

Objektově orientované programování

Cvičení č. 3

Ing. Lukáš Slánský

Ústav elektrotechniky a informatiky, Univerzita Pardubice

17. 10. 2006



Obsah

- 1 Co si vzít z přednášek?
- 2 Samostatná práce
 - Řešení samostatné práce



Co si vzít z přednášek?

- Dědičnost



Co si vzít z přednášek?

- Dědičnost
 - ▶ přebírání atributů a metod od předka



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem integer:



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem integer:

- Umí zpracovat zprávy *push*, *pop*, *full* a *empty*.



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem `integer`:

- Umí zpracovat zprávy *push*, *pop*, *full* a *empty*.
- Všechny ostatní metody (existují-li) a atributy nejsou použitelné.



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem integer:

- Umí zpracovat zprávy *push*, *pop*, *full* a *empty*.
- Všechny ostatní metody (existují-li) a atributy nejsou použitelné.

Časový plán:



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem integer:

- Umí zpracovat zprávy *push*, *pop*, *full* a *empty*.
- Všechny ostatní metody (existují-li) a atributy nejsou použitelné.

Časový plán:

- 30 minut – samostatná práce
- Společná analýza problému



Samostatná práce

Napište (a lehce otestujte) třídu, která implementuje zásobník (LIFO) nad typem integer:

- Umí zpracovat zprávy *push*, *pop*, *full* a *empty*.
- Všechny ostatní metody (existují-li) a atributy nejsou použitelné.

Časový plán:

- 30 minut – samostatná práce
- Společná analýza problému
- 30 minut na dokončení samostatné práce
- Společný rozbor implementace



Řešení samostatné práce – analýza

- Atributy:



Řešení samostatné práce – analýza

- Atributy:
 - ▶ seznam uložených položek – `Items`,



Řešení samostatné práce – analýza

- Atributy:

- ▶ seznam uložených položek – `Items`,
- ▶ aktuální vrchol zásobníku – `StackTop`,



Řešení samostatné práce – analýza

- Atributy:

- ▶ seznam uložených položek – `Items`,
- ▶ aktuální vrchol zásobníku – `StackTop`,
- ▶ maximální počet uložitelných položek – `MaxItemCount`.



Řešení samostatné práce – analýza

- Atributy:

- ▶ seznam uložených položek – `Items`,
- ▶ aktuální vrchol zásobníku – `StackTop`,
- ▶ maximální počet uložitelných položek – `MaxItemsCount`.

- Metody:



Řešení samostatné práce – analýza

- Atributy:

- ▶ seznam uložených položek – `Items`,
- ▶ aktuální vrchol zásobníku – `StackTop`,
- ▶ maximální počet uložitelných položek – `MaxItemsCount`.

- Metody:

- ▶ již známé metody připojené ke zprávám (`push`, `pop`, `full` a `empty`),



Řešení samostatné práce – analýza

- Atributy:

- ▶ seznam uložených položek – `Items`,
- ▶ aktuální vrchol zásobníku – `StackTop`,
- ▶ maximální počet uložitelných položek – `MaxItemsCount`.

- Metody:

- ▶ již známé metody připojené ke zprávám (`push`, `pop`, `full` a `empty`),
- ▶ prvotní nastavení objektu – `init`.



Řešení samostatné práce – analýza II.



Řešení samostatné práce – analýza II.

Zpracování chyb:



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ `push` (vlození do plného zásobníku),



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).
- Metody pro ošetřování chyb:



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).
- Metody pro ošetřování chyb:
 - ▶ nastala nějaká chyba při poslední operaci? – `isError`,



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).
- Metody pro ošetřování chyb:
 - ▶ nastala nějaká chyba při poslední operaci? – `isError`,
 - ▶ jaká chyba nastala při poslední operaci? – `getError`.



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).
- Metody pro ošetřování chyb:
 - ▶ nastala nějaká chyba při poslední operaci? – `isError`,
 - ▶ jaká chyba nastala při poslední operaci? – `getError`.
- Atributy pro ošetřování chyb:



Řešení samostatné práce – analýza II.

Zpracování chyb:

- Které metody generují (mohou generovat) chyby?
 - ▶ push (vlození do plného zásobníku),
 - ▶ pop (vybrání z prázdného zásobníku).
- Metody pro ošetřování chyb:
 - ▶ nastala nějaká chyba při poslední operaci? – `isError`,
 - ▶ jaká chyba nastala při poslední operaci? – `getError`.
- Atributy pro ošetřování chyb:
 - ▶ vnitřní příznak, jaká chyba nastala – `LastError`.



Deklarace třídy

```
type CLifo=object
```

```
end;
```

Deklarace třídy

```
type CLifo=object  
    public
```

```
    private
```

```
end;
```

Deklarace třídy

```
type CLifo=object
  public

  private
    Items:array[1..10] of integer;
    StackTop:integer;
    MaxItemCount:integer;
    LastError:integer;
end;
```



Deklarace třídy

```
type CLifo=object
  public
    procedure init;
    procedure push(i:integer);
    function pop:integer;
    function empty:boolean;
    function full:boolean;
    function isError:boolean;
    function getError:string;
  private
    Items:array[1..10] of integer;
    StackTop:integer;
    MaxItemsCount:integer;
    LastError:integer;
end;
```



Definice metod – I.

```
procedure CLifo.init;
```



Definice metod – I.

```
procedure CLifo.init;  
begin  
    LastError:=0;  
    StackTop:=0;  
    MaxItemsCount:=10;  
end;
```



Definice metod – I.

```
procedure CLifo.init;  
begin  
    LastError:=0;  
    StackTop:=0;  
    MaxItemsCount:=10;  
end;  
  
function CLifo.empty:boolean;
```



Definice metod – I.

```
procedure CLifo.init;  
begin  
    LastError:=0;  
    StackTop:=0;  
    MaxItemsCount:=10;  
end;
```

```
function CLifo.empty:boolean;  
begin  
    empty:=(StackTop=0);  
end;
```



Definice metod – I.

```
procedure CLifo.init;  
begin  
    LastError:=0;  
    StackTop:=0;  
    MaxItemsCount:=10;  
end;  
  
function CLifo.empty:boolean;  
begin  
    empty:=(StackTop=0);  
end;  
  
function CLifo.full:boolean;
```



Definice metod – I.

```
procedure CLifo.init;  
begin  
    LastError:=0;  
    StackTop:=0;  
    MaxItemsCount:=10;  
end;  
  
function CLifo.empty:boolean;  
begin  
    empty:=(StackTop=0);  
end;  
  
function CLifo.full:boolean;  
begin  
    full:=(StackTop=MaxItemsCount);  
end;
```



Definice metod – II.

```
procedure CLifo.push(i:integer);  
begin
```

```
end;
```



Definice metod – II.

```
procedure CLifo.push(i:integer);  
begin
```

```
    StackTop:=StackTop+1;  
    Items[StackTop]:=i;
```

```
end;
```



Definice metod – II.

```
procedure CLifo.push(i:integer);
begin
  if full then
    LastError:=1
  else
    begin
      LastError:=0;
      StackTop:=StackTop+1;
      Items[StackTop]:=i;
    end;
end;
```



Definice metod – III.

```
function CLifo.pop:integer;
```



Definice metod – III.

```
function CLifo.pop:integer;  
begin  
    if empty then  
        LastError:=2  
    else  
        begin  
            LastError:=0;
```



Definice metod – III.

```
function CLifo.pop:integer;  
begin  
    if empty then  
        LastError:=2  
    else  
        begin  
            LastError:=0;  
            pop:=Items[StackTop];  
            StackTop:=StackTop-1;  
        end;  
    end;  
end;
```



Definice metod – IV.

```
function CLifo.isError:boolean;
```



Definice metod – IV.

```
function CLifo.isError:boolean;  
begin  
    isError:=(LastError<>0);  
end;
```



Definice metod – IV.

```
function CLifo.isError:boolean;  
begin  
    isError:=(LastError<>0);  
end;  
  
function CLifo.getError:string;
```



Definice metod – IV.

```
function CLifo.isError:boolean;  
begin  
    isError:=(LastError<>0);  
end;
```

```
function CLifo.getError:string;  
begin  
    case LastError of  
        0: getError:='';  
        1: getError:='Stack is full';  
        2: getError:='Stack is empty';  
        else getError:='Unknown error';  
    end;  
end;
```

