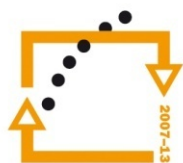




MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání
pro konkurenceschopnost**

INVESTICE
DO ROZVOJE
VZDĚLÁVÁNÍ

Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1

Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT

Název: Základy programování a algoritmizace úloh

Téma: Objekt

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_26–12

Anotace: *Materiál popisuje definici a použití datového typu objekt a stručný úvod do objektově orientovaného programování v jazyce Pascal.
Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: březen 2013.*

12. Objekt

- datový typ **objekt** je nejsložitějším, ale také nejsilnějším datovým typem v Pascalu
- na použití tohoto datového typu je v mnoha programovacích jazycích založen celý speciální princip programování zvaný **objektově orientované programování** nebo také objektové programování, které v současnosti představuje vrchol programátorských technik
- objekt je strukturovaný datový typ – v jedné proměnné můžeme mít **více informací**, u kterých **nezáleží na pořadí**
- objekty patří mezi uživatelsky definovatelné typy – programátor si je může vytvořit dle svých potřeb

Z čeho jsou složeny objekty

- zvláštností objektů je, že kromě úložiště pro data, jak to známe z běžných proměnných, slouží také jako nositelé programového kódu, který je obvykle určen pro správu a obsluhu dat objektu
- datům v objektu se obvykle říká **vlastnosti**
- programový kód, který se stará o vlastnosti objektu má podobu **procedur a funkcí** (viz 26-14 – Podprogramy) a ty se v této souvislosti souhrnně nazývají **metody**
- kdyby neobsahoval žádné metody ale jen vlastnosti, byl by objekt velmi podobný **záznamu**
- právě rozšířením datového typu záznam (viz 26-10 – Záznam) objekty vznikly

Vlastnosti

- vlastnosti se podobně jako u záznamu mohou nazývat **libovolnými jmény** a mohou být **libovolného typu** (tedy i typu objekt; tzn. že v objektu mohou být obsaženy objekty)
- ukázka deklarace typu objekt (pouze vlastnosti)

type

```
TDatumACas = object
```

```
den : byte;
```

```
mesic : 1..12;
```

```
rok : integer;
```

```
hodina : 0..23;
```

```
minuta, vterina : 0..59;
```

```
...
```

```
end;
```

Metody – definice

- metody jsou procedury a funkce primárně určené pro obsluhu vlastností objektu
- v definici datového typu objekt se uvádí pouze jejich deklarační část (typ, název, parametry)

type

```
TDatumACas = object
```

```
...
```

```
function Nastav(d, m, r, h, mi,  
s:integer):boolean;
```

```
procedure VypisDatum(format:string);
```

```
procedure VypisCas(sec:boolean);
```

```
end;
```

Metody – implementace

- všechny metody uvedené v definici datového typu objekt musí být zapsány v hlavičce programu spolu s běžnými procedurami a funkcemi
- před název procedury se zde přidává tečka a název celého datového typu jehož metodou je

```
procedure TDatumACas.VypisCas (sec:boolean) ;  
begin  
    if sec then  
        writeln (hodina, ' : ' ,minuta, ' : ' ,sekunda)  
    else  
        writeln (hodina, ' : ' ,minuta) ;  
end;
```

Vlastnosti objektů

- kvalita a síla objektového programování vychází ze tří základních vlastností objektů

dědičnost – můžeme vytvářet nové objekty odvozením z již hotových; programujeme pak jen změny, ne celý objekt

zapouzdření – součástí objektu nejsou jen data (vlastnosti), ale také nástroje pro práci s nimi (metody)

polymorfismus – objekty mohou volat stejně pojmenovanou metodu, která je však pro každý z nich jinak naprogramována