initialize(int, float, int = 4);
    int passengers( void );
private:
    int passengerLoad; // брой пътници
};
```

24.3.2020 г.

Наследяване на класове

13

2. Деклариране на производни класове

```
class Truck : public Vehicle
{
public:
    void initTruck( int=2, float= 24000.0);
    float efficiency( void );
    int passengers( void );
private:
    int passengerLoad;
    float payLoad;
};
```

24.3.2020 г.

Наследяване на класове

14

3. Достъп до наследени компоненти на производни класове

Базовият клас е деклариран като **public** в декларацията на производния клас

class X : public Y

Декларация на наследен компонент в базовия клас	Достъп на методите на производния клас до наследения компонент	Достъп на външни функции до наследения компонент
public	Да	Да
private	Не	Не
protected	Да	Не

24.3.2020 г.

Наследяване на класове

15

3. Достъп до наследени компоненти на производни класове

Базовият клас е деклариран като **private** в декларацията на производния клас

class X : private Y

Декларация на наследен компонент в базовия клас	Достъп на методите на производния клас до наследения компонент	Достъп на външни функции до наследения компонент
public	Да	Не
private	Не	Не
protected	Да	Не

24.3.2020 г.

Наследяване на класове

16

4. Конструктори и деструктори на производни класове

Как да се инициализира наследената част на производния клас?

- ✓ Функциите по инициализирането са разделени между конструкторите на производния и на базовия клас.
- ✓ Конструкторът на производния клас инициализира само новите собствени членове на този клас.
- ✓ Наследените членове на производния клас се инициализират от конструктор на базовия клас.

24.3.2020 г.

Наследяване на класове

17

4. Конструктори и деструктори на производни класове

Как се активира конструкторът на базовия клас?

Това става в дефиницията на конструктора на производния клас:

```
<произв._клас>::<произв._клас> (<параметри>) : <иниц_списък>
{ <тяло_на_конструктора> }
```

<инициализиращ_списък> е обръщение към конструктор на базовия клас.

24.3.2020 г.

Наследяване на класове

18

4. Конструктори и деструктори на производни класове

Ред на активиране на конструкторите:

- при обръщение към конструктор на производен клас стандартно се заменят формалните параметри с фактическите;
- изпълнява се обръщение към конструктора на базовия клас и изпълнение на операторите в тялото му;
- изпълняват се операторите в тялото на конструктора на производния клас.

24.3.2020 г.

Наследяване на класове

19

4. Конструктори и деструктори на производни класове

- ❖ Ако производният клас има членове-обекти, то техните конструктори се извикват преди конструктора на производния клас.
- ❖ Конструкторите на обектите се извикват по реда на тяхното деклариране в тялото на производния клас.

Деструкторите се извикват по ред, обратен на този на активиране на конструкторите.

24.3.2020 г.

Наследяване на класове

20

4. Конструктори и деструктори на производни класове

При инициализиране на обект от производен клас могат да възникнат следните по-особени случаи:

- ❖ Базовият клас няма конструктор;
- ❖ Базовият клас има само конструктор с параметри;
- ❖ Базовият клас има няколко конструктора и между тях подразбираш се конструктор;
- ❖ Производният клас няма конструктор, а базовият има подразбираш се конструктор;
- ❖ Производният клас няма конструктор. Базовият клас има конструктор с параметри и няма подразбираш се конструктор.

24.3.2020 г.

Наследяване на класове

21

4. Конструктори и деструктори на производни класове

```
class Vehicle
{
    public:
        Vehicle( char *, int );
        void printVehicle( void );
        ~Vehicle( void )
        { delete mark; }
    protected:
        char *mark;
        int year;
};
```

24.3.2020 г.

Наследяване на класове

22

4. Конструктори и деструктори на производни класове

```
class Car : public Vehicle
{
    public:
        Car( char *, int, int );
        void printCar( void );
    private:
        int regNum;
};
```

24.3.2020 г.

Наследяване на класове

23

5. Методи с еднакви имена в производни и базови класове

- Поради локалността на методите в рамките на класовете, използването на еднакви имена на методите в различни класове не води до конфликт.
- За да се извика наследеният метод, трябва да се използва неговото **пълно име**, състоящо се от **името на класа**, на който той принадлежи, **оператора за принадлежност ::** и **името на метода**.

24.3.2020 г.

Наследяване на класове

24

5. Методи с еднакви имена в производни и базови класове

```
class Base
{
    public:
        void getData( void )
        { cout << "X = ";  cin >> x; }
    protected:
        int x;
};
```

24.3.2020 г.

Наследяване на класове

25

5. Методи с еднакви имена в производни и базови класове

```
class Derived : public Base
{
    public:
        void getData( void );
    private:
        int y;
};
```

24.3.2020 г.

Наследяване на класове

26

6. Множествено наследяване

➤ В случаите, когато производният клас наследява няколко базови класа, се казва, че има *множествено наследяване*.

```
➤ class Derived : public Base1, private Base2 ..., BaseN
{    <Декларации на компонентите>    };
```

➤ При множествено наследяване производният клас наследява компонентите на всички базови класове.

24.3.2020 г.

Наследяване на класове

27

6. Множествено наследяване

Последователността на базовите класове в списъка на производния клас се изготвя в следните случаи:

- наследените части на производния клас се разполагат в паметта в съответствие с последователността на базовите класове в декларацията му;
- неявното извикване на конструкторите (деструкторите) на базовите класове се извършва съгласно тази последователност;
- няма значение в какъв ред са записани явно конструкторите на базовите класове в инициализиращия списък на конструктора на производния клас - те ще се активират съгласно последователността на базовите класове в декларацията на производния клас

24.3.2020 г.

Наследяване на класове

28

7. Достъп до наследени компоненти, които имат еднакви имена

➤ Ако в различни базови класове на един произведен клас бъдат дефинирани компоненти с еднакви имена, то в резултат на множественото наследяване всички те ще бъдат наследени от производния клас.

➤ Проблемът, който възниква, е как да бъдат различавани тези компоненти в рамките на производния клас.

➤ Този проблем се решава отново чрез използване на пълните имена на компонентите.

24.3.2020 г.

Наследяване на класове

29

7. Достъп до наследени компоненти, които имат еднакви имена

```
class Base1
{
    public:
        Base1( void )
        { cout << "X= "; cin >> x; }
    protected:
        int x;
};
```

24.3.2020 г.

Наследяване на класове

30

7. Достъп до наследени компоненти, които имат еднакви имена

```
class Base2
{
    public:
        Base2( void )
        { cout << "X= "; cin >> x; }
    protected:
        int x;
};
```

24.3.2020 г.

Наследяване на класове

31

7. Достъп до наследени компоненти, които имат еднакви имена

```
class Derived : public Base1, public Base2
{
    public:
        void storeData( void );
        void printData( void );
    protected:
        int z1, z2;
};
```

24.3.2020 г.

Наследяване на класове

32