

7. laboratorijas darbs

Ir *noskaņojamā* pakotne *Stack*, kas ir uzrakstīta *Ada* valodā un paredzēta *steka* programmēšanai. Steka pamats ir ieraksts *StackType*, kas satur *diskriminanta daļu*. Ierakstam ir trīs lauki:

1. *Maksimālais* elementu daudzums stekā (piemēram, *MaxSize*).
2. *Aktuālais* elementu daudzums stekā (piemēram, *Counter*).
3. *Viendimensiju masīvs* steka implementēšanai (piemēram, *St*).

Maksimālo elementu daudzumu norāda, lietojot ieraksta *diskriminanta daļu*.

Steka elementu *adresēšanai* izmanto *Integer* tipa indeksus.

Noskaņojamā pakotnes daļa (steka *elementu tips*):

```
type ElemType is delta <> digits <>;
```

Pakotnē paredzēt trīs operācijas:

1. Procedūru *Push*(*<Stack>*, *<Elem>*) elementa *izvietošanai* stekā.
2. Procedūru *Pop*(*<Stack>*, *<Elem>*) elementa *izslēgšanai* no steka.
3. Procedūru *Print*(*<Stack>*) steka elementu *izvadei ekrānā*.

Darba gaitā *kontrolē* elementa izvietošanas/izslēgšanas iespējamību, apstrādājot izņēmumu *CONSTRAINT_ERROR* procedūrās (1) un (2). Piemēram, tukšā steka gadījumā var atgriezt vērtību *-1*. Steka elementu izvadei ekrānā ieteicams izmantot pakotni *Decimal_IO*.

Iespējamais galvenās programmas fragments:

```
...  
type Elem is delta 0.01 digits 5;  
...  
S : StackType(2);  
...  
Push(S, 1.0);  
Push(S, 2.0);  
Push(S, 3.0); -- PUSH Error!  
  
Print(S);      -- [1.0 2.0]  
  
Pop(S, X);  
Put(X);        -- 2.0  
Print(S);      -- [1.0]
```