

1.

Signatura datového typu

- a) popisuje vlastnosti operací datového typu
- b) popisuje sémantiku datového typu
- c) popisuje jména druhů a operací a druhy argumentů a výsledku
- d) je grafickým vyjádřením implementace datového typu

Čtvrtá možnost odpadá ihned, protože tím, že je datový typ abstraktní, existuje on sám i jeho signatura dříve než jakékoli implementace a vůbec na nich nezávisí. Sémantika operací je zachycena axiomami, nikoli signaturou, druhá možnost odpadá. První varianta říká prakticky totéž co druhá, takže zbývá jen třetí správná možnost.

2.

Signatura datového typu

- a) popisuje vlastnosti operací datového typu
- b) lze ji popsat graficky pomocí oválů, malých kroužků a spojovacích čar a písmen
- c) vyjadřuje se pomocí axiomů
- d) popisuje universum hodnot

Třetí možnost nedává smysl, neboť axiomami se k signatuře *přidávají* právě proto, aby bylo možno ADT kompletně popsat. Signatura nepopisuje ani vlastnosti operací, neboť právě ony se vyjadřují pomocí axiomů. První a třetí možnost tedy nedává smysl. Ještě méně smyslu dává poslední čtvrtá možnost, universum hodnot představuje pouze všechny vůbec možné hodnoty se kterými ADT může pracovat, vlastnosti této množiny ovšem ADT vůbec nezachycuje. Například ADT fronta ned celými čísly nijak blíže celá čísla nespecifikuje. Zbývá jen možnost b), jakkoli přitom vypadá primitivně.

3.

Signatura datového typu

- a) je zájmová skupina naturistů
- b) popisuje syntaxi datového typu
- c) sestává z druhů, jmen operací a axiomů
- d) popisuje sémantiku datového typu

Popis sémantiky, tj. „jaký má význam a co se stane když...“ zajišťují axiomami datového typu. varianta d) odpadá. Stejně tak odpadá varianta c) neboť axiomami právě součástí signatury nejsou. Protože naturisté jsou osoby programově holdující návratu k přírodě, kde se s abstraktními datovými typy nesečkáme (nanejvýš s jejich implementacemi), platí varianta b).

4.

Datová struktura

- a) je konkrétní realizací datového typu
- b) je základním stavebním kamenem datových typů
- c) slouží k ukládání polí v počítači

- d) vznikne ze signatury doplněním axiomů

Přidáním axiomů k signatuře vzniká abstraktní datový typ, varianta d) neplatí. Pole v počítači je sice příkladem datové struktury, nikoli však jediným, varianta c) je přinejmenším podezřelá. U variant a) a b) je nutno rozhodnout, zda se ADT odvozuje od datové struktury či naopak. Většinou se konkrétní odvozuje od abstraktního, takže přednost dostane varianta a).

5.

Datová struktura

- a) popisuje jména druhů, jména operací a druhy parametrů
- b) popisuje syntaxi datového typu
- c) je plně popsána soustavou axiomů
- d) vznikne z datového typu volbou nosičů druhů a implementací operací

Datová struktura je realizací určitého ADT. Všechny její podstatné vlastnosti jsou tedy v odpovídajícím ADT popsány, takže ona sama nemusí nic poposovat, varianty a) a b) odpadají. Naopak, soustava axiomů k popisu nestačí, axiomy neříkají, nad jakým druhem dat se operace provádějí, varianta c) také odpadá. Zbývá poslední varianta d).

6.

Datový typ

- a) je plně popsán soustavou axiomů
- b) je určen druhy dat a typickými operacemi
- c) je určen typickými operacemi
- d) je určen druhy dat

To je otázka pamatování si definice. Pro datový typ musí být známo, s jakými vůbec hodnotami může pracovat (= druhy dat) a také musí být známo, jaké operace nad těmito hodnotami lze v rámci datového typu provádět (= typické operace). Platí b).

7.

Druh dat

- a) popisuje syntaxi datového typu
- b) popisuje sémantiku datového typu
- c) je povolená množina hodnot
- d) lze popsat graficky pomocí oválů, malých kroužků a spojovacích čar a písmen

Toto je také otázka směrem do definic – druh dat je to, co se zmiňuje ve variantě c).

8.

Soustava axiomů

op(x, true) = true

op(x, false) = x

op(x, y) = op(y, x)

- a) popisuje operaci **and**
- b) popisuje operaci **xor**
- c) popisuje operaci **or**
- d) popisuje operaci **not**

Operace **op** je binární, operace **not** je unární, čtvrtá možnost odpadá. Po dosazení false za x v prvním axiomu vidíme, že **op** nemůže být operací **and**. Po dosazení false za x ve druhém axiomu vidíme, že se nemůže jednat o operaci **xor**. Zbývá tudíž jen **or** – třetí možnost.

9.

Soustava axiomů

op(x, true) = x

op(x, false) = false

op(x, y) = op(y, x)

- a) Popisuje operaci **or**
- b) Popisuje operaci **and**
- c) Popisuje operaci **xor**
- d) Popisuje operaci **not**

Operace **op** je binární, operace **not** je unární, čtvrtá možnost odpadá. Po dosazení false za x v prvním axiomu vidíme, že **op** nemůže být operací **or**. Po dosazení false za x ve druhém axiomu vidíme, že se nemůže jednat o operaci **xor**. Zbývá tudíž jen **and** – druhá možnost.

10.

Je dán datový typ množina bez opakování prvků. Kterým výrazem je nutno nahradit písmena xxx v následujících axiomech, popisujících mazání prvku z množiny?

del(x, []) = []

del(x, ins(y, s)) = if eq(x, y)

then xxx

else ins(y, del(x, s))

- a) ins(x, s)
- b) del(x, s)
- c) s
- d) x

Jsme tázáni, co se stane, když do množiny vložíme prvek x vzápětí jej opět smažeme. Pokud na začátku prvek x již v množině byl, pokus o vložení množinu s nezmění (množina je bez opakování!) a další operace mazání jej skutečně z množiny odstraní,

takže celou událost lze popsat výrazem $\text{del}(x,s)$. Žádný z dalších tří nabízených výrazů a), c), d) se nehodí.

Nyní pozor. Pokud na začátku prvek x v množině nebyl, zřejmě vložení a následné smazání zanechá množinu s beze změny. Naštěstí v takovém případě formálně platí $s = \text{del}(x,s)$.

Ať již tedy prvek x v množině s byl nebo nebyl, platí $\text{del}(x, \text{ins}(x,s)) = \text{del}(x,s)$. Jediná možná varianta odpovědi je b).

11.

Je dán datový typ množina bez opakování prvků. Kterým výrazem je nutno nahradit písmena xxx v následujících axiomech, popisujících velikost množiny?

```

card([ ])=0
card(ins(y,s))= if in(y, s)
                    then xxx
                    else succ(card(s))

```

- a) $\text{succ}(\text{card}(\text{ins}(y,s)))$
- b) $\text{card}(s)$
- c) s
- d) $\text{card}(y)$

Jsme tázáni, jak velká bude množina poté, co se do ní pokusíme vložit prvek, který už je v ní obsažen. Protože množina je bez opakování, prvek se nevloží a velikost množiny se nezmění. To znamená, že pravdivá je varianta b).

12.

Popište abstraktní datový typ „množina s opakováním prvků“ s operacemi:

[]	„prázdná množina“
ins	„vložení prvku do množiny“
isIn	„test příslušnosti prvku k množině“
del	„smazání prvku z množiny“
eq	„rovnost dvou množin“
card	„mohutnost množiny“

Signatura (1 bod), axiomy (0+1+1+1+1+1):

Řešení

typ: MNOŽINA(ELEMENT)

parametr: typ prvků ELEMENT

požadavky na parametr:

druhy: Elem

operace: $\text{eq}(_,_): \text{Elem}, \text{Elem} \rightarrow \text{Bool}$

použité typy: ELEMENT, přirozená čísla, logické hodnoty

druhy: Elem, Bool, Set, Nat

operace:

```

[ ]: Set                                (prázdná množina)
ins(,_): Elem,Set ->Set (vložení prvku)
del(,_): Elem,Set ->Set (zrušení "-")
in(,_): Elem,Set ->Bool(test přísluš.)

```

ins(,): Elem,Set ->Set (vložení prvku)

