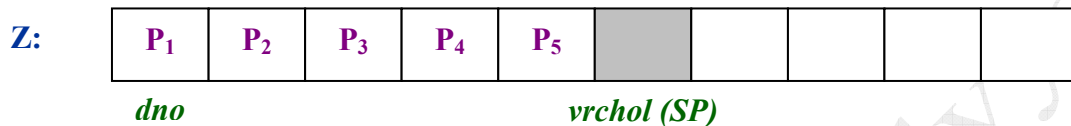


Zásobníky a fronty

Zásobníky

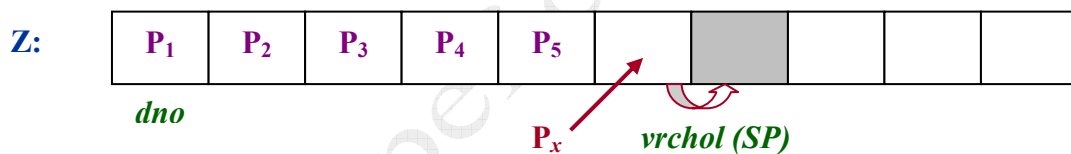
- **Zásobník** je datová struktura **Z**, do které se objekty vkládají a vybírají podle strategie **LIFO** (Last In First Out).



- **Vrchol** zásobníku (SP – stack pointer) ukazuje na první volnou položku zásobníku.
- **Dno** zásobníku obsahuje první vloženou položku, která podle strategie LIFO opustí zásobník jako poslední.

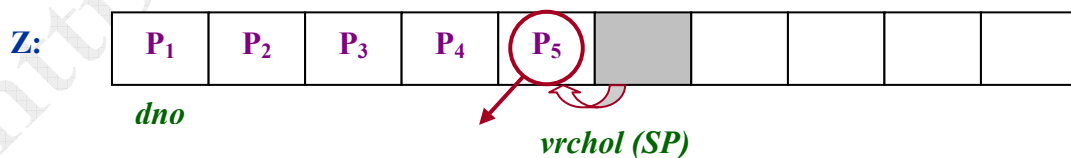
Operace se zásobníkem

- **Vložení položky (P_x)** - operace vloží položku P_x na vrchol zásobníku, tj. na pozici **SP**.
 - Tato operace patří i mezi základní instrukce strojového kódu – **PUSH**



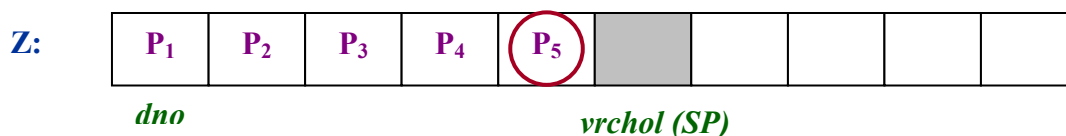
- Vrchol zásobníku se po vložení položky P_x zvýší o jedničku.

- **Vyjmutí položky** – operace vyjme položku z vrcholu (z pozice SP-1) zásobníku.
 - Tato operace patří i mezi základní instrukce strojového kódu – **POP**



- Vrchol zásobníku se po výběru položky sníží o jedničku.

- **Položka na vrcholu** – operace vrací položku na vrcholu zásobníku bez jejího odstranění



- **Velikost zásobníku** – operace vrací počet položek v zásobníku
- **Prázdný zásobník** – operace vrací hodnotu pravda (*true*), jestliže zásobník neobsahuje žádnou položku

Implementace zásobníku

- Implementace zásobníku se provádí pomocí **pole** pevné nebo proměnné délky. Implementaci je možné provést i pomocí **spojového seznamu** (viz spojové seznamy).

- **Deklarace zásobníku:**

```
type Polozka = typ polozky (integer, real, record, ...);
```

```
var Zasobnik: array [1..MAX] of Polozka;
    SP: integer;
```

```
begin
    SP:=1
end.
```

- Položkou může být jakákoliv hodnota, tj. celé číslo (*integer*), reálné číslo (*real*), záznam (*record*) nebo jiná.
- Hodnota **MAX** udává maximální rozsah zásobníku. Použití této hodnoty je nutné při deklaraci pole s pevnou délkou.
- Hodnota **SP** udává vrchol zásobníku. Na začátku, je-li zásobník prázdný, obsahuje **SP** hodnotu 1.

Implementace operací se zásobníkem

- **Vložení položky (P_x)**

```
procedure Vlozeni_polozky(Px:Polozka);
begin
    Zasobnik[SP]:=Px;
    SP:=SP+1
end;
```

■ Vyjmutí položky

```
function Vyjmuti_položky:Polozka;  
begin  
    Vyjmuti_položky:=Zasobnik[SP-1];  
    SP:=SP-1  
end;
```

■ Položka na vrcholu

```
function Polozka_na_vrcholu:Polozka;  
begin  
    Polozka_na_vrcholu:=Zasobnik[SP-1]  
end;
```

■ Velikost zásobníku

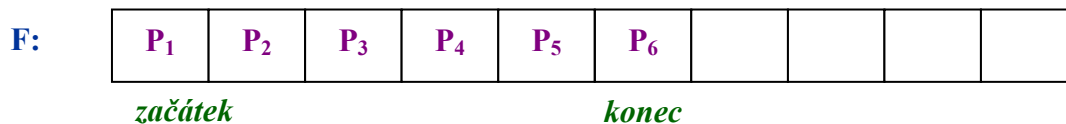
```
function Velikost_zasobniku:Integer;  
begin  
    Velikost_zasobniku:=SP-1  
end;
```

■ Prázdný zásobník

```
function Prazdny_zasobnik:Boolean;  
begin  
    Prazdny_zasobnik:=(SP=1)  
end;
```

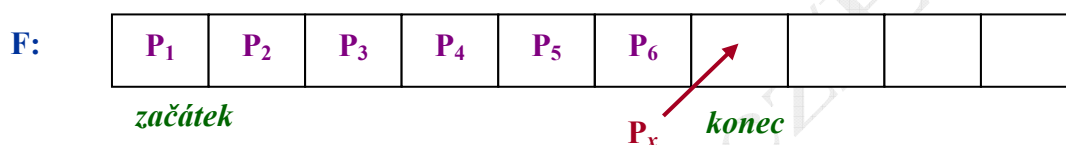
Fronty

■ **Fronta** je datová struktura **F**, do které se objekty vkládají a vybírají podle strategie **FIFO** (*First In First Out*).

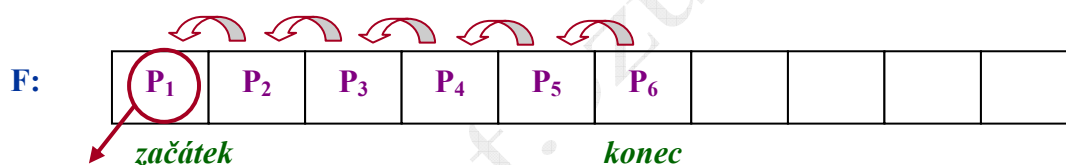


Operace s frontou

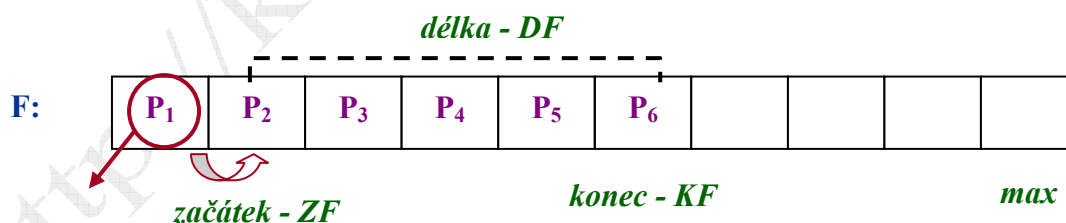
■ **Vložení položky (P_x)** - operace vloží položku P_x na konec fronty.



Vyjmutí položky – operace vyjme položku ze začátku fronty.

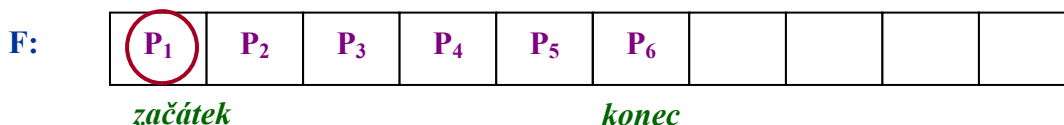


- Nejprve se vyjme položka ze začátku fronty.
- Poté se všechny ostatní položky přesunou o jednu pozici směrem k začátku fronty.
 - Aby nebylo nutné při každém vyjmutí položky z fronty přesouvat všechny ostatní položky, lze tuto operaci řešit i pomocí tzv. **cyklické fronty**.



- Posun začátku fronty při vyjmutí položky:
$$ZF := (ZF + 1) \bmod max$$
- Posun konce fronty při vložení položky:
$$KF := (KF + 1) \bmod max$$
- Délka fronty:
$$DF := (KF - ZF + max + 1) \bmod max$$
- Operace **mod** je zbytek po celočíselném dělení.

- **Položka na začátku** – operace vrací položku na začátku fronty bez jejího odstranění



- **Délka fronty** – operace vrací počet položek ve frontě.
- **Prázdná fronta** – operace vrací hodnotu pravda (*true*), jestliže fronta neobsahuje žádnou položku

Implementace fronty

- Implementace fronty se provádí pomocí **pole** pevné nebo proměnné délky. Implementaci je možné provést i pomocí **spojového seznamu** (viz spojové seznamy).

- **Deklarace fronty:**

```
type Polozka = typ polozky (integer, real, record, ...);

var   Fronta: array [1..MAX] of Polozka;
      ZF, KF, DF: integer;

begin
  ZF:=1; KF:=0; DF:=0
end.
```

- Položkou může být jakákoliv hodnota, tj. celé číslo (*integer*), reálné číslo (*real*), záznam (*record*) nebo jiná.
- Hodnota **MAX** udává maximální rozsah fronty. Použití této hodnoty je nutné při deklaraci pole s pevnou délkou.
- Hodnota **ZF** udává začátek, hodnota **KF** konec a hodnota **DF** délku fronty.
- Na začátku, je-li fronta prázdná, obsahuje **ZF** hodnotu 1, **KF** a **DF** jsou rovny nule.

Implementace operací s frontou

- **Vložení položky (P_x)**

- klasický přístup

```
procedure Vlozeni_polozky(Px:Polozka);
begin
  KF:=KF+1;
  Fronta[KF]:=Px
end;
```

- použití cyklické fronty

```
procedure Vlozeni_položky(Px:Polozka);  
begin  
    KF:=(KF+1) mod MAX;  
    Fronta[KF]:=Px;  
end;
```

■ Vyjmutí položky

- klasický přístup

```
function Vyjmuti_položky:Polozka;  
var i:integer;  
begin  
    Vyjmuti_položky:=Fronta[1];  
    for i:=2 to KF do Fronta[i-1]:=Fronta[i];  
    KF:=KF-1  
end;
```

- použití cyklické fronty

```
function Vyjmuti_položky:Polozka;  
var i:integer;  
begin  
    Vyjmuti_položky:=Fronta[ZF];  
    ZF:=(ZF+1) mod MAX  
end;
```

■ Položka na začátku

- klasický přístup

```
function Polozka_na_vrcholu:Polozka;  
begin  
    Polozka_na_vrcholu:=Fronta[1]  
end;
```

- použití cyklické fronty

```
function Polozka_na_vrcholu:Polozka;  
begin  
    Polozka_na_vrcholu:=Fronta[ZF]  
end;
```

■ Délka fronty

- klasický přístup

```
function Delka_fronty:Integer;  
begin  
    Delka_fronty:=KZ  
end;
```

- použití cyklické fronty

```
function Delka_fronty:Integer;  
begin  
    Delka_fronty:=(KF-ZF+MAX+1) mod MAX  
end;
```

■ Prázdná fronta

- klasický přístup

```
function Prazdna_fronta:Boolean;  
begin  
    Prazdna_fronta:=(KZ=0)  
end;
```

- použití cyklické fronty

```
function Prazdna_fronta:Boolean;  
begin  
    Prazdna_fronta:=((KF-ZF+MAX+1) mod MAX)=0)  
end;
```