

Класове и обекти



Определения

□ **Клас** - структуриран тип данни, който освен краен брой разнотипни полета съдържа и набор от функции за достъп до полетата и тяхната обработка

□ **Обект** - отделен екземпляр от даден клас (отделна променлива от този тип)

```
class Tochka
{
public:
    int x;
    int y;
    Tochka();
    void vhod();
    void pechat() const;
    ~Tochka();
};

Tochka a, b;
```

Елементи на класа

- **Данни** (или член-данни, атрибути) — съдържат данните на обектите от класа
- **Методи** (или член-функции) – съдържат функциите за обработка и достъп до данните. Биват:
 - **конструктори** – служат за създаването на обекти и инициализирането им; извикват се автоматично при създаване на статична или динамична променлива (т.е. обект) от указания клас
 - **деструктори** – за унищожаването на обектите; извикват се автоматично при разрушаването на обекта и освобождаването на неговата памет
 - **функции за достъп** – за достъп до данните на обектите, но без тяхната промяна
 - **мутатори** – за промяна на данните им

Достъп до елементите на класа

- при дефиниране на методите на класа – чрез *клас::метод(...)* за да се покаже, че функцията е свързана с даден клас
- от методите на класа - посредством името на елемента. Достъпът до самия обект е възможен посредством *this->елемент*
- от всички други места – трябва да укажем до елементите на кой точно обект желаем достъп. Указва се чрез *обект.елемент* – ако имаме стат.променлива *указател->елемент* – ако е динамична

Пример за клас

```
class Tochka
{
public:
    int x;
    int y;
    Tochka();
    void vnod();
    void pechat() const;
    ~Tochka();
};
```

```
Tochka a, b;
```

```
Tochka::Tochka() {
    x=y=0;
}

void Tochka::vnod() {
    cout<<"x="; cin>>x;
    cout<<"y="; cin>>y;
}

void Tochka::pechat() const {
    cout<<'('<<x<<', '
        <<y<<')'<<endl;
}
```

Етапи в живота на един обект

- **създаване** - за обекта се заделя памет и данните му се инициализират
Tochka a; // с конструктор по подразбиране
Tochka b(10, 20); // конструктор с параметри
Tochka *pc = new Tochka(a); // конструктор за копиране
- **използване** - осъществява се достъп до елементите на обекта
a.vhod(); // извикване на метод на обекта a
cout<<'('<<b.x<<','<<b.y<<')'; // достъп до данни на b
pc->x = 0; // достъп до данни на pc (указател)
pc->pechat(); // извикване на метод за pc
- **унищожаване** - паметта и ресурсите, заделени за обекта се освобождават
delete pc; // извикване на деструктор за pc

Край

