

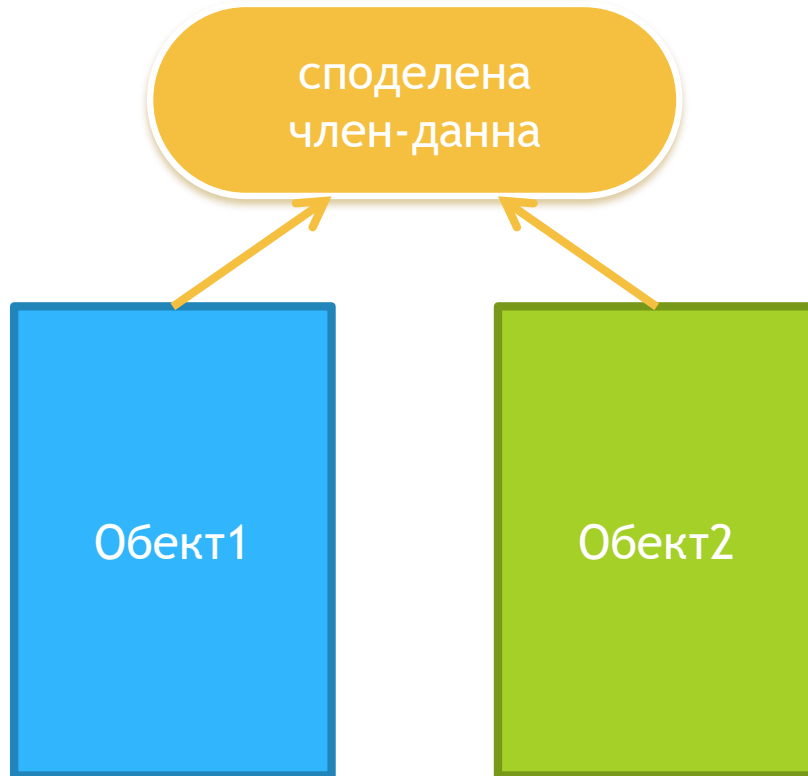
СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ.

гл.ас., д-р. Нора Ангелова

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Статични член-данни на класове

Всички обекти на клас да поделят (използват едновременно) една или повече член-данни на класа.



СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Реализация

За да се създадат поделени от всички обекти на клас член-данни, последните се декларират като статични.

Използва се запазената дума `static`.

```
class A {  
    public:  
    static int counter;  
};
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Дефиниция

Осъществява се **извън** декларацията на класа.

Преди да бъдат използвани, статичните член-данни трябва задължително да бъдат **дефинирани с инициализация**.

```
class A {  
    public:  
    static int counter;  
};
```

```
int A::counter = 0;
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

Особености

- Заделя се **еднократно памет** за този вид член-данни и всички обекти на класа имат достъп до тази памет.

```
class A {  
    public:  
    static int counter;  
};
```

```
int A::counter = 0;
```

```
int main() {  
    A obj1, obj2;
```

```
    cout << &obj1.counter << endl;  
    cout << &obj2.counter << endl;
```

```
    return 0;
```

```
}
```

```
00B37138
```

```
00B37138
```

```
Press any key to continue . . .
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

Особености

- ◎ Памет за статичните член-данни се заделя не в програмния стек, а в **областта за статични данни**.
- ◎ Имат **статут** на **външни променливи**, поради което статичните член-данни могат да се използват и самостоятелно, а не само свързано с обектите на класа.
(*т.е. не са свързани с this, но са част от класа*)
- ◎ Достъп до статичните член-данни на клас може да се осъществява от всяка външна функция, чрез използване на пълното им име, стига статичните член-данни да са дефинирани в **public** секция на класа. При този достъп **не е нужно статичните член-данни да са свързани с обект** на класа.

```
int main() {  
    cout << A::counter << endl;  
  
    return 0;  
}
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

Особености

- Ако се промени статична член-данна от един обект, промяната е валидна за всеки друг обект.

```
int main() {  
    A obj1, obj2;  
  
    cout << obj1.counter << " " << obj2.counter << endl;  
    obj1.counter = 5;  
    cout << obj1.counter << " " << obj2.counter << endl;  
  
    return 0;  
}
```

```
0 0  
5 5  
Press any key to continue . . . _
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Използване

```
class A {  
    public:  
    static int counter;
```

```
    A() {  
        counter++;  
    }
```

```
    ~A() {  
        counter--;  
    }
```

```
};
```

```
int A::counter = 0;
```


СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

● Използване

```
class A {
    private:
        static int counter;
};
```

```
int A::counter = 0;
```

[illegible]

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Област

Статичните член-данни имат за област - тази на класа, в който са дефинирани.

Това е причината достъпът до тях да е чрез оператора `::`.

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Наследяване

Статичните член-данни се наследяват като се запазва статичността им.

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Наследяване

```
class A {  
    public:  
    A() {  
        counter++;  
    }  
    static int counter;  
};  
int A::counter = 0;
```

```
class B : public A {  
    public:  
    B() {  
        counter++; // Извиква се и конструктор на A  
    };  
};
```

```
int main() {  
    A objA1, objA2;  
    cout << objA1.counter << endl << objA2.counter << endl;  
  
    B objB;  
    cout << objA1.counter << endl << objA2.counter << endl << objB.counter << endl;  
    return 0;  
}
```

Резултат:

```
2  
2  
4  
4  
4  
Press any key to continue . . .
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Статични член-данни и константни функции

```
class testClass {  
    public:  
    void updateCounter() const {  
        counter++; // correct (without this)  
        member++;  // incorrect (this->member)  
    }  
  
    private:  
    int member;  
    static int counter;  
};  
  
int testClass::counter = 0;
```

СТАТИЧНИ ЧЛЕН-ФУНКЦИИ НА КЛАСОВЕ

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Статични член-функции на класове

Осъществява се чрез ключовата дума `static`, която се поставя пред типа в `прототипа` на член-функцията.

```
static int staticFunc(int);
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

Особености

- ◎ Няма **неявен параметър**

За тях не е дефиниран указателят `this`.

Затова:

- Статичните член-функции на клас **нямат пряк достъп до нестатичните компоненти на обектите**, чрез които са били извикани, т.е. в тялото на статичен метод могат да се използват само статични компоненти (член-данни и член-функции). Нестатични компоненти могат да се извикват чрез помощен указател, подобен на указателя `this`.

- ◎ Статичните член-функции на класовете могат да бъдат **извиквани**, както **чрез обекти на класа**, така **и самостоятелно**.

СТАТИЧНИ ЧЛЕН-ФУНКЦИИ НА КЛАСОВЕ

- Особености

```
class A {  
    public:  
    int temp;  
  
    static int staticFunc(int tempData) {  
        temp = tempData; // Грешка  
                           // Статичен метод се обръща към  
                           // нестатична член-данна.  
    }  
};
```

СТАТИЧНИ ЧЛЕН-ФУНКЦИИ НА КЛАСОВЕ

Особености

Нестатични компоненти могат да се извикват чрез помощен указател, подобен на указателя `this`.

Пример:

```
class A {  
    public:  
    int temp;  
  
    static int staticFunc(A * obj, int tempData) {  
        obj->temp = tempData; // Правилна дефиниция  
    }  
};  
  
int main() {  
    A obj;  
    cout << A::staticFunc(&obj, 5) << endl;  
    cout << obj.staticFunc(&obj, 5) << endl;  
  
    return 0;  
}
```

СТАТИЧНИ ЧЛЕН-ФУНКЦИИ НА КЛАСОВЕ

● Ограничения

- Статичен метод не може да е виртуален.
- Статичен метод не може да е член-функция за достъп.
- Модификаторът `static` **не е основание** за различаване на две член-функции с еднакви прототипи.

Статичните методи **се наследяват** в производните класове **като статични**.

КРАЙ