

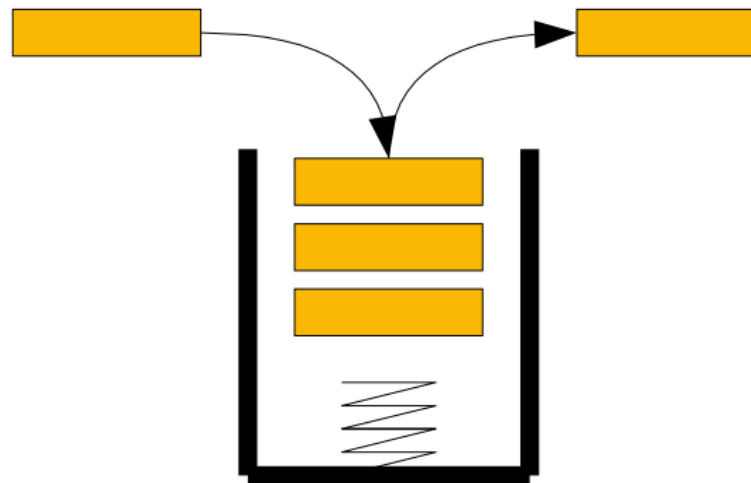
A decorative yellow wave graphic in the top right corner of the slide.

# Algoritmy cvičení

# Zásobník

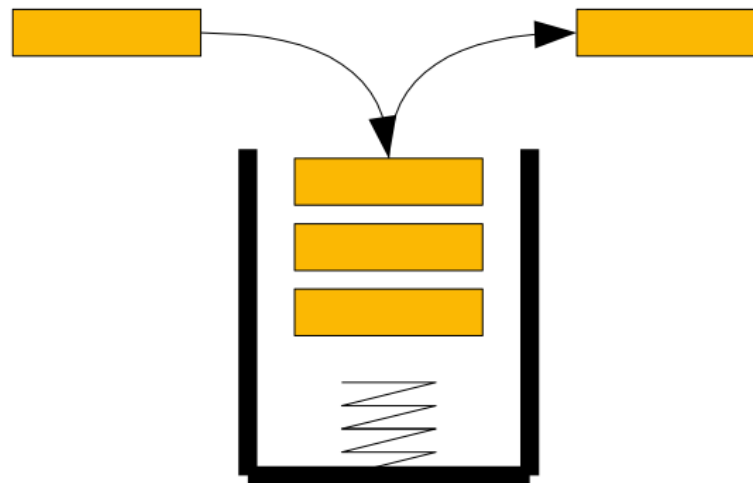
# Zásobník

- Abstraktní datová struktura
  - Vkládá prvky na začátek
  - Odebírá prvky ze začátku



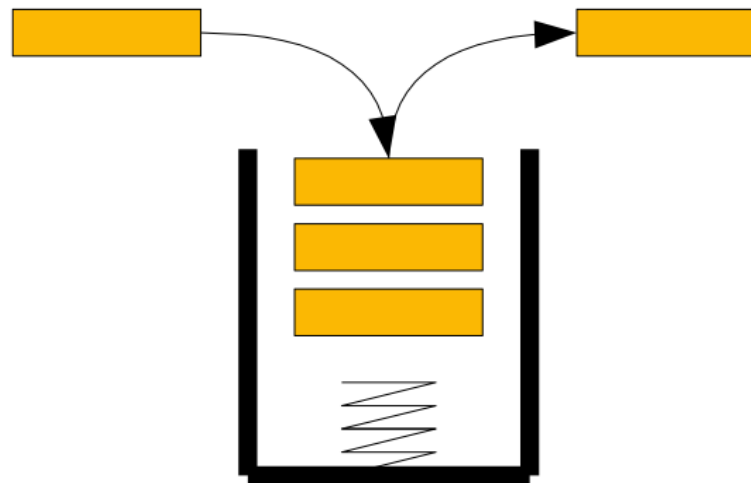
# Zásobník

- Prvek, který byl do zásobníku vložen poslední, bude odebrán jako ....



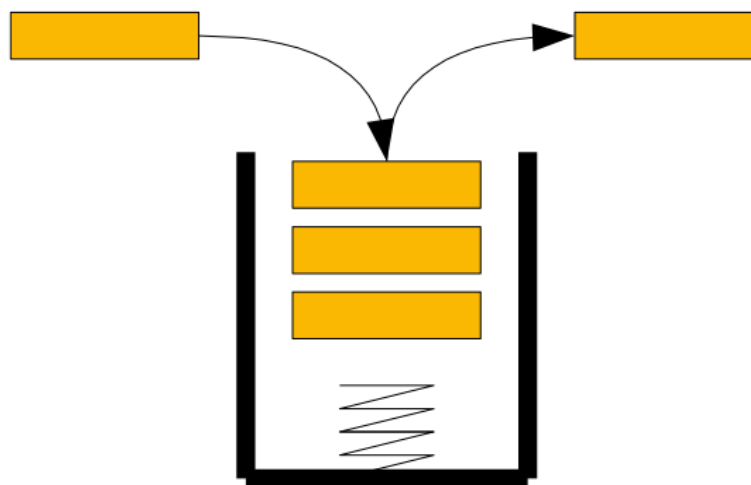
# Zásobník

- Prvek, který byl do zásobníku vložen poslední, bude odebrán jako první



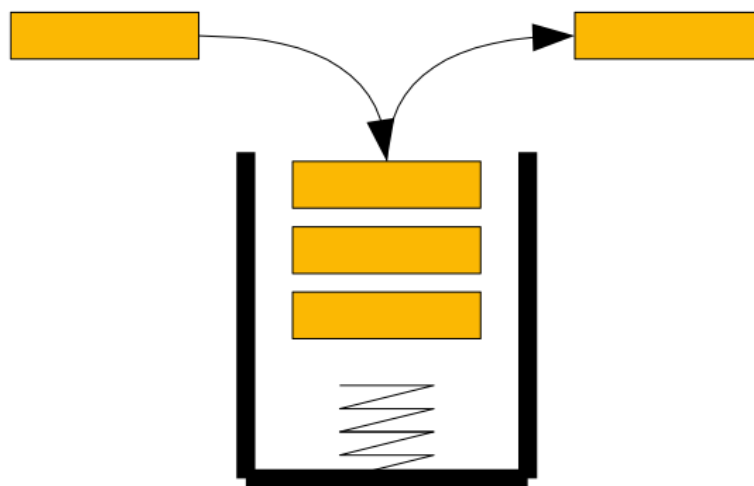
# Zásobník

- Prvek, který byl do zásobníku vložen poslední, bude odebrán jako první
- Prvek, který byl do zásobníku vložen jako první, bude odebrán jako ....



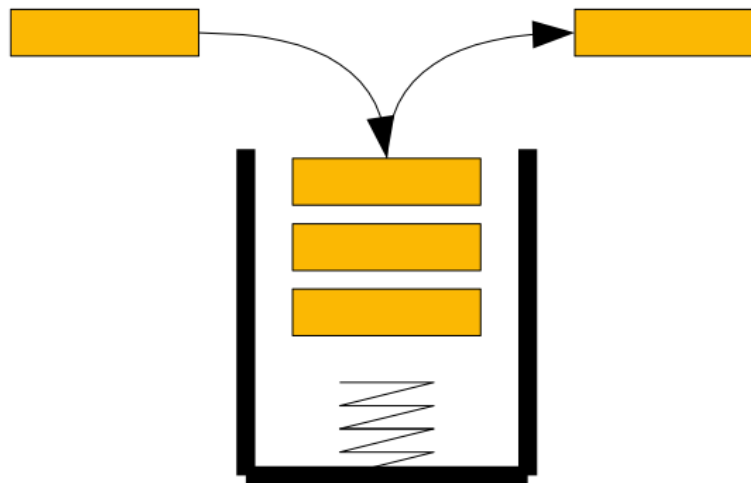
# Zásobník

- Prvek, který byl do zásobníku vložen poslední, bude odebrán jako první
- Prvek, který byl do zásobníku vložen jako první, bude odebrán jako poslední



# Zásobník

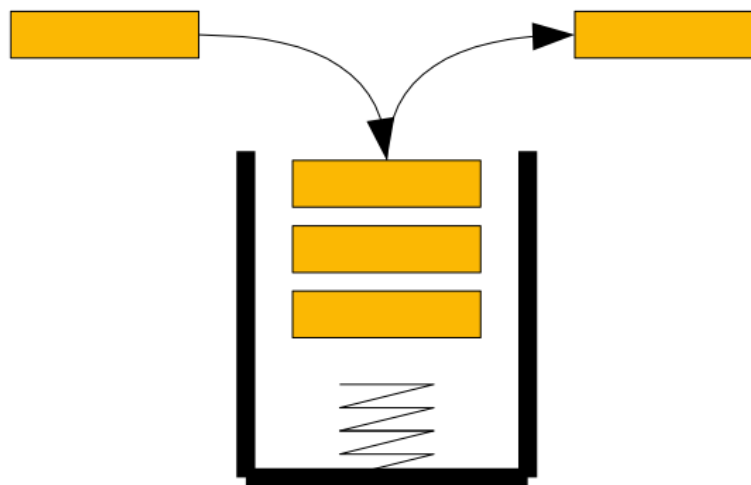
- Příklady ze života





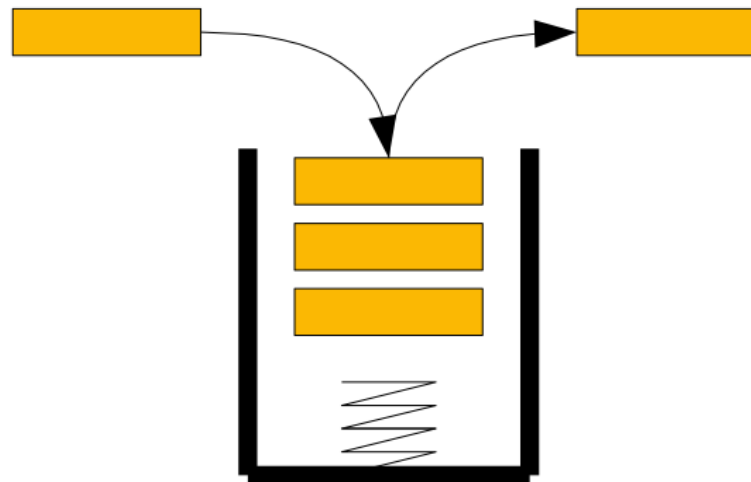
# Zásobník

- Příklady ze života
  - Oprerace zpět v textovém editoru
  - Věž z kostiček, přidávání a odebírání
  - Volání funkcí v programování



# Zásobník

- Vložení prvku – Push
- Odebrání prvku – Pop
- Prázdný zásobník



# Zásobník

```
public class CharZasobnik {
    private char[] zas;
    private int vrchol;
    final int MAX=100;

    public CharZasobnik() {
        zas = new char[MAX];
        vrchol = 0;
    }

    public boolean jePrazdny() {
        return (vrchol == 0);
    }

    public void vloz(char klic) {
        zas[vrchol] = klic;
        vrchol=vrchol+1;
    }

    public char odeber() {
        vrchol=vrchol-1;
        return zas[vrchol];
    }

    public char zjistihodnotu() {
        return zas[vrchol-1];
    }
}
```

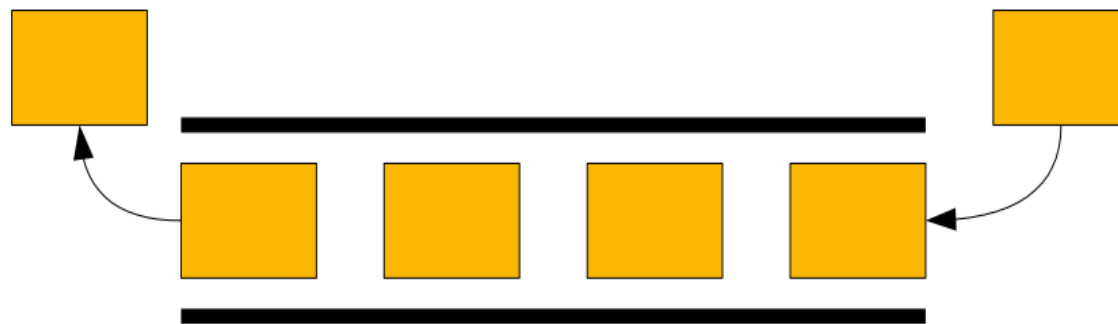
# Uzávorkování

- Vymyslete algoritmus, který zkontroluje správnost uzávorkování
  - `((()))`
  - `( )))`
  - `<[()](>`
  - `<[ ( ] )>`

# Fronta

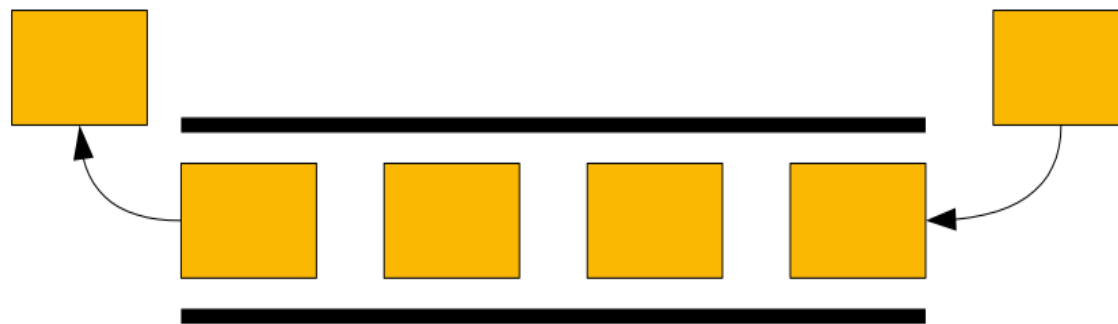
# Fronta

- Abstraktní datová struktura
  - Vkládá prvky na konec
  - Odebírá prvky ze začátku



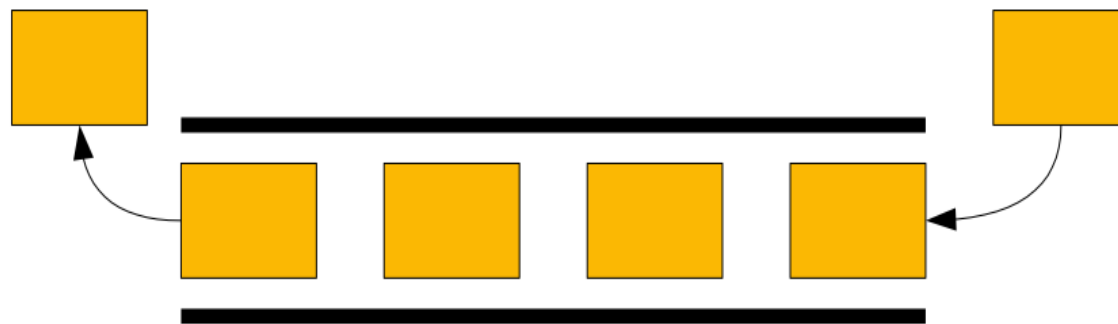
# Fronta

- Prvek, který byl do fronty přidán jako první, bude odebrán jako ...



# Fronta

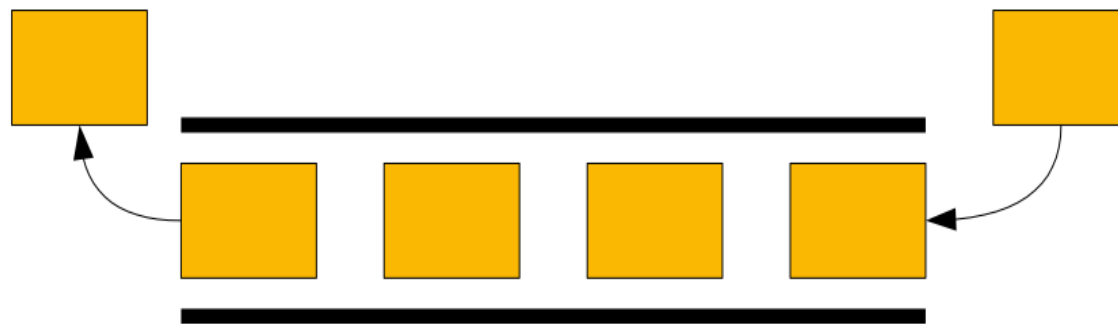
- Prvek, který byl do fronty přidán jako první, bude odebrán jako první
- Prvek, který byl do fronty přidán poslední, bude odebrán jako ...





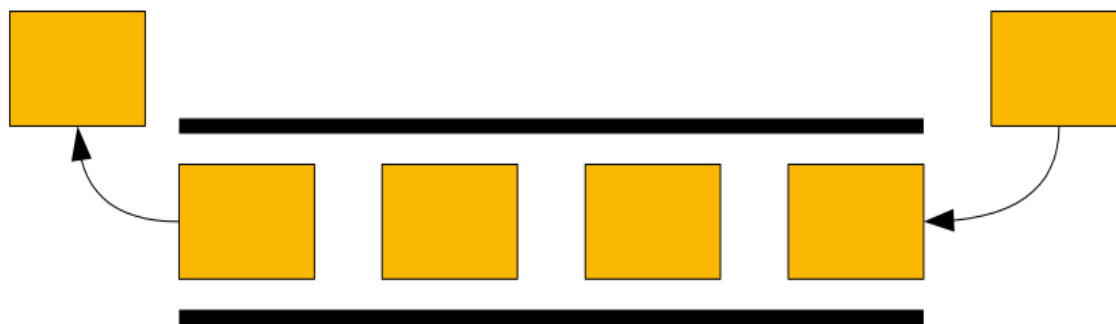
# Fronta

- Prvek, který byl do fronty přidán jako první, bude odebrán jako první
- Prvek, který byl do fronty přidán poslední, bude odebrán jako poslední



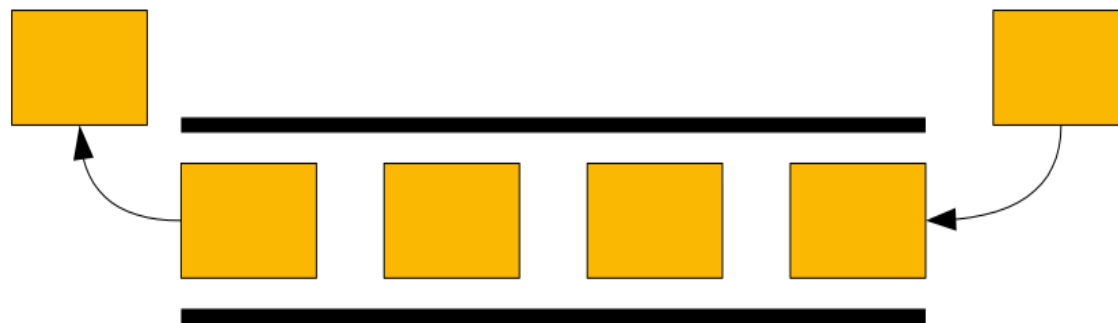
# Fronta

- Příklady ze života



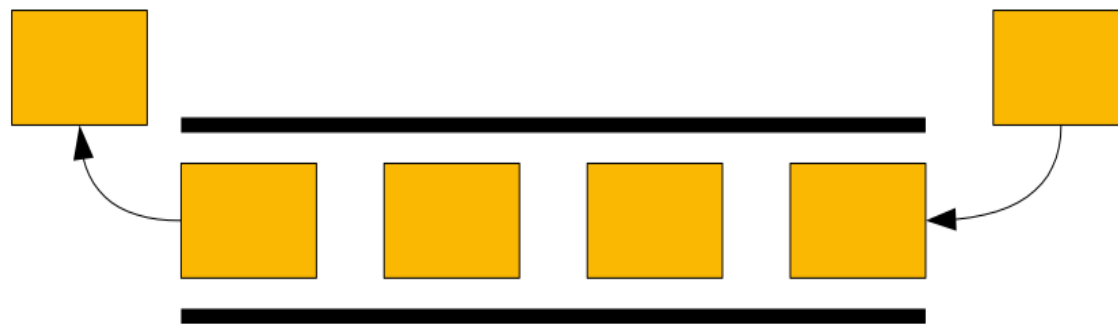
# Fronta

- Příklady ze života
  - Fronta v obchodě
  - Tisková fronta
  - Postupné plnění úkolů



# Fronta

- Vložení prvku – Push
- Odebrání prvku – Pop
- Prázdná fronta



# Fronta

- Po určitém počtu kroků se pole zaplní



# Fronta

- Po určitém počtu kroků se pole zaplní



- a tak se pokračuje od začátku



# Fronta

```
class FrontaInt {
    private int[] fronta;
    private int zacatek, konec;
    final int delka = 100;

    public FrontaInt() {
        fronta = new int[delka];
        zacatek = 0;
        konec = 0;
    }

    public boolean jePrazdna() {
        return (zacatek == konec);
    }

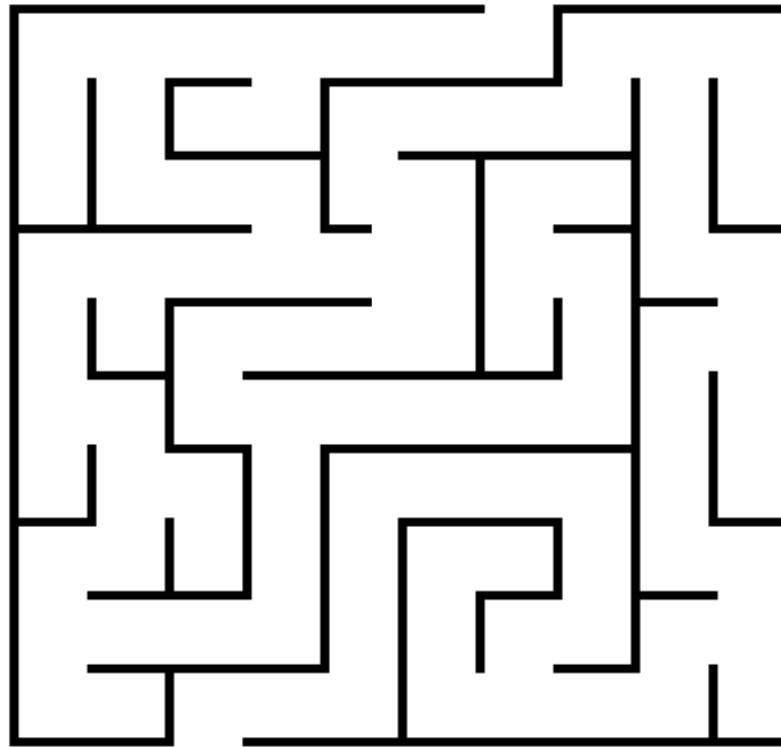
    public void vloz(int klic) {
        fronta[konec++] = klic;
        konec = konec % delka;
    }

    public int odeber() {
        int v = fronta[zacatek];
        zacatek = (zacatek+1) % delka;
        return v;
    }

    public int zjistHodnotu() {
        return fronta[zacatek];
    }
}
```

# Bludiště

- Vymyslete algoritmus, který nalezne nejkratší cestu bludištěm





# Bludiště

- Vymyslete algoritmus, který nalezne nejkratší cestu bludištěm

