Abstraktní datové typy

Programovací techniky

doc. Ing. Jiří Rybička, Dr.
ústav informatiky
PEF MENDELU v Brně
rybicka@mendelu.cz

Pojem abstraktního typu

Pojem abstraktního typu

- Datový typ = hodnoty, operace

Pojem abstraktního typu

- Datový typ = hodnoty, operace
- Konkrétní datové typy jsou implementovány v daném programovacím jazyce

Pojem abstraktního typu

- Datový typ = hodnoty, operace
- Konkrétní datové typy jsou implementovány v daném programovacím jazyce
- Model objektivní reality obvykle není konkrétním typem přímo implementovatelný

Pojem abstraktního typu

- Datový typ = hodnoty, operace
- Konkrétní datové typy jsou implementovány v daném programovacím jazyce
- Model objektivní reality obvykle není konkrétním typem přímo implementovatelný
- Abstraktní datový typ – stanovení hodnot a operací modelujících potřebnou skutečnost

Pojem abstraktního typu

- Datový typ = hodnoty, operace
- Konkrétní datové typy jsou implementovány v daném programovacím jazyce
- Model objektivní reality obvykle není konkrétním typem přímo implementovatelný
- Abstraktní datový typ – stanovení hodnot a operací modelujících potřebnou skutečnost
- N. Wirth: Algoritmy + Datové struktury = Programy

Pojem abstraktního typu

- Datový typ = hodnoty, operace
- Konkrétní datové typy jsou implementovány v daném programovacím jazyce
- Model objektivní reality obvykle není konkrétním typem přímo implementovatelný
- Abstraktní datový typ – stanovení hodnot a operací modelujících potřebnou skutečnost
- N. Wirth: Algoritmy + Datové struktury = Programy
- prioritní jsou datové struktury, jsou modelem reality

Popis abstraktního typu

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)
- Popis syntaxe operací – diagram signatury

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)
- Popis syntaxe operací – diagram signatury
- Popis sémantiky operací – vyjádření algoritmu

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)
- Popis syntaxe operací – diagram signatury
- Popis sémantiky operací – vyjádření algoritmu
- Ve všech částech popisu je potřebné co nejpřesněji vyjádřit modelované vlastnosti

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)
- Popis syntaxe operací – diagram signatury
- Popis sémantiky operací – vyjádření algoritmu
- Ve všech částech popisu je potřebné co nejpřesněji vyjádřit modelované vlastnosti
- Každý prvek popisu by měl mít jednoznačný způsob převodu do zvoleného implementačního jazyka

Popis abstraktního typu

- Popis hodnot (obecně, bez jakékoliv vazby na implementaci)
- Popis syntaxe operací – diagram signatury
- Popis sémantiky operací – vyjádření algoritmu
- Ve všech částech popisu je potřebné co nejpřesněji vyjádřit modelované vlastnosti
- Každý prvek popisu by měl mít jednoznačný způsob převodu do zvoleného implementačního jazyka
- Převod do programovacího jazyka může mít více variant podle zvolených jazykových nástrojů (neobjektová/objektová implementace, moduly, rekurze/iterace apod.)

Popis hodnot

Popis hodnot

- Obvykle jde o neskalární hodnoty

Popis hodnot

- Obvykle jde o neskalární hodnoty
- Lze zvolit grafické prostředky, schémata, matematický formalismus

Popis hodnot

- Obvykle jde o neskalární hodnoty
- Lze zvolit grafické prostředky, schémata, matematický formalismus
- Příklad: Abstraktní typ zásobník

Popis hodnot

- Obvykle jde o neskalární hodnoty
- Lze zvolit grafické prostředky, schémata, matematický formalismus
- Příklad: Abstraktní typ zásobník
- Hodnotou je posloupnost prvků, z nichž jeden je význačný = vrchol zásobníku

Popis hodnot

- Obvykle jde o neskalární hodnoty
- Lze zvolit grafické prostředky, schémata, matematický formalismus
- Příklad: Abstraktní typ zásobník
- Hodnotou je posloupnost prvků, z nichž jeden je význačný = vrchol zásobníku
- Graficky lze vyjádřit jako sloupec pod sebe zapsaných prvků, prvek zcela nahoře je vrchol zásobníku

Popis hodnot

- Obvykle jde o neskalární hodnoty
- Lze zvolit grafické prostředky, schémata, matematický formalismus
- Příklad: Abstraktní typ zásobník
- Hodnotou je posloupnost prvků, z nichž jeden je význačný = vrchol zásobníku
- Graficky lze vyjádřit jako sloupec pod sebe zapsaných prvků, prvek zcela nahoře je vrchol zásobníku
- Matematicky lze vyjádřit jako posloupnost označenou různými typy závorek, například:

$$\langle a_1, a_2, \ldots, a_n]$$

kde závorka $\langle$ představuje vrchol zásobníku, závorka] pak dno zásobníku.

Popis syntaxe operací

Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury

Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury
- Jde o orientovaný graf

Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury
- Jde o orientovaný graf
- Uzly jsou dvojího druhu:
  1. Datové typy – znázorněny oválky, hlavní datový typ je zvýrazněn (tučná čára, dvojitá čára)
  2. Operace – plná kolečka

Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury
- Jde o orientovaný graf
- Uzly jsou dvojího druhu:
  1. Datové typy – znázorněny oválky, hlavní datový typ je zvýrazněn (tučná čára, dvojitá čára)
  2. Operace – plná kolečka
- Uzly jsou propojeny orientovanými hranami. Hrana spojuje vždy uzel typu s uzlem operace

Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury
- Jde o orientovaný graf
- Uzly jsou dvojího druhu:
  1. Datové typy – znázorněny oválky, hlavní datový typ je zvýrazněn (tučná čára, dvojitá čára)
  2. Operace – plná kolečka
- Uzly jsou propojeny orientovanými hranami. Hrana spojuje vždy uzel typu s uzlem operace
- Hrana vedoucí směrem do uzlu operace = vstupní parametr

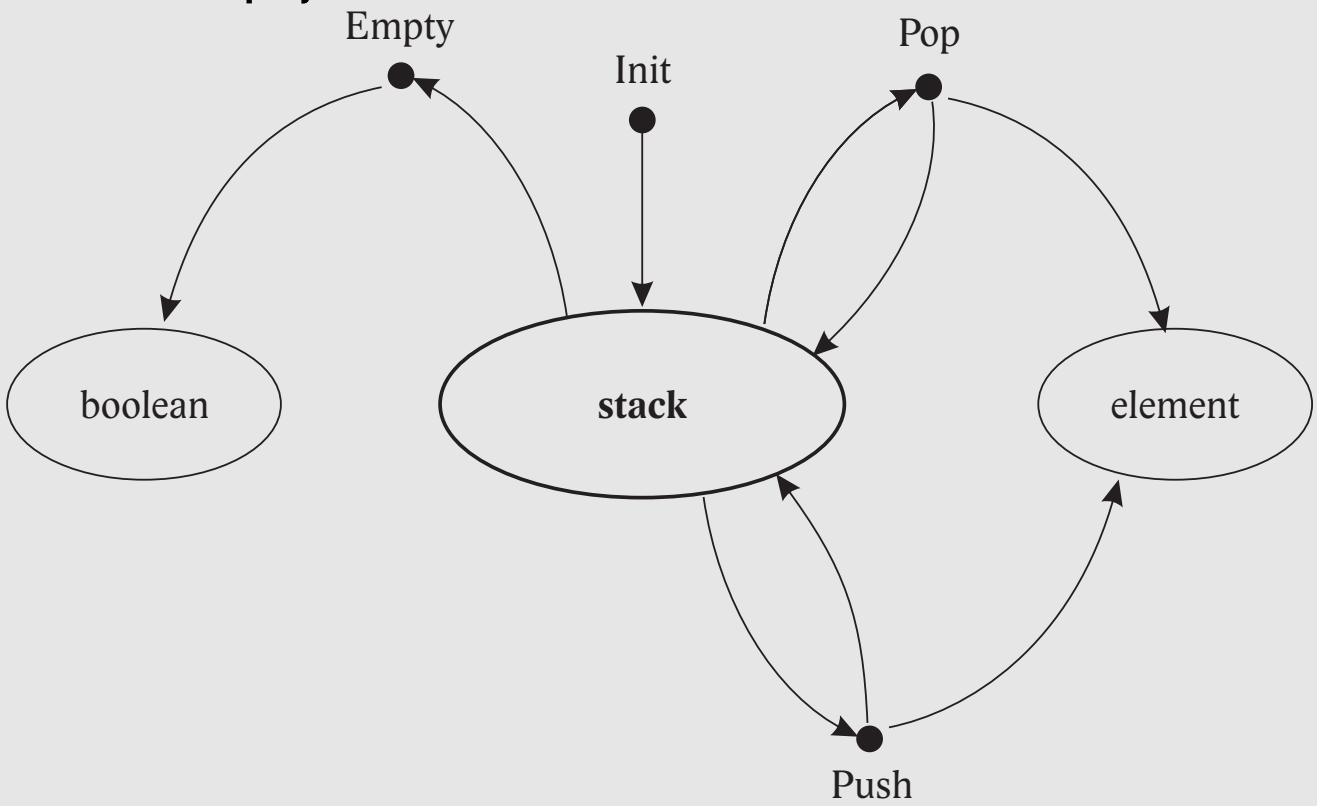
Popis syntaxe operací

- Pro vyjádření syntaxe operací se s výhodou používá tzv. diagram signatury
- Jde o orientovaný graf
- Uzly jsou dvojího druhu:
  1. Datové typy – znázorněny oválky, hlavní datový typ je zvýrazněn (tučná čára, dvojitá čára)
  2. Operace – plná kolečka
- Uzly jsou propojeny orientovanými hranami. Hrana spojuje vždy uzel typu s uzlem operace
- Hrana vedoucí směrem do uzlu operace = vstupní parametr
- Hrana vedoucí směrem ven z uzlu operace = výstupní parametr

Příklad diagramu signatury

Příklad diagramu signatury

- Abstraktní datový typ zásobník s operacemi Init, Pop, Push, Empty



Operace v diagramu signatury

Operace v diagramu signatury

- Význačné operace: konstruktor, predikáty

Operace v diagramu signatury

- Význačné operace: konstruktor, predikáty
- Konstruktor (Init) – vytváří počáteční hodnotu hlavního typu

Operace v diagramu signatury

- Význačné operace: konstruktor, predikáty
- Konstruktor (Init) – vytváří počáteční hodnotu hlavního typu
- V diagramu signatury jde o operaci s výstupem hlavního typu, nemusí mít žádné vstupy

Operace v diagramu signatury

- Význačné operace: konstruktor, predikáty
- Konstruktor (Init) – vytváří počáteční hodnotu hlavního typu
- V diagramu signatury jde o operaci s výstupem hlavního typu, nemusí mít žádné vstupy
- Predikát – logická funkce, obvykle udává stav hlavního typu (Empty)

Operace v diagramu signatury

- Význačné operace: konstruktor, predikáty
- Konstruktor (Init) – vytváří počáteční hodnotu hlavního typu
- V diagramu signatury jde o operaci s výstupem hlavního typu, nemusí mít žádné vstupy
- Predikát – logická funkce, obvykle udává stav hlavního typu (Empty)
- V diagramu signatury je to operace se vstupem hlavního typu a s výstupem logického typu

Sémantika operací

Sémantika operací

- Vyjádření algoritmu, který je danou operací realizován

Sémantika operací

- Vyjádření algoritmu, který je danou operací realizován
- Většinou jde o přesný způsob zpracování vstupních hodnot na výstupní

Sémantika operací

- Vyjádření algoritmu, který je danou operací realizován
- Většinou jde o přesný způsob zpracování vstupních hodnot na výstupní
- Jednou z možností vyjádření je popis možných vstupů a jim odpovídajících výstupů

Sémantika operací

- Vyjádření algoritmu, který je danou operací realizován
- Většinou jde o přesný způsob zpracování vstupních hodnot na výstupní
- Jednou z možností vyjádření je popis možných vstupů a jim odpovídajících výstupů
- Hodnoty mohou být vyjádřeny stejným způsobem jako u popisu hodnot typu

Příklad sémantiky operací zásobníku

Příklad sémantiky operací zásobníku

- Operace Init:

$$\Rightarrow \langle]$$

Příklad sémantiky operací zásobníku

- Operace Init:

$$\Rightarrow \langle]$$

- Operace Push:

$$x, \langle a_1, \ldots, a_n] \Rightarrow \langle x, a_1, \ldots, a_n]$$

Příklad sémantiky operací zásobníku

- Operace Init:

$$\Rightarrow \langle]$$

- Operace Push:

$$x, \langle a_1, \ldots, a_n] \Rightarrow \langle x, a_1, \ldots, a_n]$$

- Operace Pop:

$$\begin{aligned} \langle a_1, a_2, \ldots, a_n] &\Rightarrow a_1, \langle a_2, \ldots, a_n] \\ \langle] &\Rightarrow \text{error} \end{aligned}$$

Příklad sémantiky operací zásobníku

- Operace Init:

$$\Rightarrow \langle]$$

- Operace Push:

$$x, \langle a_1, \ldots, a_n] \Rightarrow \langle x, a_1, \ldots, a_n]$$

- Operace Pop:

$$\begin{aligned} \langle a_1, a_2, \ldots, a_n] &\Rightarrow a_1, \langle a_2, \ldots, a_n] \\ \langle] &\Rightarrow \text{error} \end{aligned}$$

- Operace Empty:

$$\begin{aligned} \langle a_1, \ldots, a_n] &\Rightarrow \text{false} \\ \langle] &\Rightarrow \text{true} \end{aligned}$$

Implementace abstraktního typu

Implementace abstraktního typu

- Volba implementačního jazyka

Implementace abstraktního typu

- Volba implementačního jazyka
- Implementace hodnot – volba vhodné metody reprezentace pomocí konkrétních typů daného jazyka

Implementace abstraktního typu

- Volba implementačního jazyka
- Implementace hodnot – volba vhodné metody reprezentace pomocí konkrétních typů daného jazyka
- Možnosti: strukturované datové typy, dynamické datové struktury, objekty

Implementace abstraktního typu

- Volba implementačního jazyka
- Implementace hodnot – volba vhodné metody reprezentace pomocí konkrétních typů daného jazyka
- Možnosti: strukturované datové typy, dynamické datové struktury, objekty
- U operací se vychází z diagramu signatury

Implementace abstraktního typu

- Volba implementačního jazyka
- Implementace hodnot – volba vhodné metody reprezentace pomocí konkrétních typů daného jazyka
- Možnosti: strukturované datové typy, dynamické datové struktury, objekty
- U operací se vychází z diagramu signatury
- Většinou realizováno podprogramy, návrh systému parametrů, logická struktura

Implementace abstraktního typu

- Volba implementačního jazyka
- Implementace hodnot – volba vhodné metody reprezentace pomocí konkrétních typů daného jazyka
- Možnosti: strukturované datové typy, dynamické datové struktury, objekty
- U operací se vychází z diagramu signatury
- Většinou realizováno podprogramy, návrh systému parametrů, logická struktura
- Volba implementačních nástrojů seskupujících podprogramy (moduly, objekty)

Implementace zásobníku

Implementace zásobníku

- Hodnoty implementovány jako dynamický lineární seznam

Implementace zásobníku

- Hodnoty implementovány jako dynamický lineární seznam
- Datové složky jsou určitého typu reprezentovaného identifikátorem `Element`; pro příklad použití zvolíme **`string`**

Implementace zásobníku

- Hodnoty implementovány jako dynamický lineární seznam
- Datové složky jsou určitého typu reprezentovaného identifikátorem `Element`; pro příklad použití zvolíme `string`
- Operace implementovány jako procedury a funkce

Implementace zásobníku

- Hodnoty implementovány jako dynamický lineární seznam
- Datové složky jsou určitého typu reprezentovaného identifikátorem `Element`; pro příklad použití zvolíme `string`
- Operace implementovány jako procedury a funkce
- Operace Pop nad prázdným zásobníkem vyvolá chybu, která je řešena procedurou `Error`

Implementace zásobníku

- Hodnoty implementovány jako dynamický lineární seznam
- Datové složky jsou určitého typu reprezentovaného identifikátorem `Element`; pro příklad použití zvolíme `string`
- Operace implementovány jako procedury a funkce
- Operace Pop nad prázdným zásobníkem vyvolá chybu, která je řešena procedurou `Error`
- Příklad použití: Na vstupu je řada řetězců, program má tuto řadu vypsat v obráceném pořadí

Implementace zásobníku

```
type
     Element = string;
     {příklad datové složky}
     PClen = ^Clen;
     Clen = record
       data: Element;
       dalsi: PClen
     end;
     Stack = PClen;
     {reprezentace hlavního typu}

procedure Init(var S: Stack);
 begin S:=nil
 end;
```

Implementace zásobníku

```
function Empty(S: Stack): boolean;
 begin Empty:=S=nil;
 end;

procedure Push(var S: Stack; D: Element);
 var Pom: PClen;
 begin new(Pom);
       Pom^.data:=D;
       Pom^.dalsi:=S;
       S:=Pom
 end;
```

Implementace zásobníku

```
procedure Pop(var S: Stack; var D: Element);
 var Pom: PClen;
 begin if not Empty(S) then begin
          Pom:=S;
          D:=S^.data;
          S:=S^.dalsi;
          dispose(Pom)
       end else Error;
 end;
procedure Error;
 begin {řešení podle implementace}
 end;
```

Implementace zásobníku

```
var Z: Stack; {použití}
    R: string;
begin Init(Z);
      while not eof do begin
      {naplnění zásobníku řetězci}
         readln(R);
         Push(Z, R)
      end;
      while not Empty(Z) do begin
      {výpis celého zásobníku}
         Pop(Z, R);
         writeln(R)
      end
end.
```