

Data, metody - práce s nimi (o)

- datové členy – jakýkoli známý typ (objekt nebo ukazatel)
- s datovými členy pracujeme stejně jako s daty uvnitř struktury (z důvodu „bezpečnosti dat“ se snažíme přístupu uživatele k datům zabránit)
- rozlišujeme přístup k datům objektu = operátor “.” K datům „ukazatele“ =operátor “->“
- metody – funkce patřící ke třídě, pracujeme s nimi stejně jako s daty a „funkční volání“ naznačíme přidáním () mezi kterými jsou uvedeny parametry metody
- uživatelsky přístupné metody - interface
- přístupová práva – **private, protected, public**

Nadefinujte třídu, která bude mít privátní členská data (po jednom) typu int, float, class Jmeno, C-řetězec. Dále bude mít veřejné členské metody a jeden datový člen int:

metoda1 – bez parametrů, vrátí int

metoda2 – dva parametry int a float, bez návratové hodnoty

metoda 3 – vrátí float a má parametry int a ukazatel na Jmeno

Napište část programu, který nadefinuje objekt dané třídy, ukazatel inicializovaný tímto objektem. Dále ukažte přístup ke členům a metodám a řekněte, které budou fungovat.

```
// deklarace (popis) třídy - v souboru jmeno_tridy.h
class Jmeno_třída { // implicitně private:
    int data1; //datové členy třídy
    float data2;
    Jmeno *j; // class není nutné uvádět
    char string[100];
public: // metody třídy
    int metoda1() {...return 2;}
    void metoda2(int a,float b) {...}
    float metoda3( int  a1,  Jmeno *a2);
    int dd;//nevhodné, veřejně přístupná proměnná
};
```

```
Jmeno_třída aa, *bb = &aa;
// obdoba struct pom aa, *bb = &aa;
// přístup jako k datovým prvkům struct
aa.dd = bb->dd;
int b = aa.metoda3(34,"34.54"), c = bb->metoda3(34,"34.54");
aa.metoda1(); bb->metoda1();
// aa.data1 = bb->data1; // nelze přistupovat k privátním datům
```