

# 9. Cvičení

Objekty

# Abstraktní třídy

- Chceme modelovat výpočty důchodu:
  - evidujeme informace o jménu a datu narození,
  - chceme mít k dispozici rozhraní, které rozhodne, zda dané osobě již vznikl nárok na důchod.
- Nárok na důchod vzniká:
  - pro muže ve věku 65 let, od tohoto věku se odečítá délka vojenské služby,
  - pro ženy ve věku 63 let, od tohoto věku se odečítají 2 roky za každé vychované dítě (neodpovídá realitě, jen příklad).

# Abstraktní třídy

- Řešení s abstraktními třídami.
- Třída **CPerson**:
  - společné vlastnosti (jméno, rok narození),
  - deklaruje rozhraní (metoda pro výpočet, zda již vznikl nárok na důchod),
  - nedefinuje tuto metodu – abstraktní metoda.
- Třída **CMan**:
  - dědí ze třídy **CPerson**,
  - navíc informace o vojenské službě,
  - definuje metodu výpočtu nároku na důchod.
- Třída **CWoman**:
  - dtto, jen ukládá počet vychovaných dětí.

# Abstraktní třídy

```
class CPerson
{
    protected:
        string name;
        int  born;
    public:
        CPerson ( string _name, int _born ) :
            name (_name), born(_born) { }

        virtual ~CPerson ( void ) { }
        virtual int retired ( int year ) const = 0;
};
```

# Abstraktní třídy

```
class CWoman : public CPerson
{
protected:
    int childs;
public:
    CWoman ( string _name, int _born, int _childs ):
        CPerson ( _name, _born ), childs(_childs) { }
    virtual int retired ( int year ) const
        { return year > born + 63 - 2 * childs; }
};
```

# Abstraktní třídy

```
class CMan : public CPerson
{
protected:
    int milSvc;
public:
    CMan ( string _name, int _born, int _milSvc ) :
        CPerson ( _name, _born ), milSvc(_milSvc) { }
    virtual int retired ( int year ) const
        { return year > born + 65 - milSvc; }
};
```

# Abstraktní třídy

```
CPerson * people [2];  
people[0] = new CMan ( "Novak", 1948, 2 );  
people[1] = new CWoman ( "Novakova", 1948, 3 );
```

```
for ( i = 0; i < 2; i ++ )  
    cout << i << ". "  
        << (people[i]->retired ( 2005 ) ? "ano" : "ne")  
        << endl;
```

```
for ( i = 0; i < 2; i ++ )  
    delete people[i];
```

# Abstraktní třídy

- Abstraktní třída deklaruje metodu:
  - je dáno rozhraní metody (jméno, parametry, ...),
  - není definované tělo metody,
  - v deklaraci označena **=0**,
- existuje v předkovi, aby se vyhradil prostor v tabulce virtuálních metod (VMT).
- Těla metod definují potomci.
- Nelze vytvořit instanci abstraktní třídy.



# Abstraktní třídy

- Abstraktní předek:
  - jednotný pohled na více heterogenních objektů,
  - využití rozhraní vyšší úrovně, netřeba rozlišovat detaily implementace podtříd,
- uplatnění zejména kolekcích.

# Úkol 1

- Implementujte následující diagram tříd.
- Jednotky jsou ze hry Dune 2.
- U každé jednotky zjistěte zda je pásová, poháněná koly nebo pěší.
- Implementujte UnitFactory tak, aby byla schopna vyrábět jednotky.
- Každá jednotka bude zdravit svého kapitána například: “I am infantry. My mobility is foot.”

