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 – členské funkce patřící ke třídě, pracujeme s nimi stejně jako s daty a „funkční volání“ naznačíme přidáním „()“, mezi kterými jsou uvedeny parametry metody
- uživatelovi přístupné metody tvoří 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, vrací int
mteoda2 – dva parametry int a float, bez návratové hodnoty
metoda 3 – vrací 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_třídy { // 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_třídy 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(); // přístup přes proměnnou/objekt
bb->metoda1(); // přístup přes ukazatel na proměnnou/objekt
// aa.data1 = bb->data1; // nelze přistupovat k privátním datům
```

```
struct Komplex { // public: - implicitní
double Re, Im; // public

private:  // přepínač přístupových práv

double Velikost(void) {return 14;}
// metoda (funkce) interní
int pom;  // interní-privátní proměnná

public: // přepínač přístupových práv

// metoda veřejná = interface
double Uhel(double a ) {return a-2;}
};

Komplex a,b, *pc = &b;
a.Re = 1;      pc->Re = 3;      // je možné
b.Uhel(3.14); pc->Uhel(0);     // je možné
a.pom = 3;     pc->pom = 5  ;  // není možné
b.Velikost(); pc->Velikost(); // není možné
```

Data, metody – postup při psaní programu

- datovou reprezentaci a metody si dobře promyslíme a rozvrhneme
- pokud není speciální důvod, potom data jsou v sekci private
- metody „nebezpečné“ nebo s omezenými kontrolami jsou private (aby je nemohl volat uživatel, který by mohl použít tyto metody, aniž by domyslel důsledky)
- metody tvořící interface (přístupné uživatelovi) – uvedeme v sekci public
- z hlediska programu obvykle jako první píšeme „speciální“ metody pro vznik a zánik proměnné – konstruktory a destruktor

```
class CComplex { // implicitně private:
   double iReal, iImag; //datové členy třídy
   double iApha, iAmplitude;
// použít všechny čtyři členy? nebo jen dva? které?

public: // metody třídy
  void Set(double aRe, double aIm) {…}
  double GetReal() {return iReal;}

  int error;//nevhodné, veřejně přístupná proměnná
 };
```