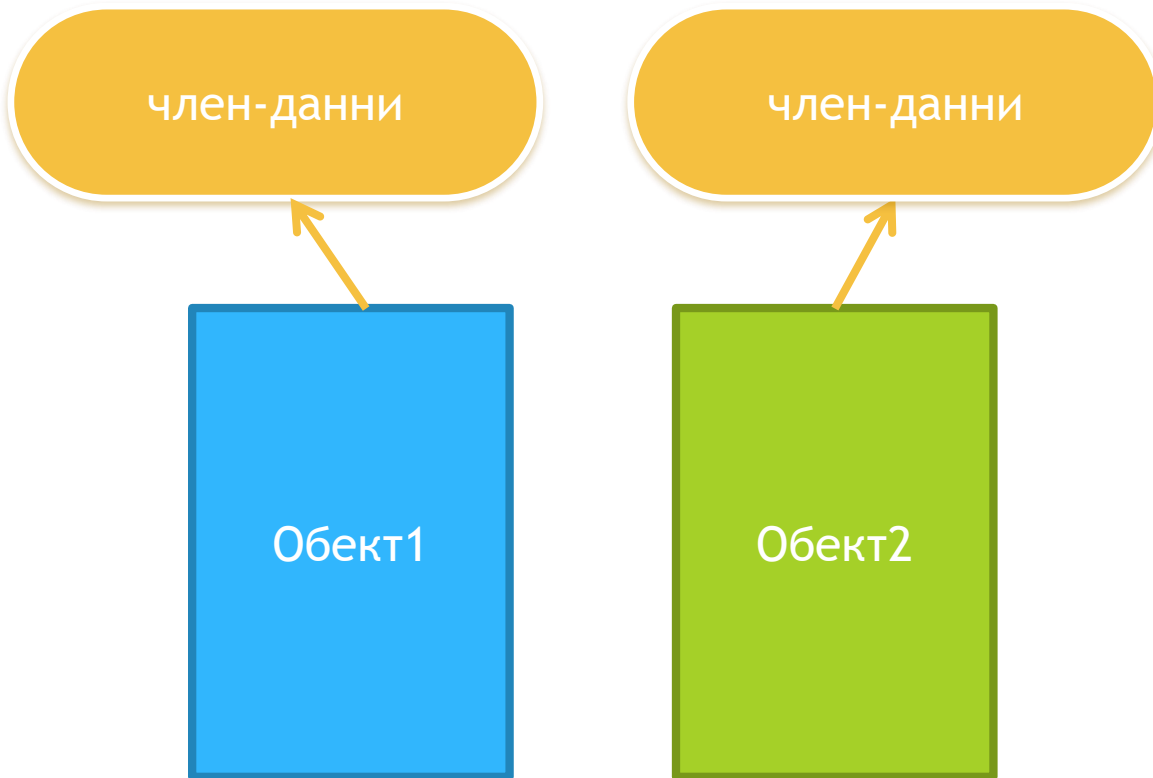


СТАТИЧНИ КОМПОНЕНТИ НА КЛАС

доц., д-р Нора Ангелова

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

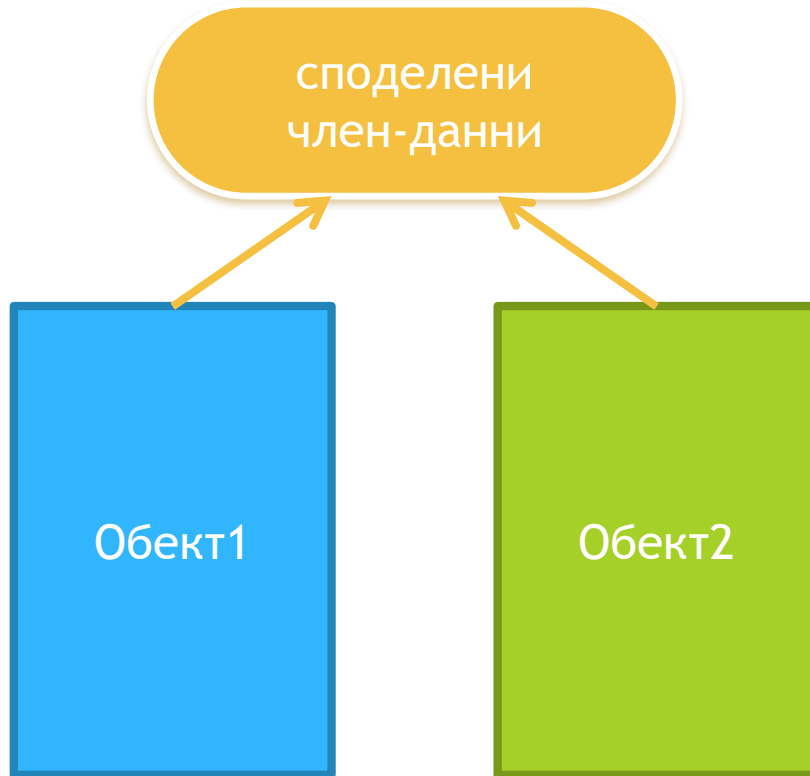
- Обекти



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 Student {  
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 Student {  
public:  
    static int counter;  
};  
  
int Student::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 Student {
public:
    static int counter;
};

int Student::counter = 0;

int main() {
    Student st1, st2;

    std::cout << &st1.counter << std::endl;
    std::cout << &st2.counter << std::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() {  
    std::cout << Student::counter << std::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() {  
    Student st1, st2;  
  
    std::cout << st1.counter << " " << st2.counter << std::endl;  
    st1.counter = 5;  
    std::cout << st1.counter << " " << st2.counter << std::endl;  
  
    return 0;  
}
```

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


СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Използване

```
class Student {  
public:  
    static int counter;  
  
    Student() {  
        counter++;  
    }  
  
    ~Student() {  
        counter--;  
    }  
};  
  
int Student::counter = 0;
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- Използване

```
class Student {
```

```
private:
```

```
    static int counter;
```

```
};
```

```
int Student::counter = 0;    // Грешка по време на компилация  
                             // counter е деклариран в  
                             // private секцията на класа.
```

```
int main() {
```

```
    std::cout << Student::counter << std::endl;
```

```
    return 0;
```

```
}
```

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- ◉ Област - съвпада с областта на класа
- ◉ Достъпът до тях се реализира чрез оператора `::`.

СТАТИЧНИ КОМПОНЕНТИ НА КЛАСОВЕ

- ◎ Статични член-данни и константни функции

```
class Group {  
public:  
    void updateCounter() const {  
        counter++; // correct (without this)  
        members++; // incorrect (this->members)  
    }  
};
```

```
private:  
    int members;  
    static int counter;  
};
```

```
int Group::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

◎ Особенности

```
class Group {  
public:  
    static int staticFunc(int count) {  
        members = count;           // Грешка  
        return count;              // Статичен метод се обръща към  
    }                               // нестатична член-данна.  
private:  
    int members;  
};
```


СТАТИЧНИ ЧЛЕН-ФУНКЦИИ НА КЛАСОВЕ

- Особености

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

Пример:

```
class Group {
public:
    static int staticFunc(Group * obj, int count) {
        obj->members = count;           // Правилна дефиниция
        return count;
    }
private:
    int members;
};

int main() {
    Group obj;
    std::cout << Group::staticFunc(&obj, 5) << std::endl;
    std::cout << obj.staticFunc(&obj, 5) << std::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(гетър).
- Модификаторът `static` **не е основание** за различаване на две член-функции с еднакви прототипи.

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

ВРЕМЕ ЗА ВАШИТЕ
ВЪПРОСИ