



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MŠMT
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

Programování a algoritmizace

Maximální hodnota a suma prvků v poli

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204



Ing. et Ing. Erik Král, Ph.D.
ÚPKS
Fakulta aplikované informatiky

Obsah

Maximální hodnota prvků v poli

Suma prvků v poli

Průměrná hodnota prvků v poli

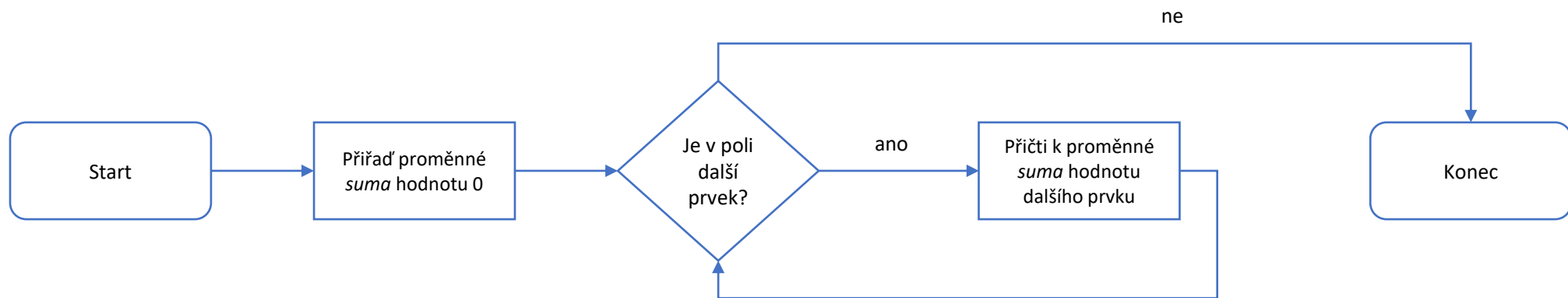
Úvod

- V následujících snímcích probereme algoritmy hledání maximální hodnoty v poli a sumy prvků v poli z hlediska jejich implementace.
- Na těchto příkladech si demonstrováme práci s jednorozměrným polem [1] a zaměříme se i na program z hlediska paměti RAM.

Suma prvků v poli

- Algoritmus výpočtu sumy prvků v poli je relativně jednoduchý.
- Nejprve si nadefinujeme proměnnou *suma* a přiřadíme jí hodnotu 0 a potom k ní postupně přičítáme další hodnoty prvků v poli.
- Variantou může být, že jako výchozí hodnotu přiřadíme proměnné *suma* hodnotu prvního prvku v poli a potom pokračujeme až od druhého prvku.

Suma prvků v poli



Algoritmus a paměť

- Algoritmus si alokuje paměť pro parametry, lokální proměnné a další hodnoty na zásobníku (Stack) a pro dynamicky alokované objekty alokuje paměť na haldě (Heap).
- V příkladech je **zjednodušeně** demonstrováno využití paměti z hlediska zásobníku a haldy.
- Práce se zásobníkem je ve skutečnosti složitější a v příkladech jsou zobrazeny **pouze proměnné přímo související s algoritmem** a jsou vynechány uložené hodnoty registrů nebo návratové hodnoty. Také pořadí předávaných argumentů a parametrů metody může být jiné.

Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}  
  
Console.WriteLine(suma);
```

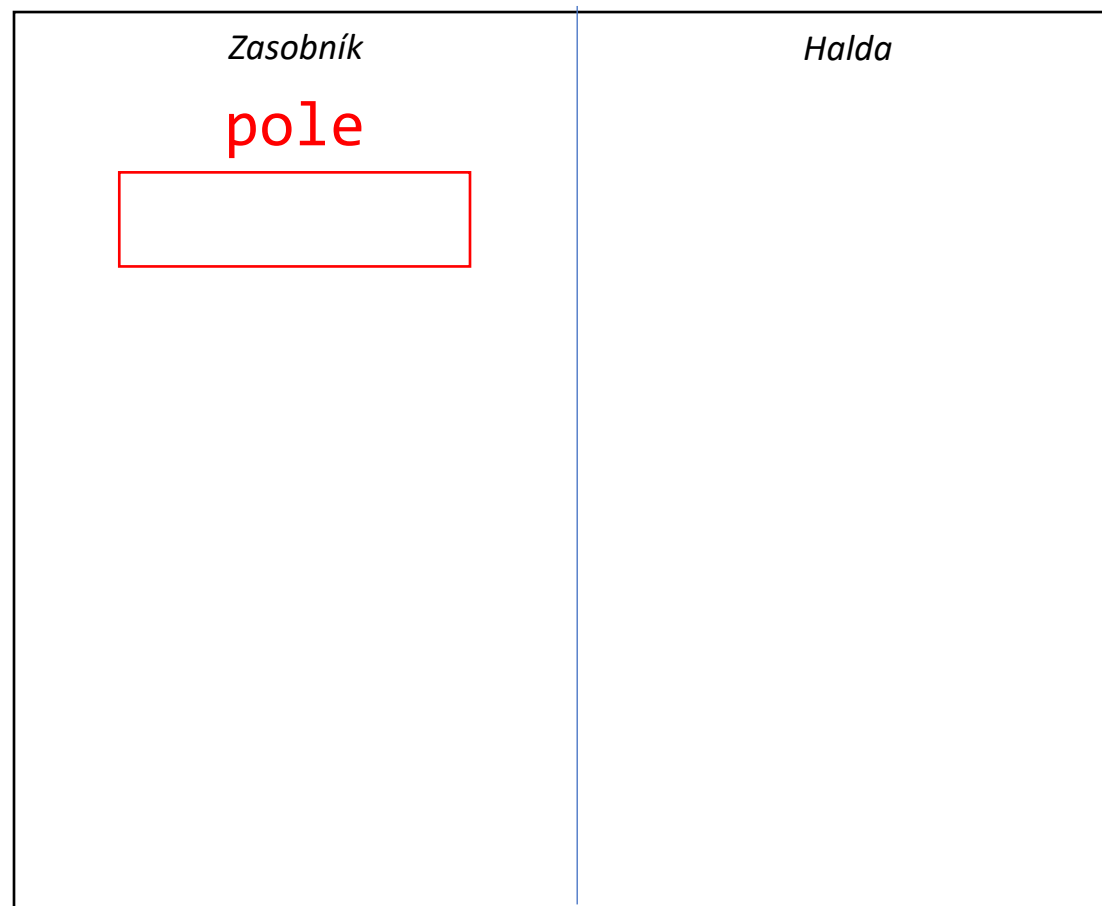
Paměť RAM

Zasobník	Halda

Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

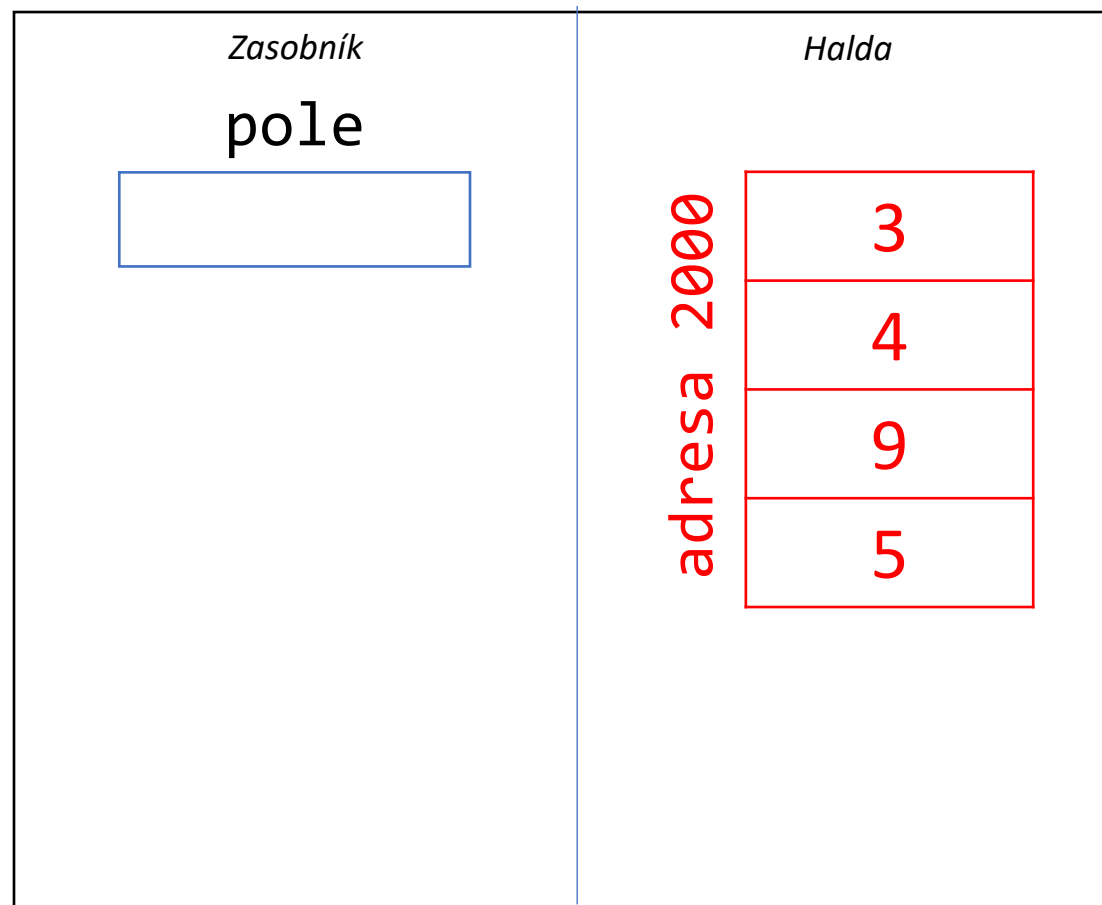
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

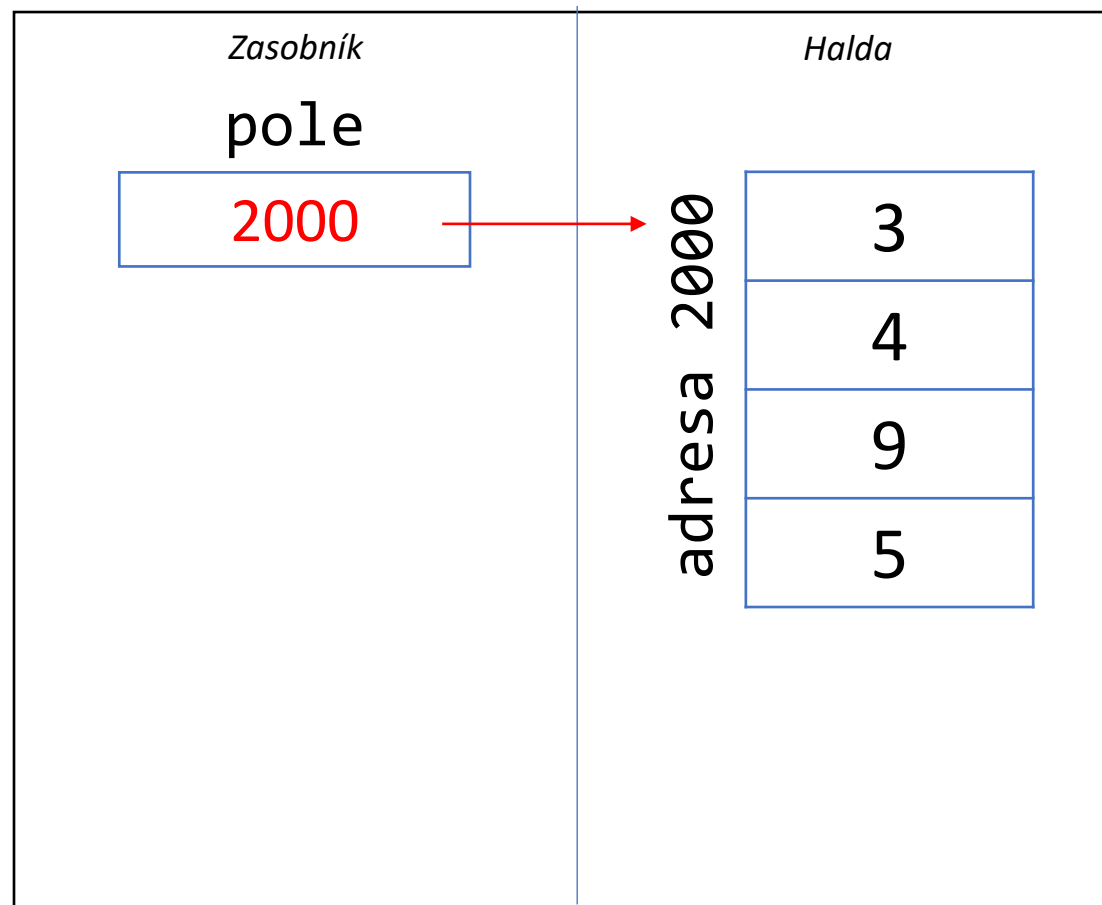
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

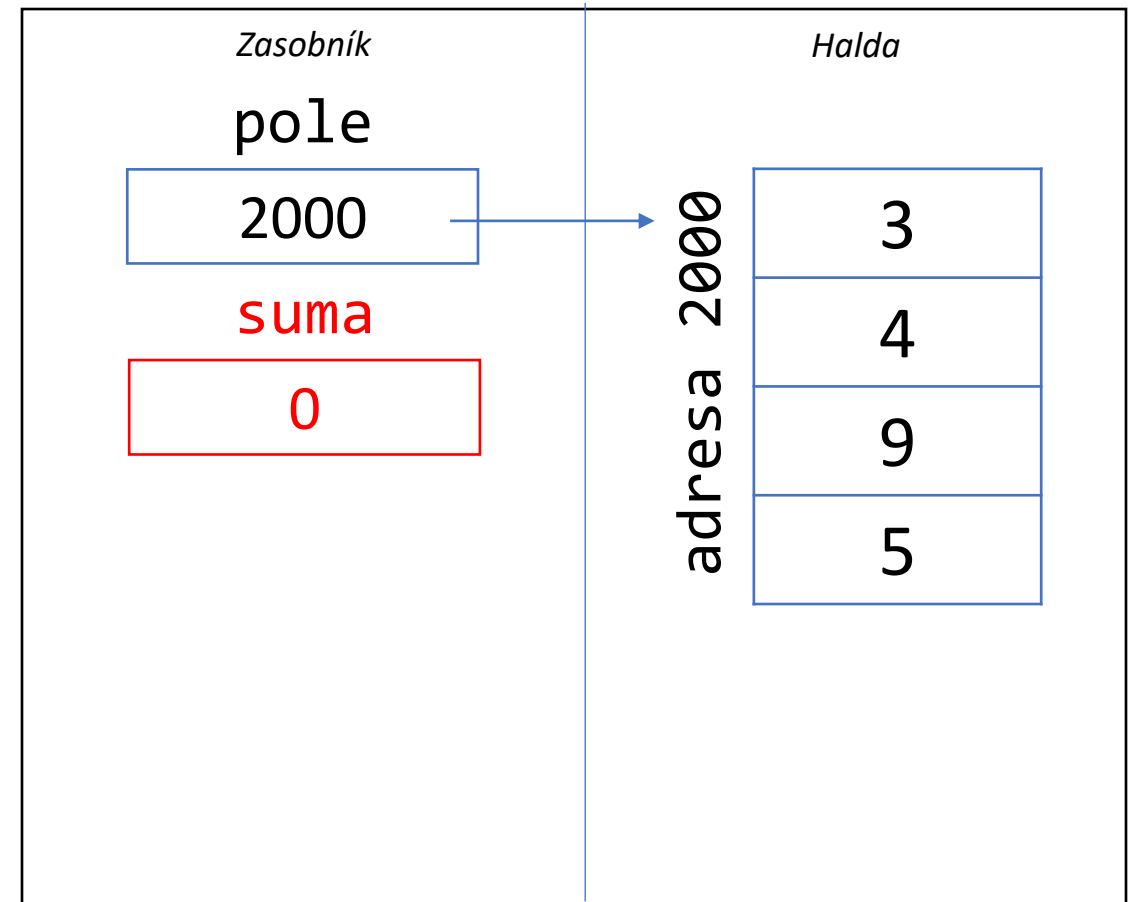
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
int suma = 0;
```

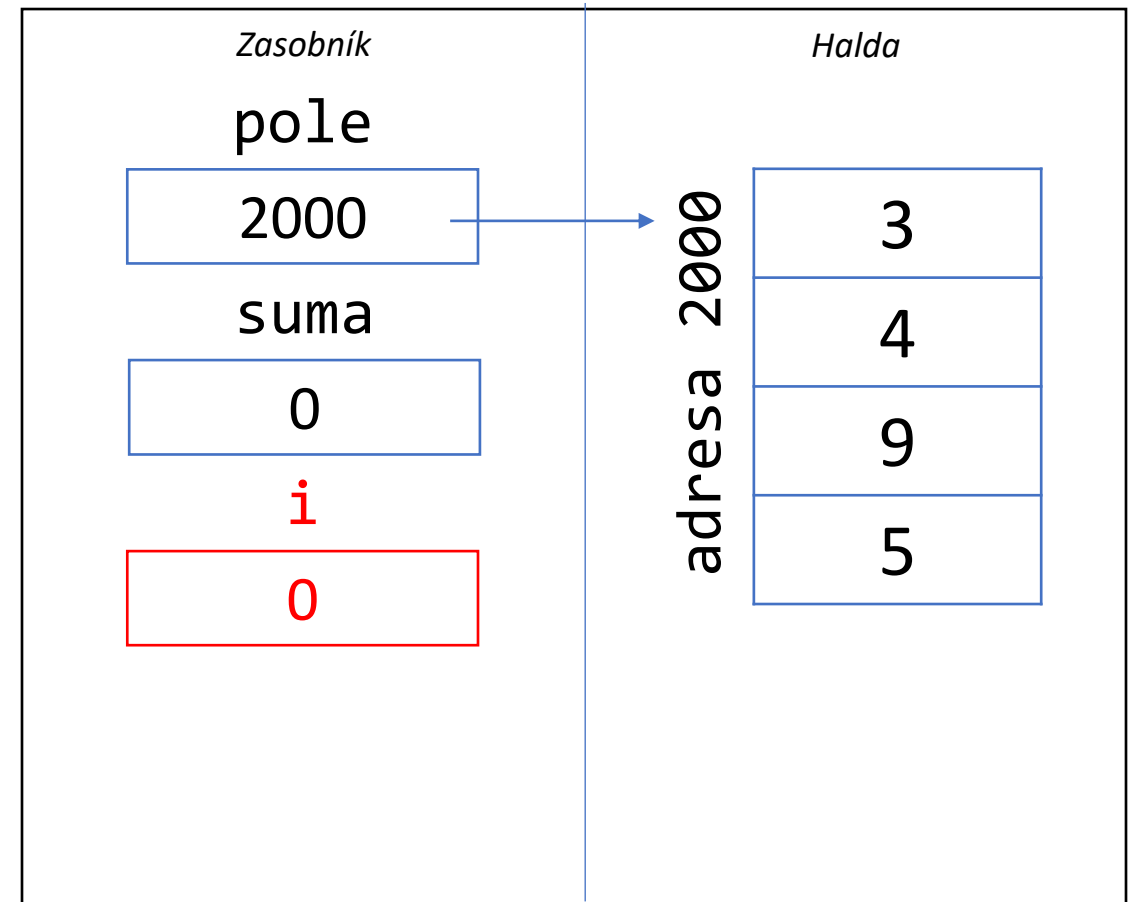
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
  
}
```

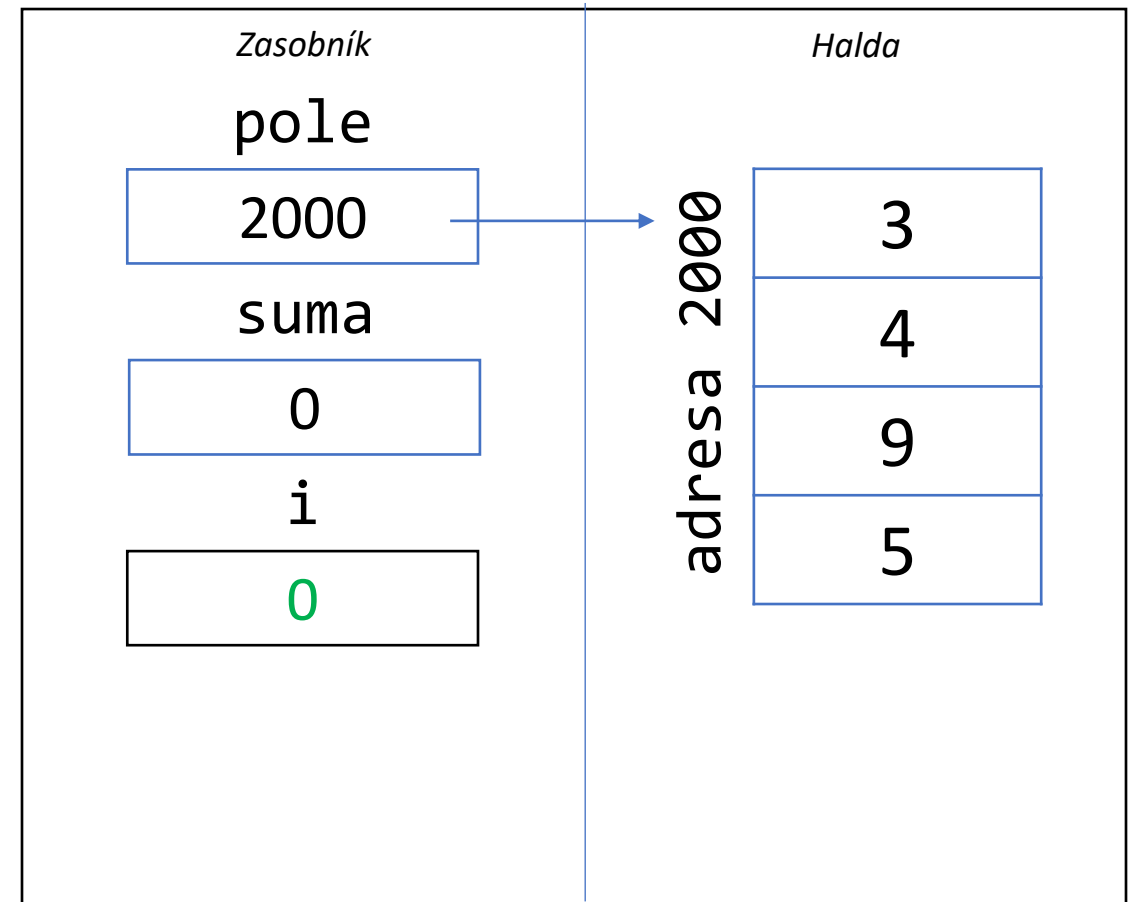
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
  
}
```

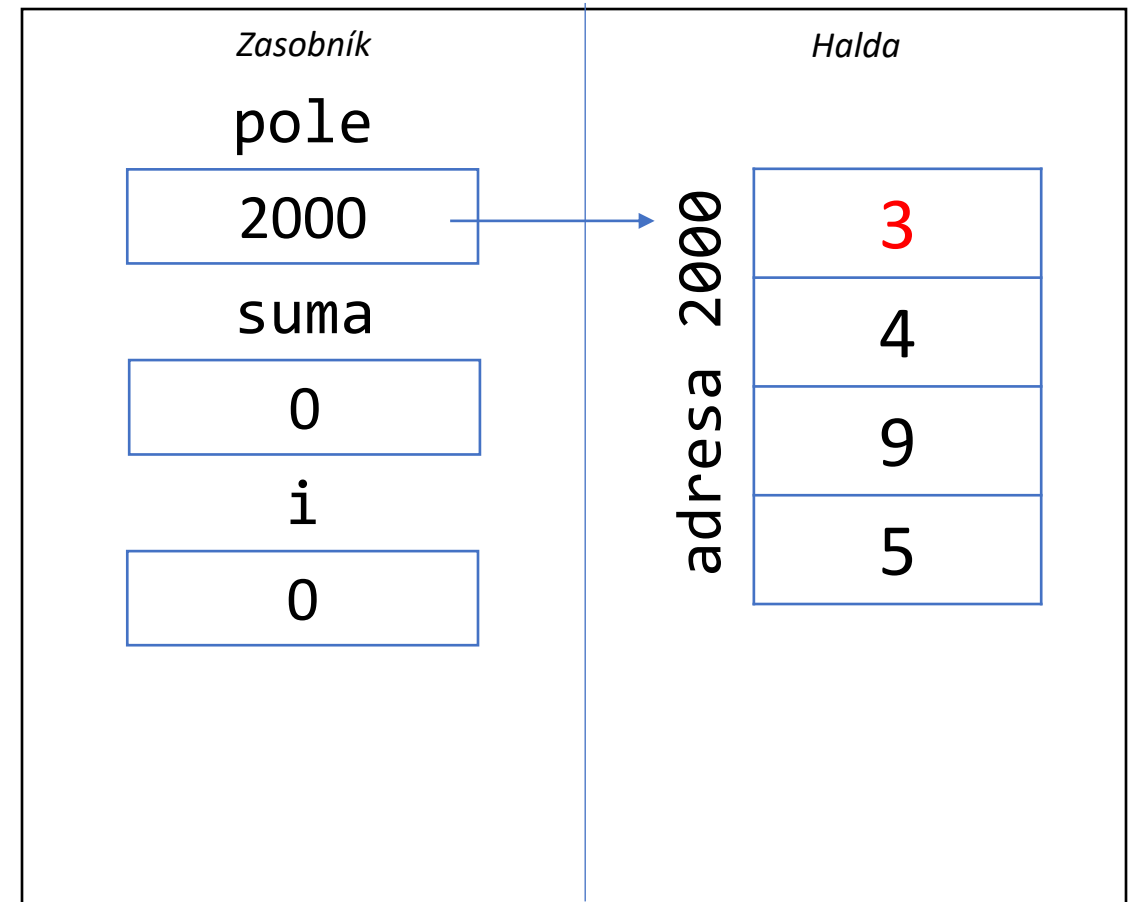
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
}
```

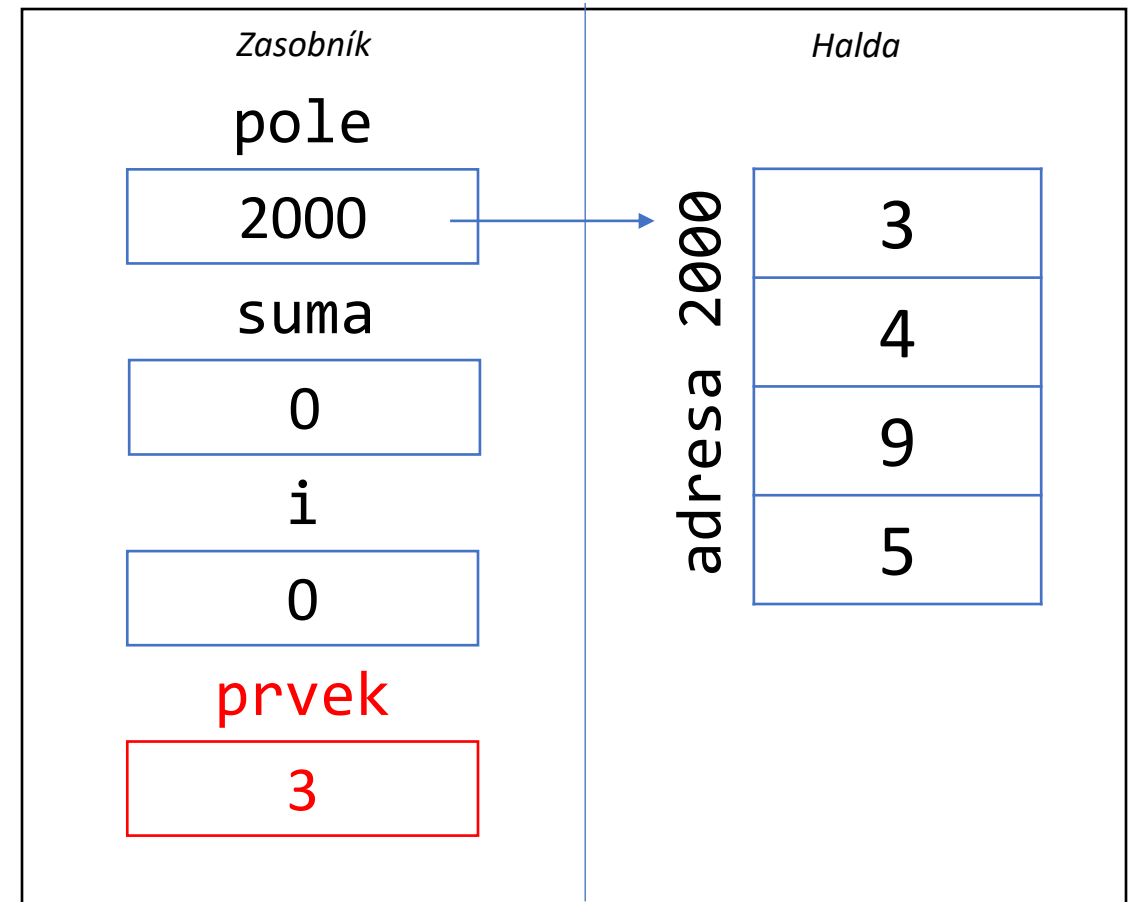
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
}
```

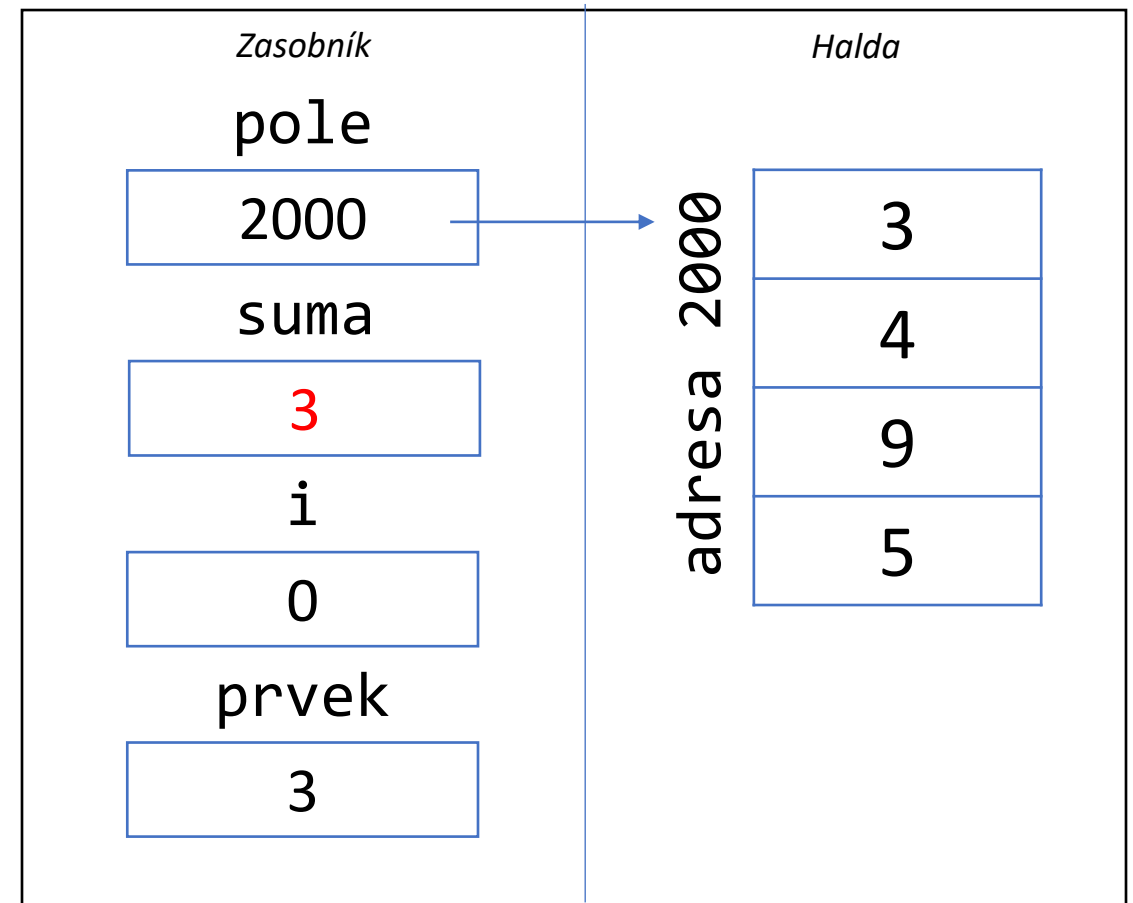
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

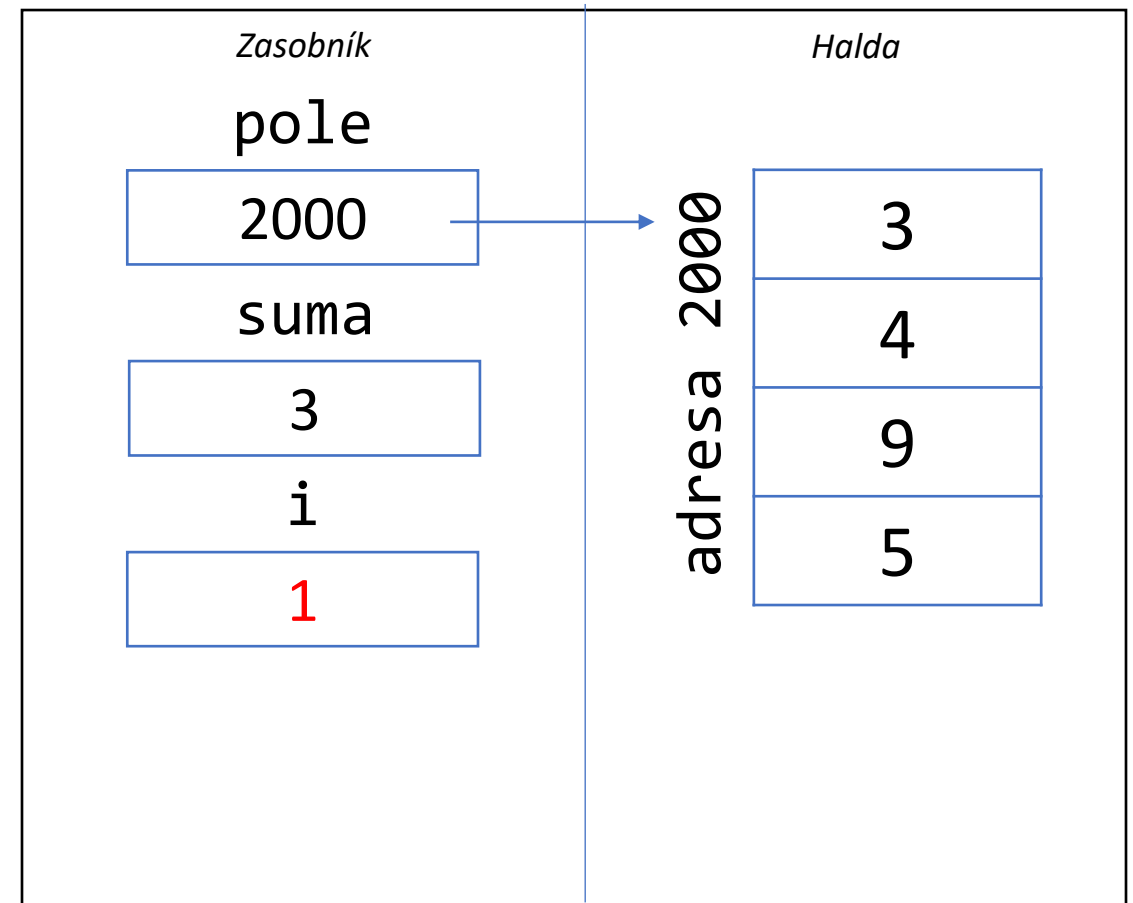
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

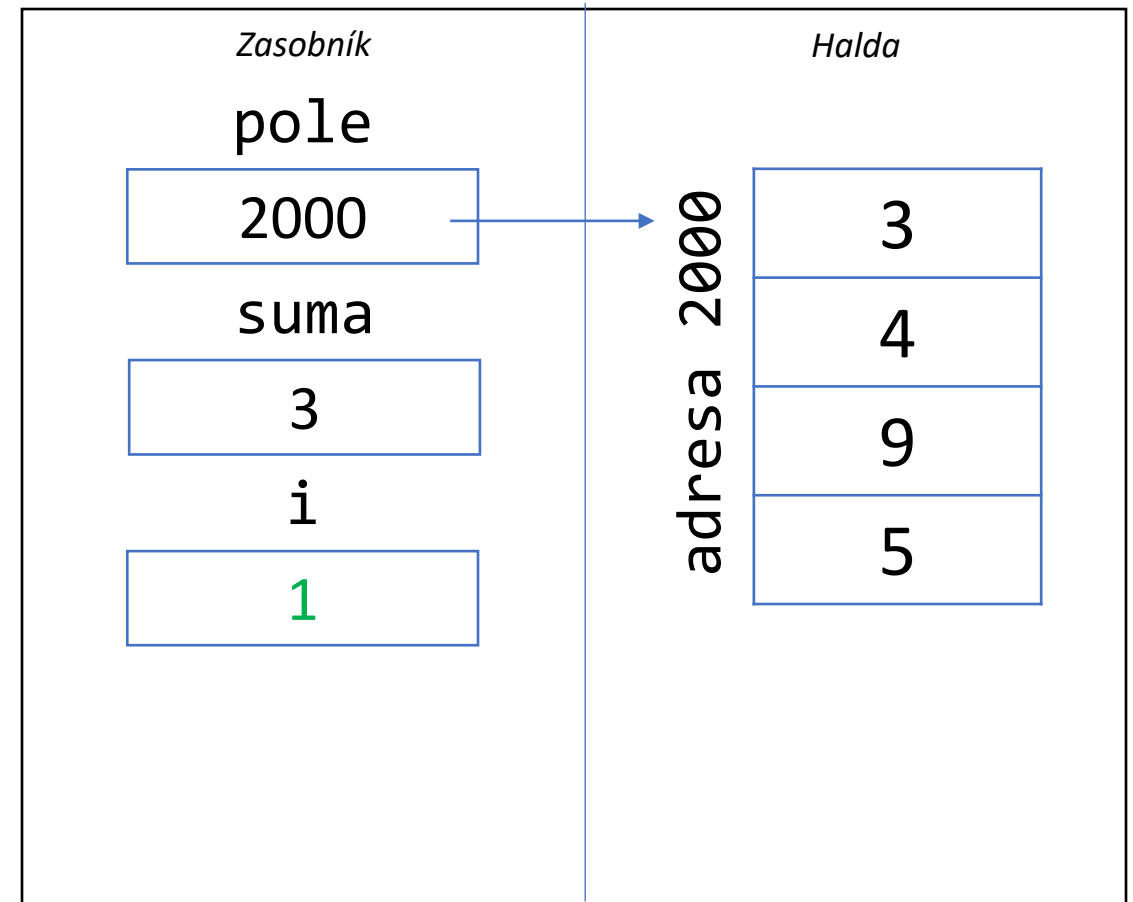
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

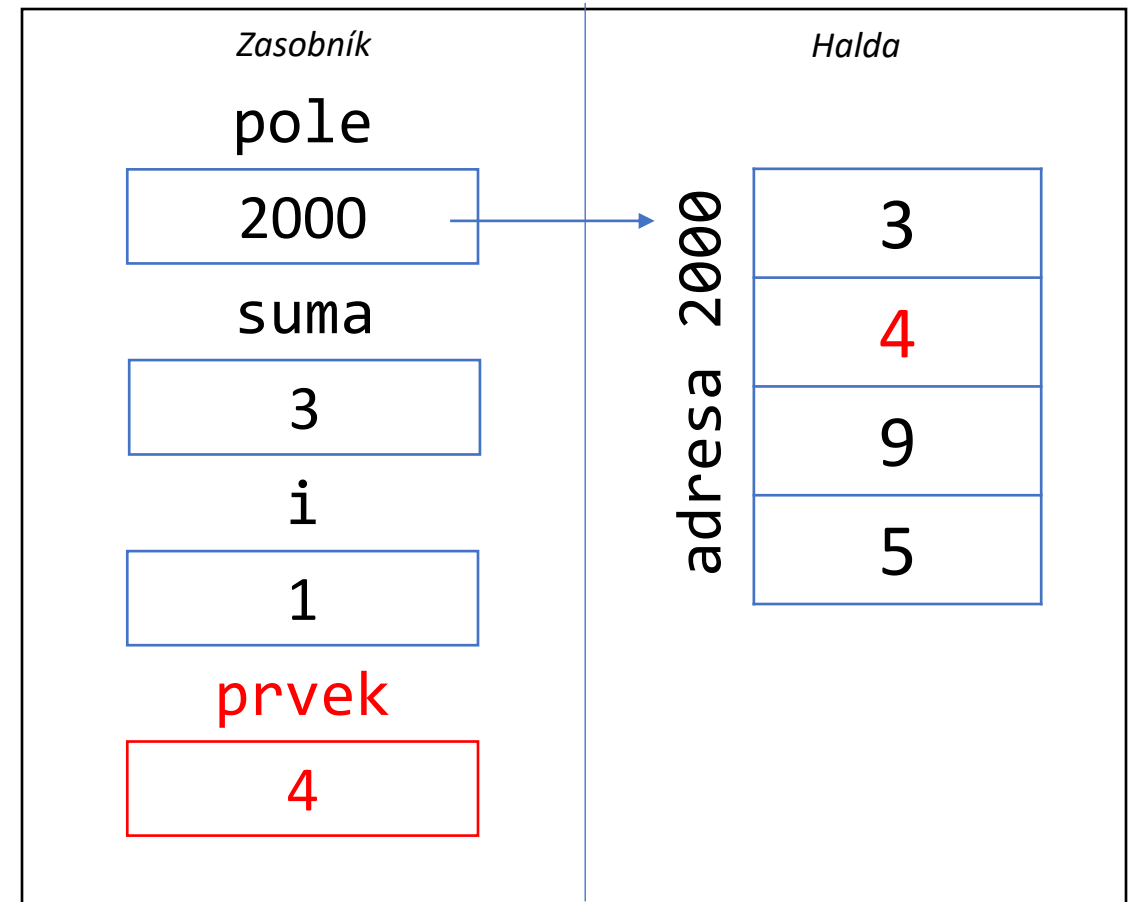
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

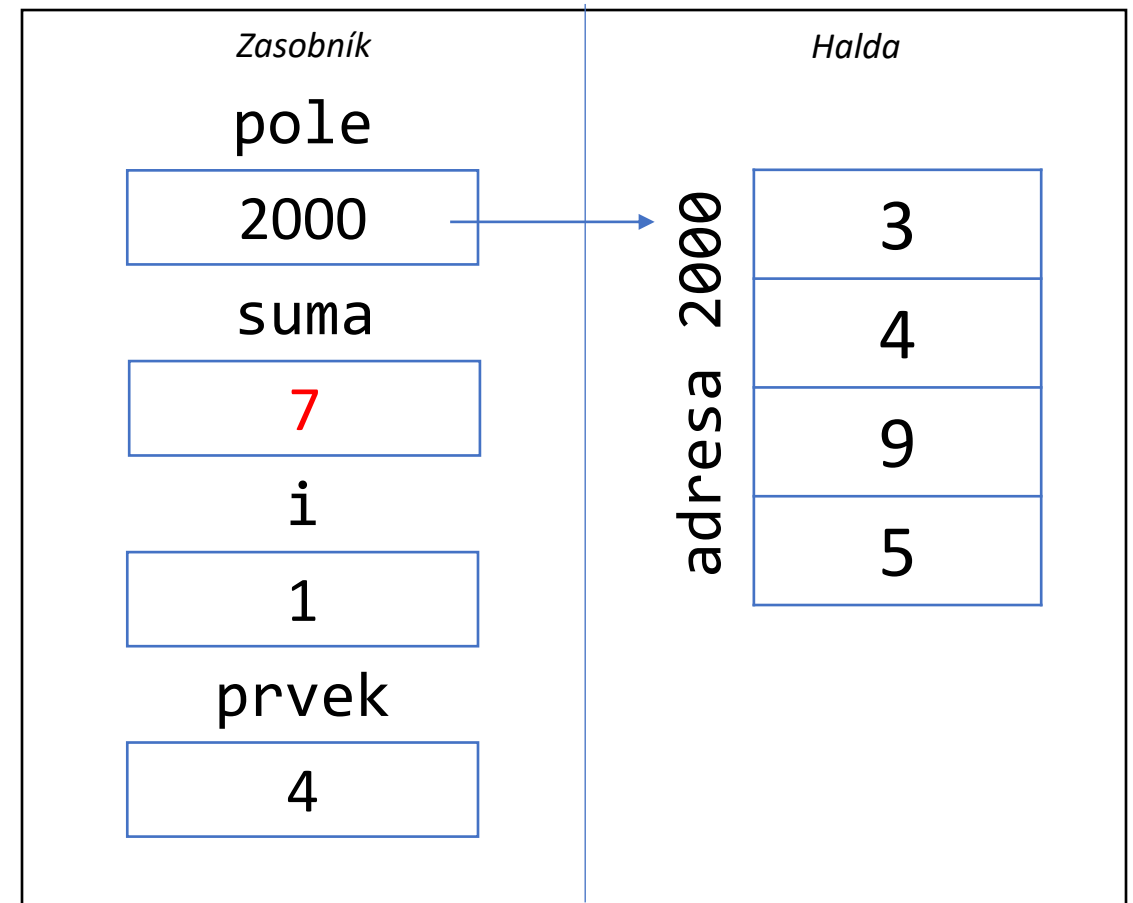
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

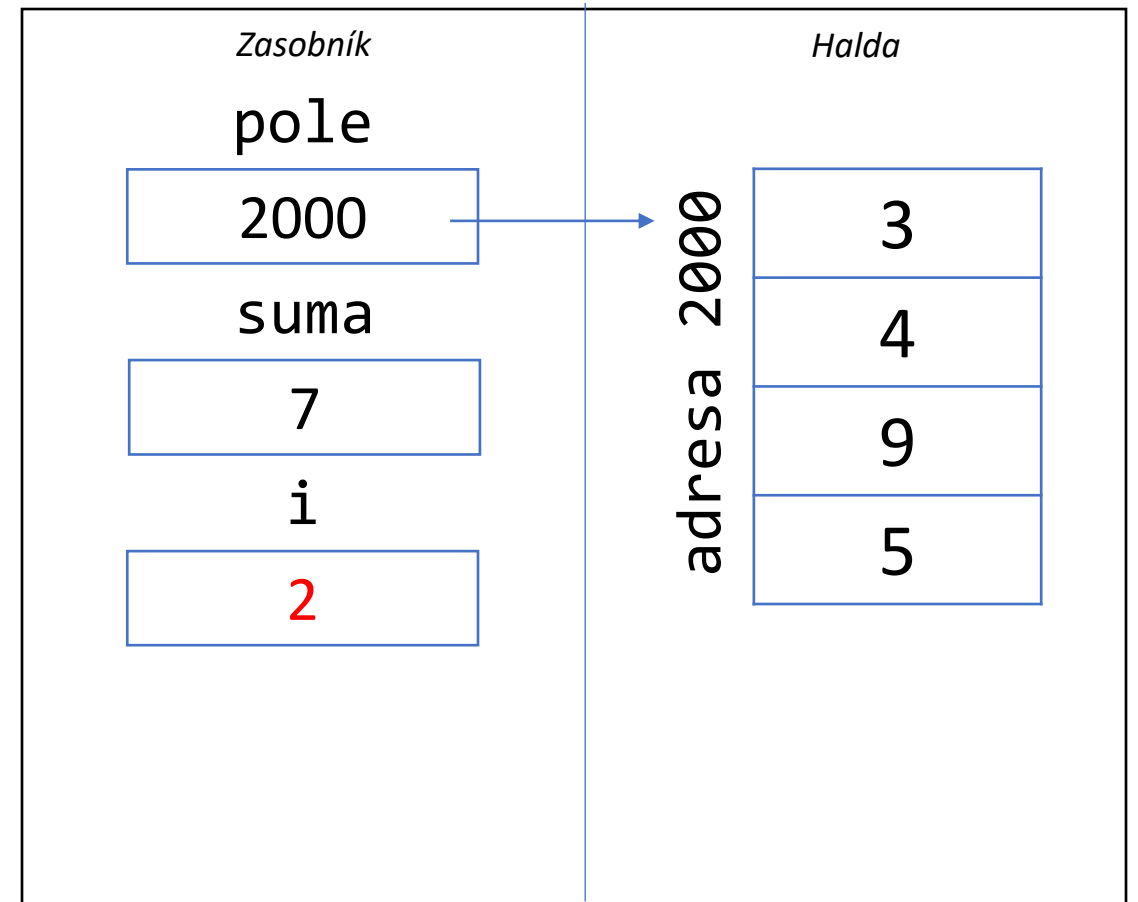
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

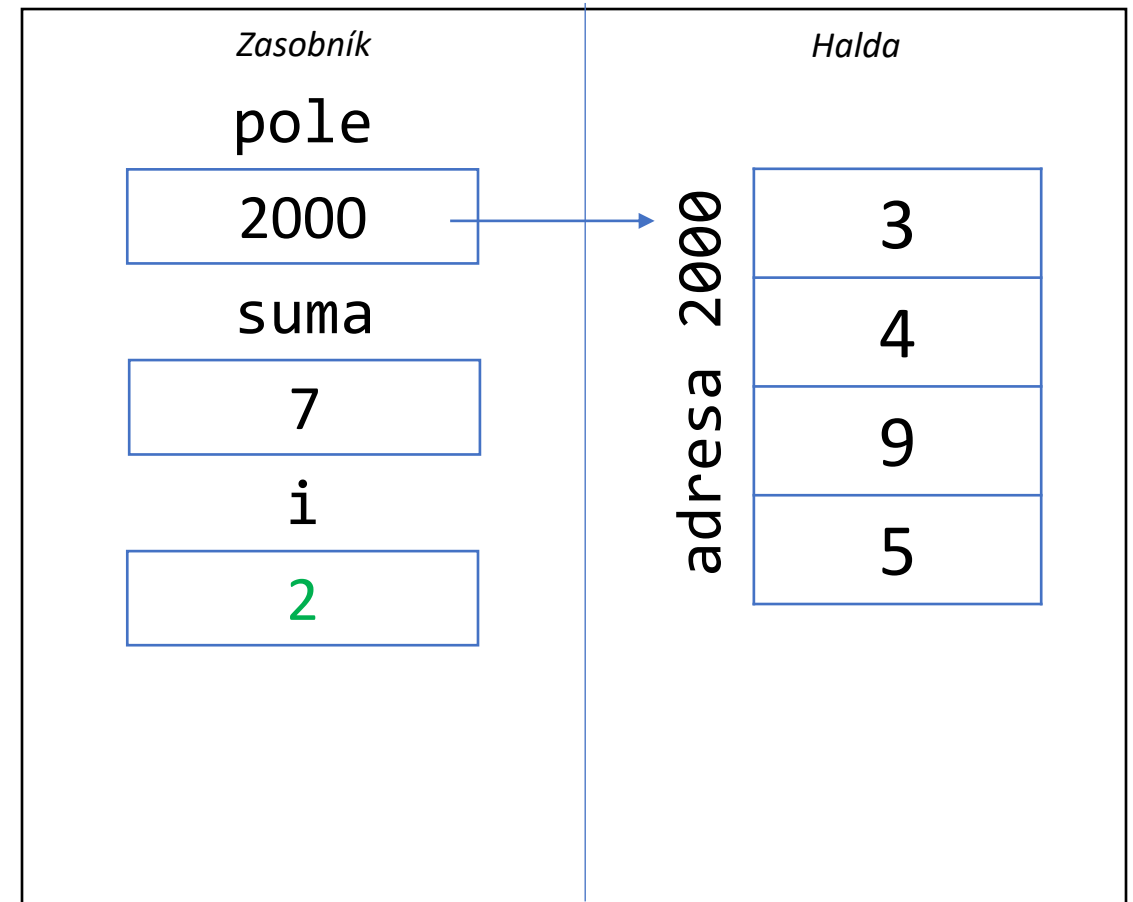
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

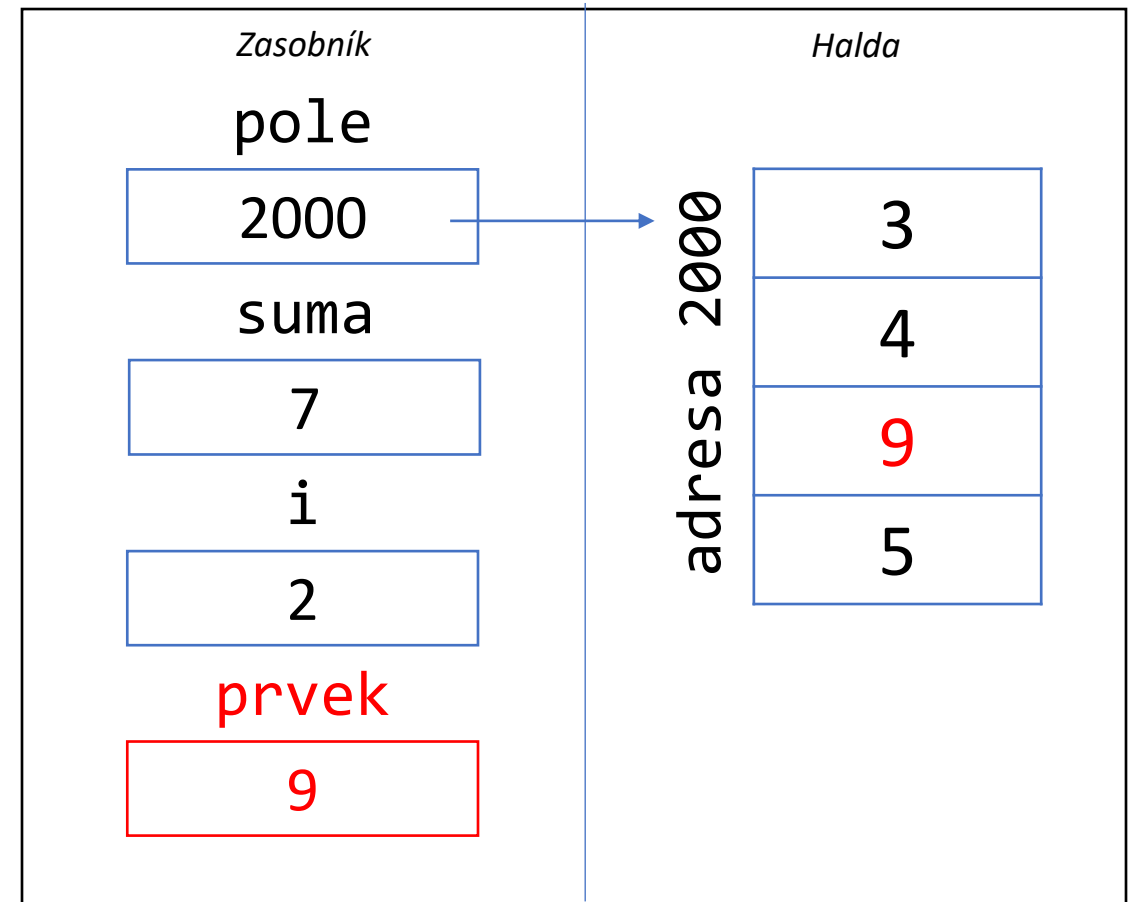
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

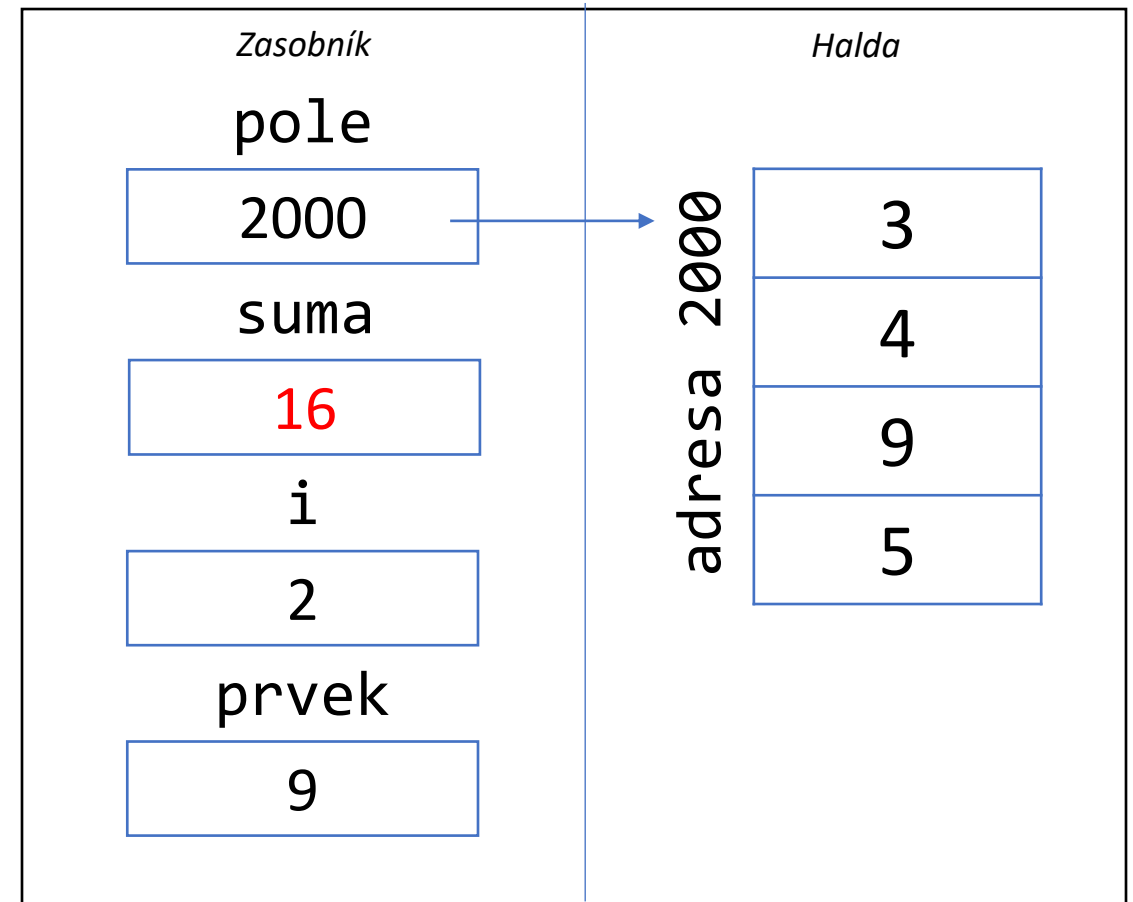
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

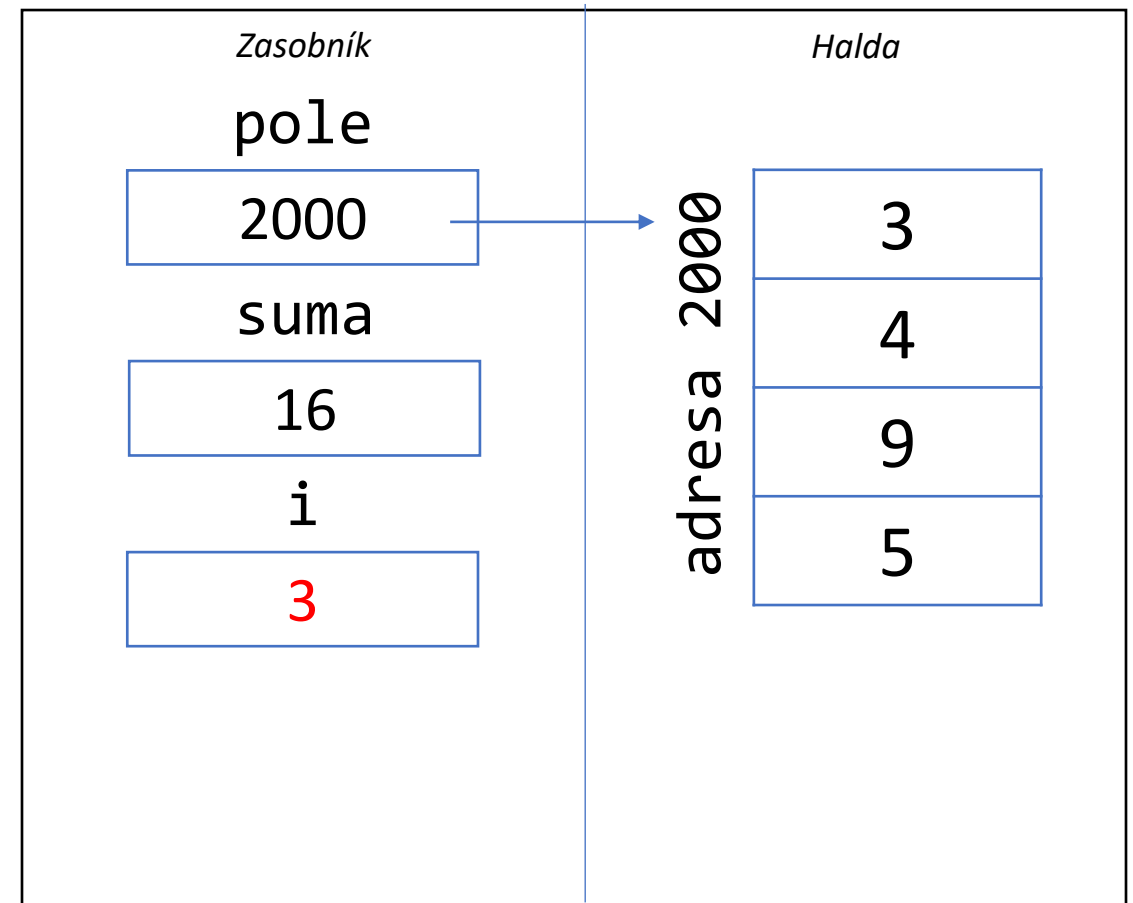
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

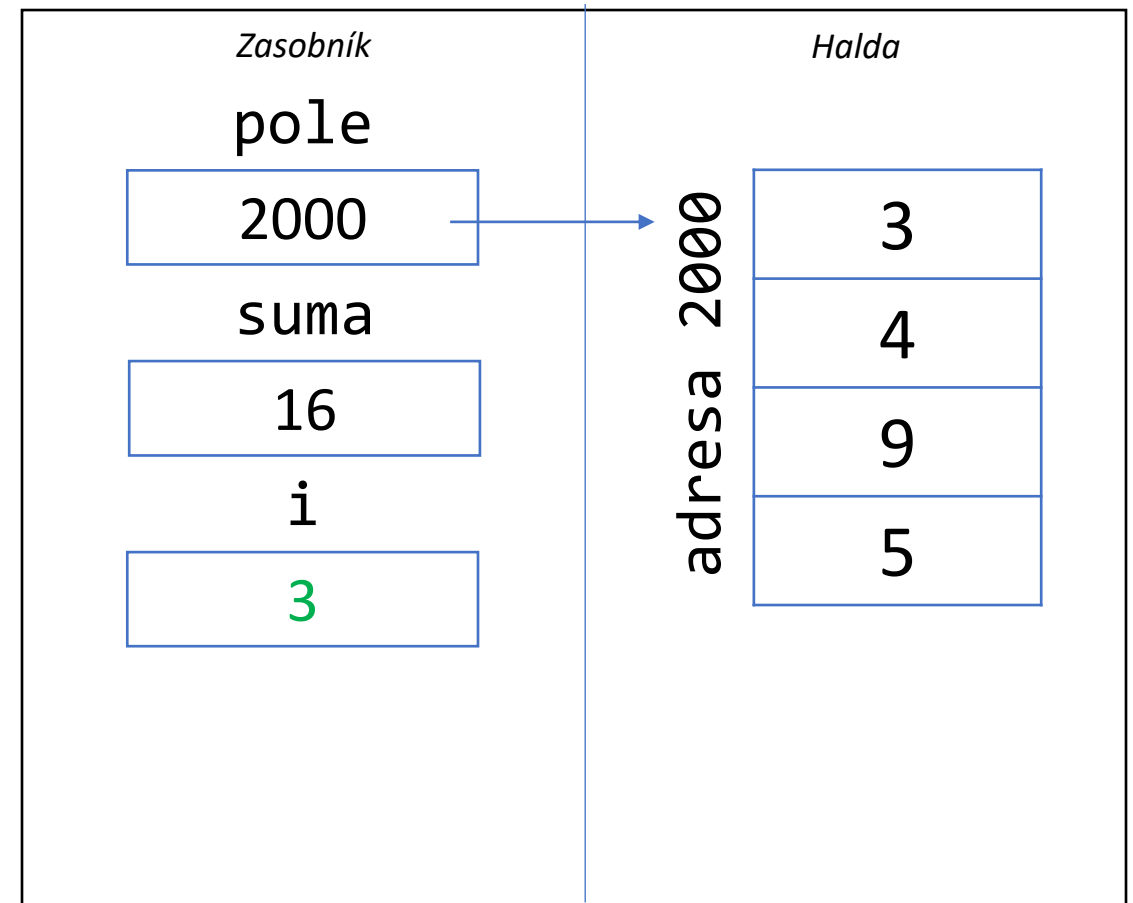
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

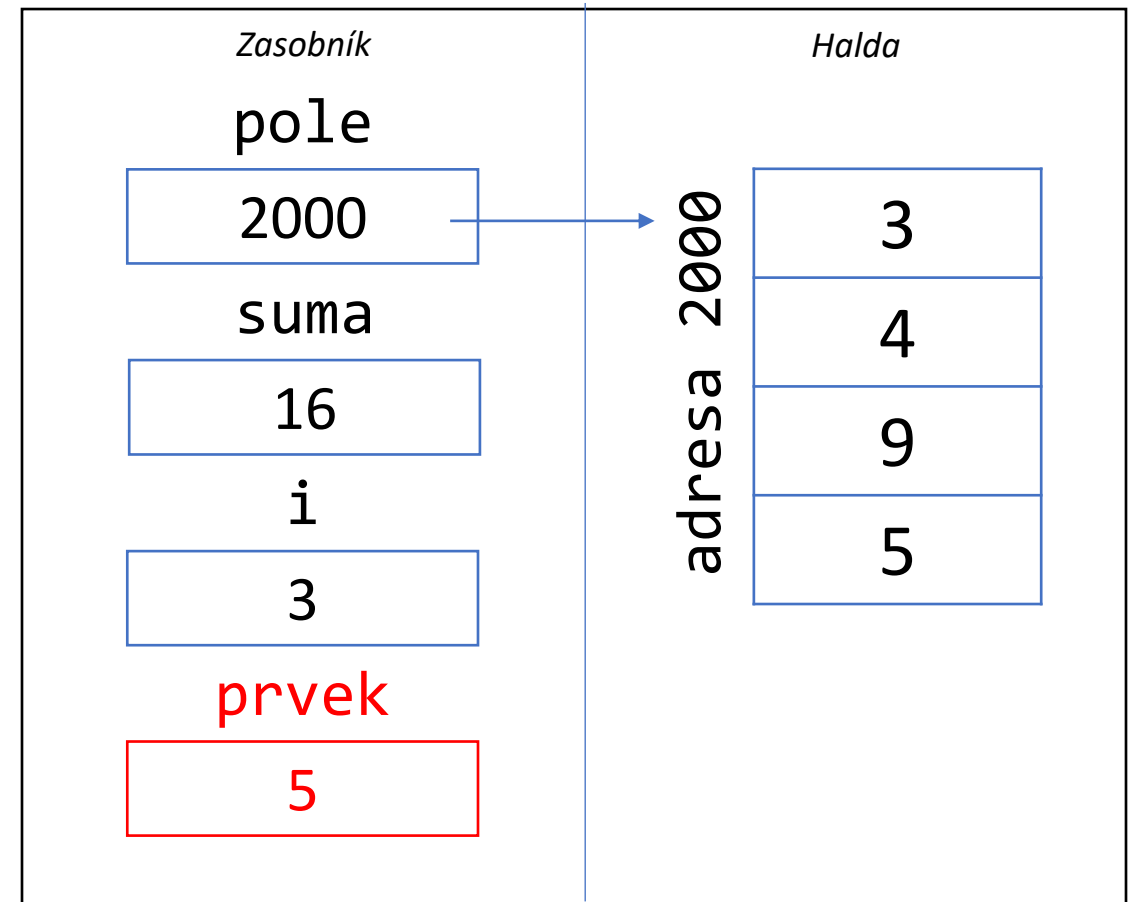
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

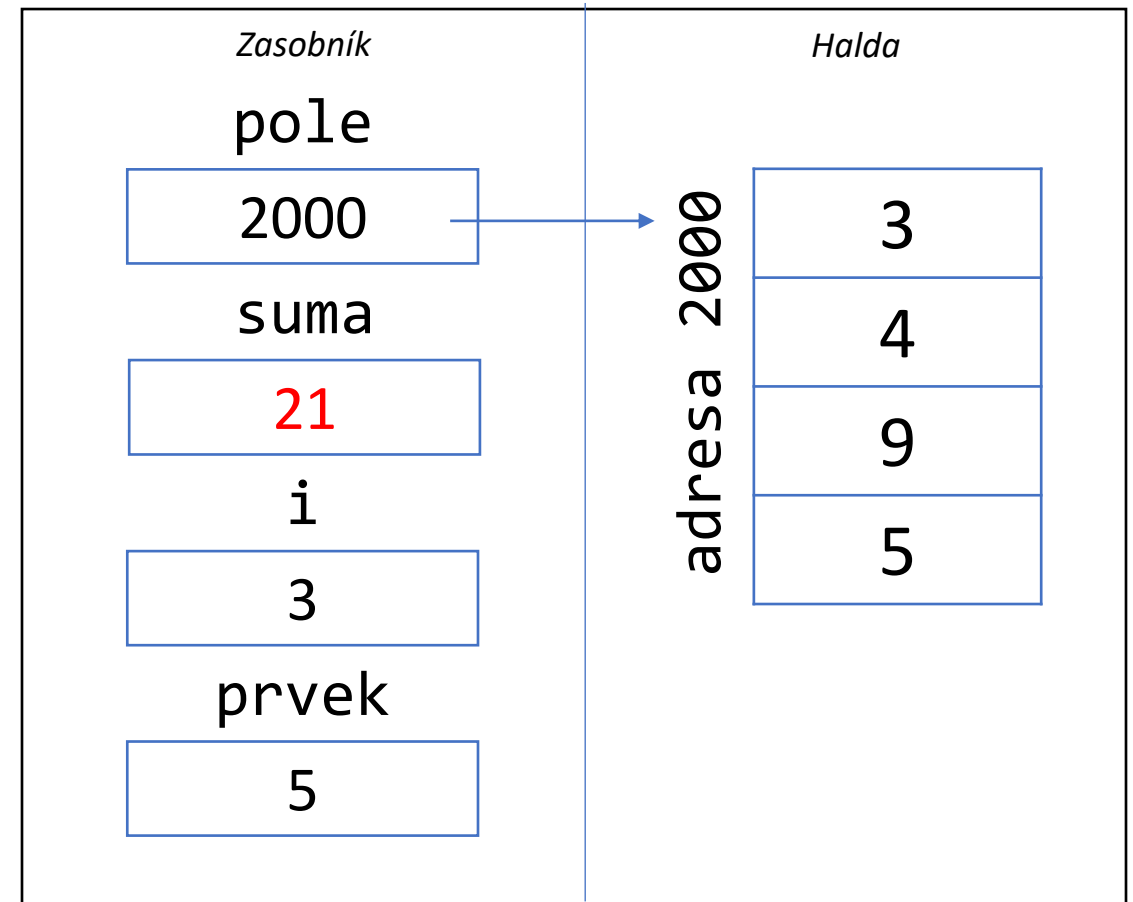
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

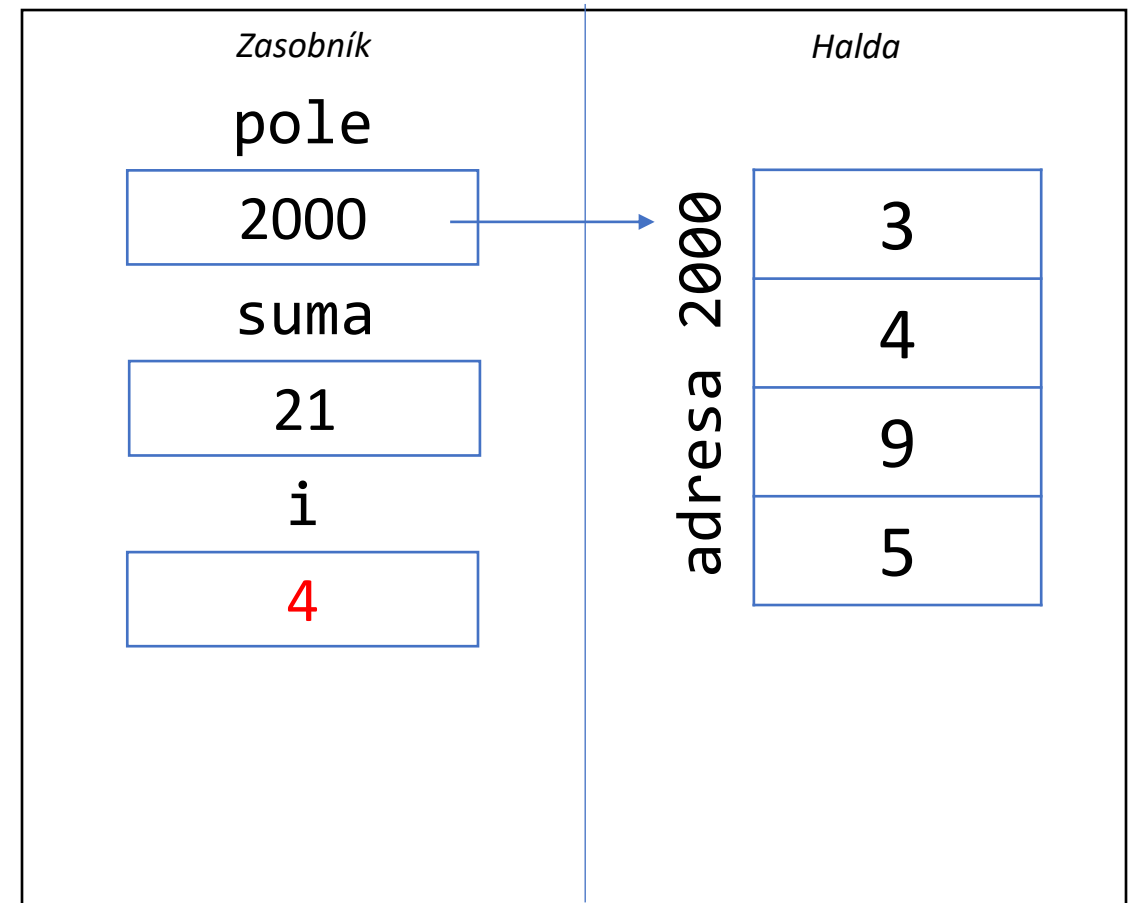
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

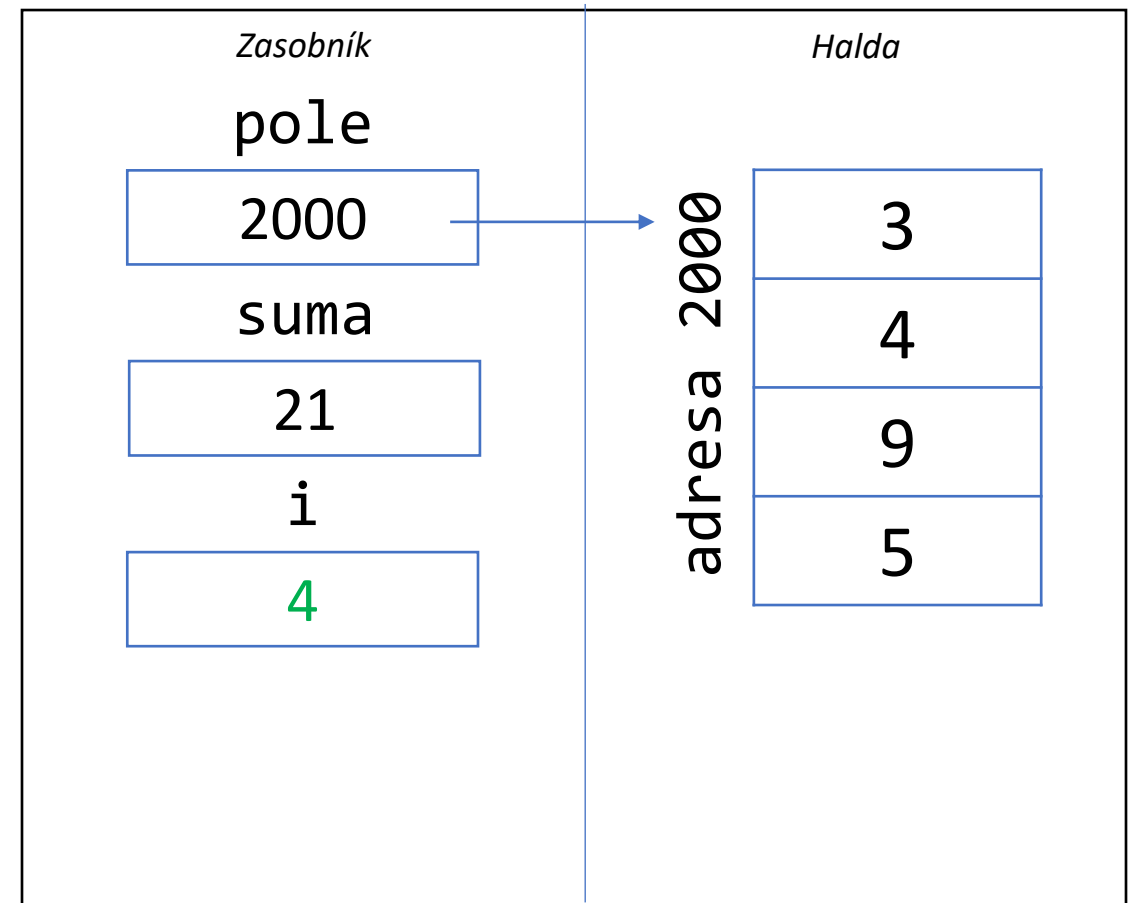
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}
```

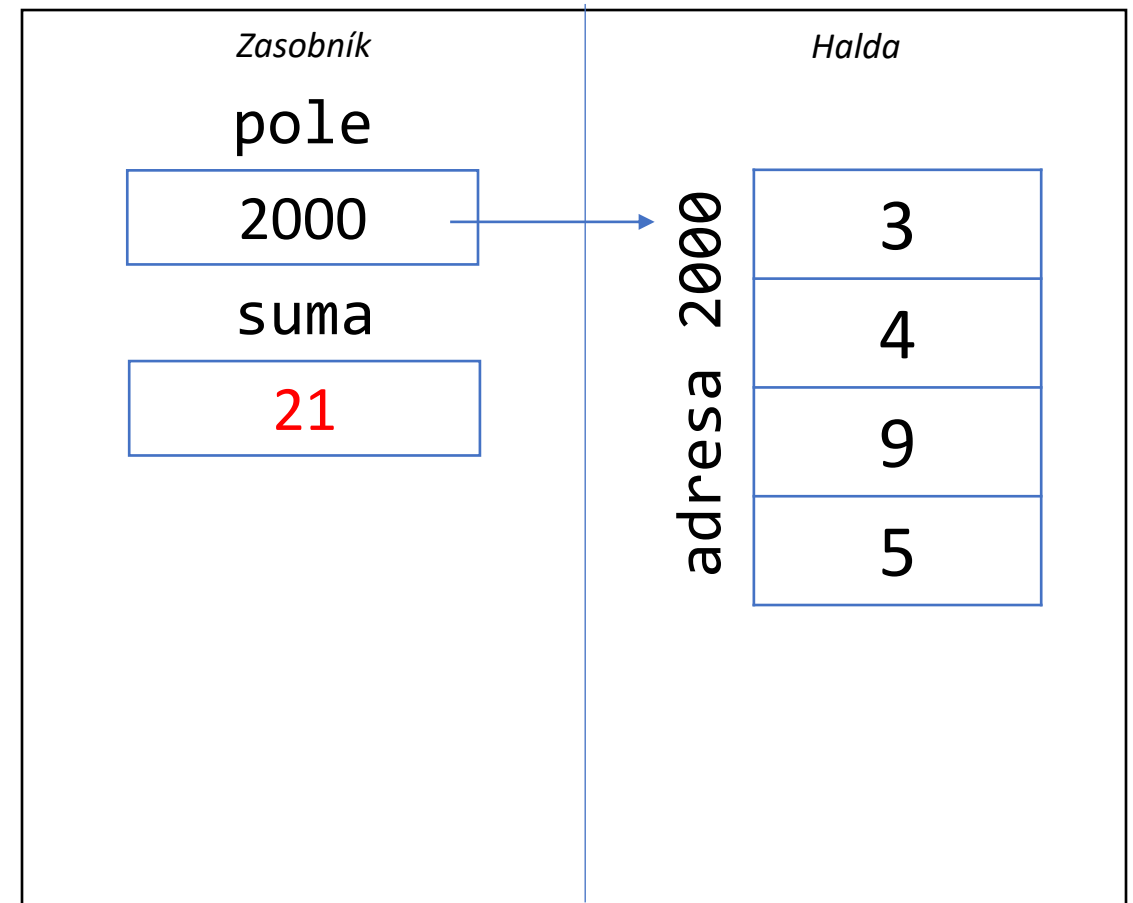
Paměť RAM



Suma prvků v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int suma = 0;  
  
for (int i = 0; i < pole.Length; i++)  
{  
    int prvek = pole[i];  
    suma += prvek;  
}  
  
Console.WriteLine(suma);
```

Paměť RAM



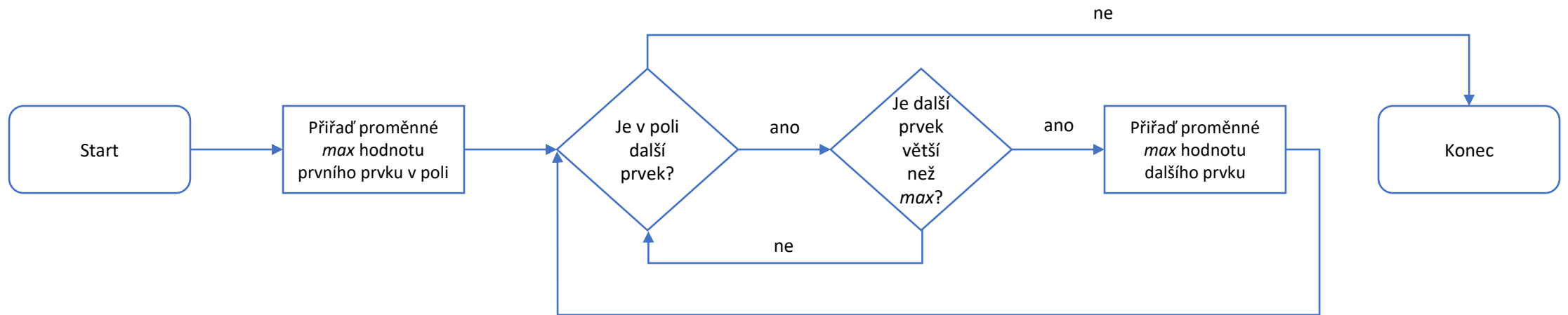
Poznámka k proměnné *prvek*

- V příkladu je znázorněné, že proměnná *suma* je definovaná znovu při každé iteraci cyklu *for* a bylo by vhodnější ji nadefinovat před začátkem tohoto cyklu.
- Ve skutečnosti **kompilátor** tento **kód zoptimalizuje** a **není nutné** tuto **proměnnou definovat před cyklem *for*** a dáváme přednost lepší čitelnosti kódu.

Hledání maximální hodnoty v poli

- Algoritmus hledání maximální hodnoty v poli je založen na tom, že si nadefinujeme **proměnou *max*** do které budeme průběžně **ukládat největší hodnotu** prvku v poli.
- Na začátku může například jako zatím největší hodnotu prvku nastavit **hodnotu prvního prvku**.
- Poté postupně procházíme další prvky v poli **a porovnáváme ji se zatím největší nalezenou hodnotou. Pokud je další prvek větší, tak změníme** zatím **maximální** hodnotu uloženou v proměnné **max**.

Hledání maximální hodnoty v poli



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };

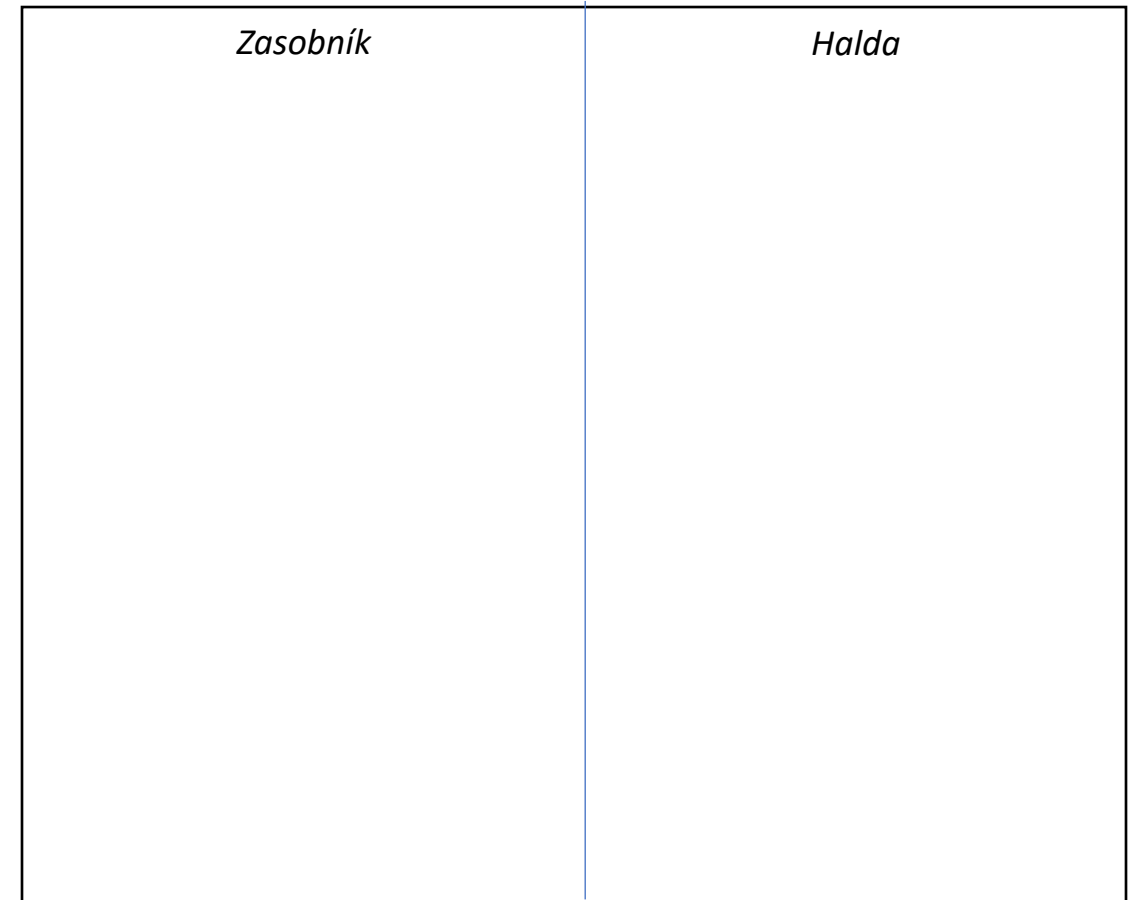
int max = pole[0];

for (int i = 1; i < pole.Length; i++)
{
    int dalsi = pole[i];

    if (dalsi > max)
    {
        max = dalsi;
    }
}

Console.WriteLine(max);
```

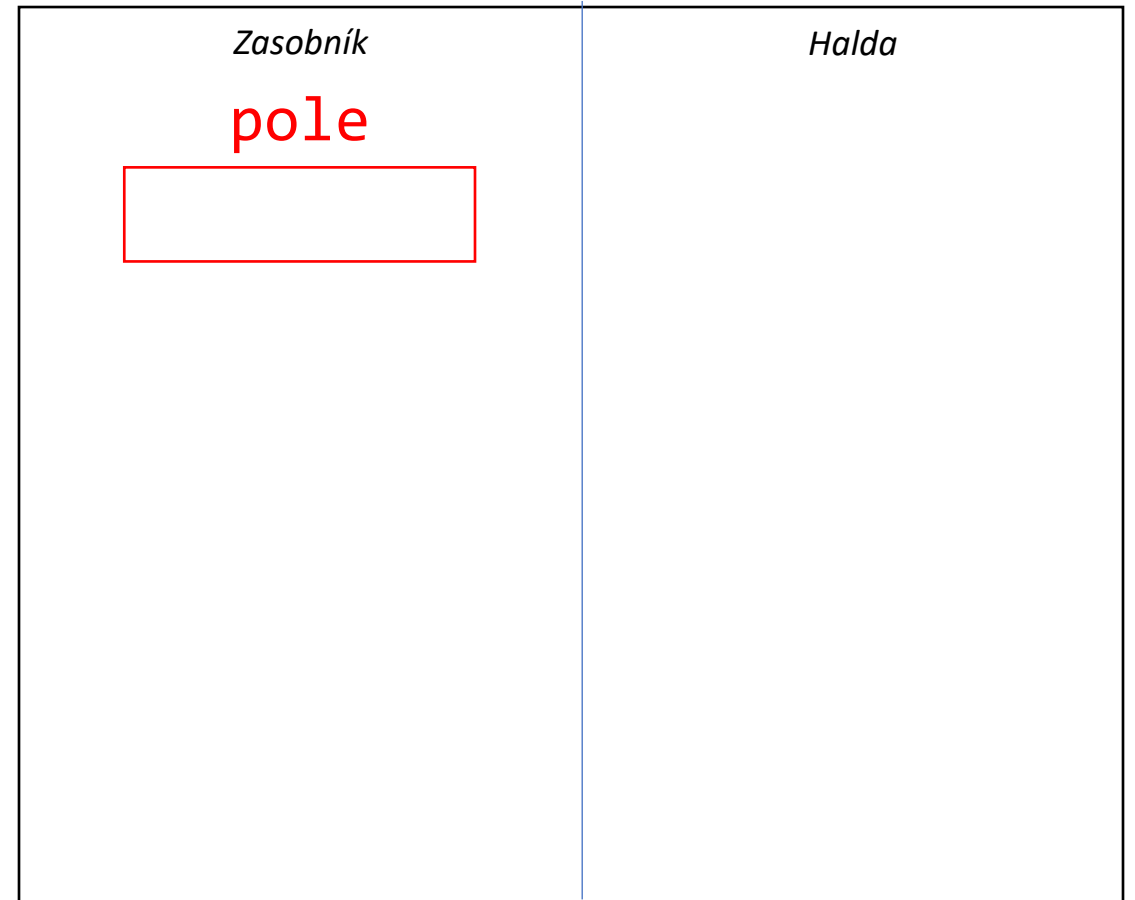
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

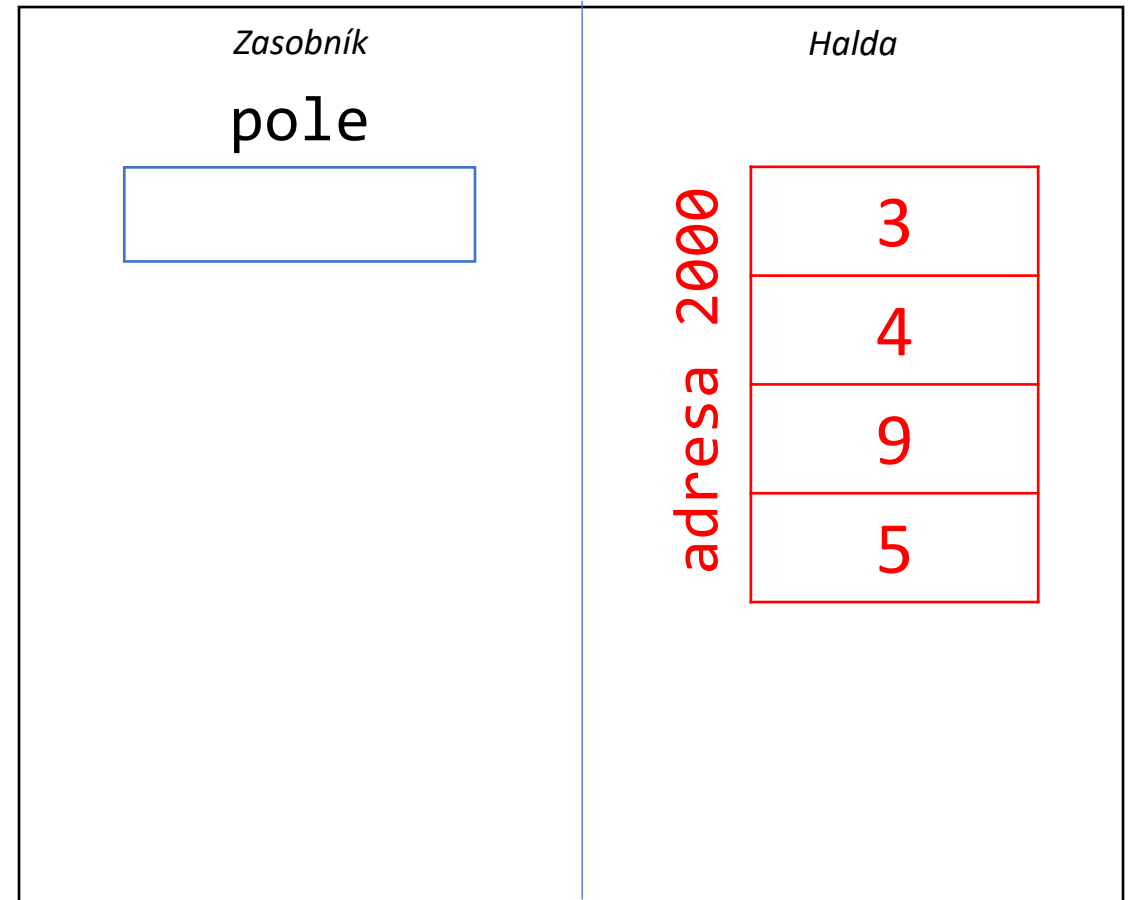
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

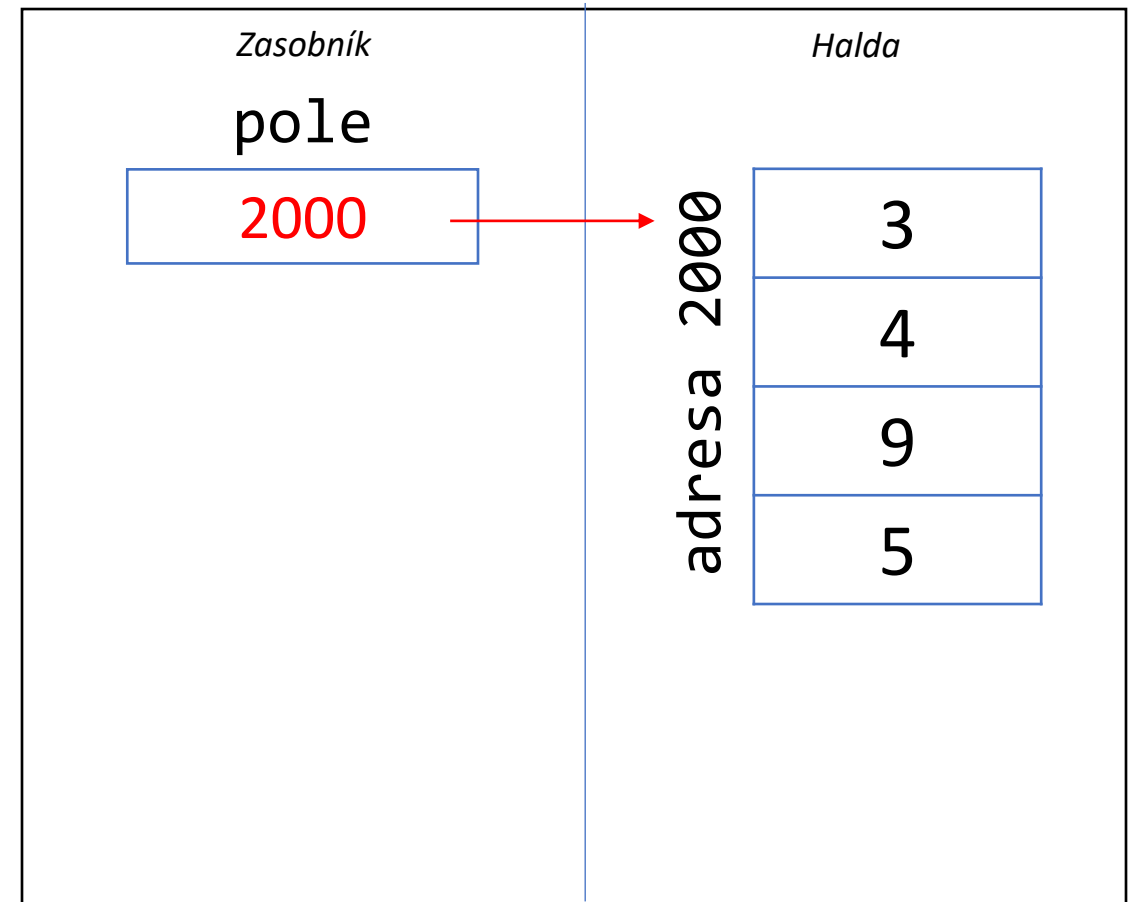
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };
```

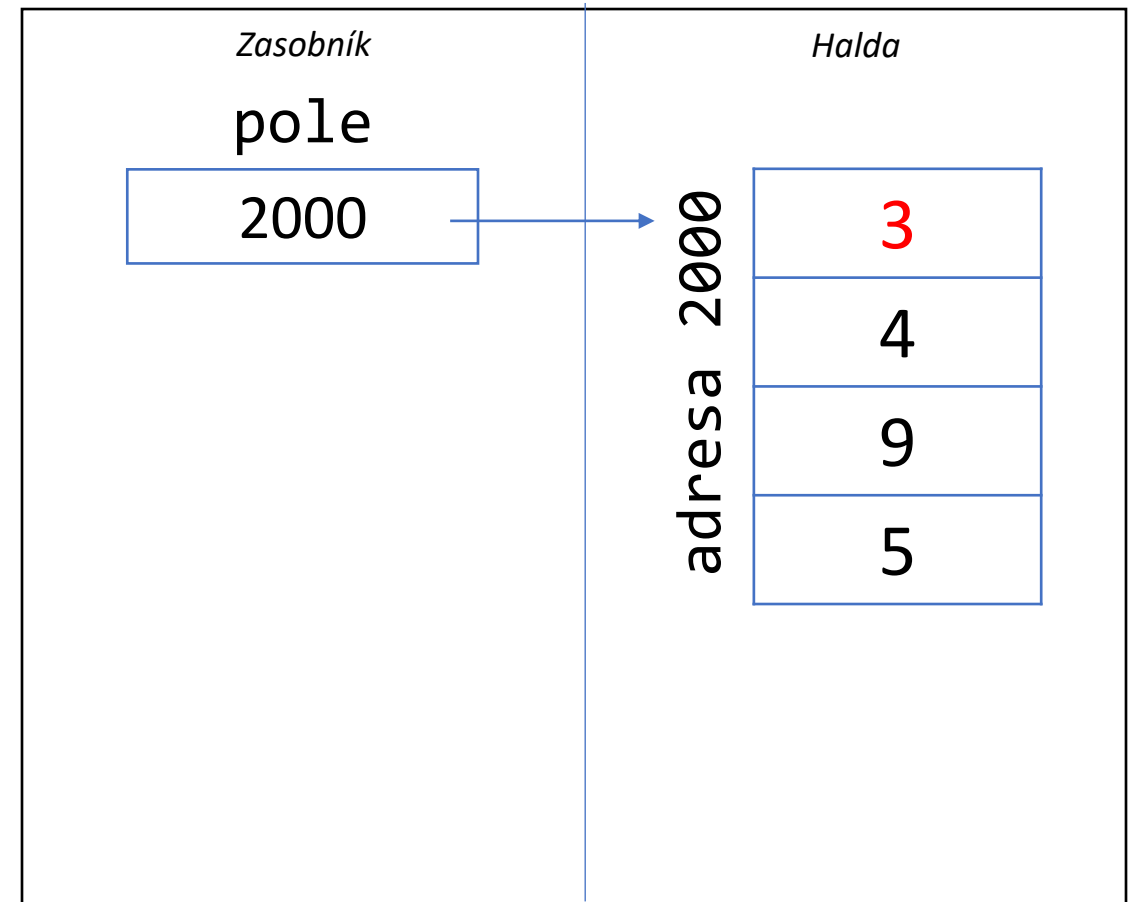
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
int max = pole[0];
```

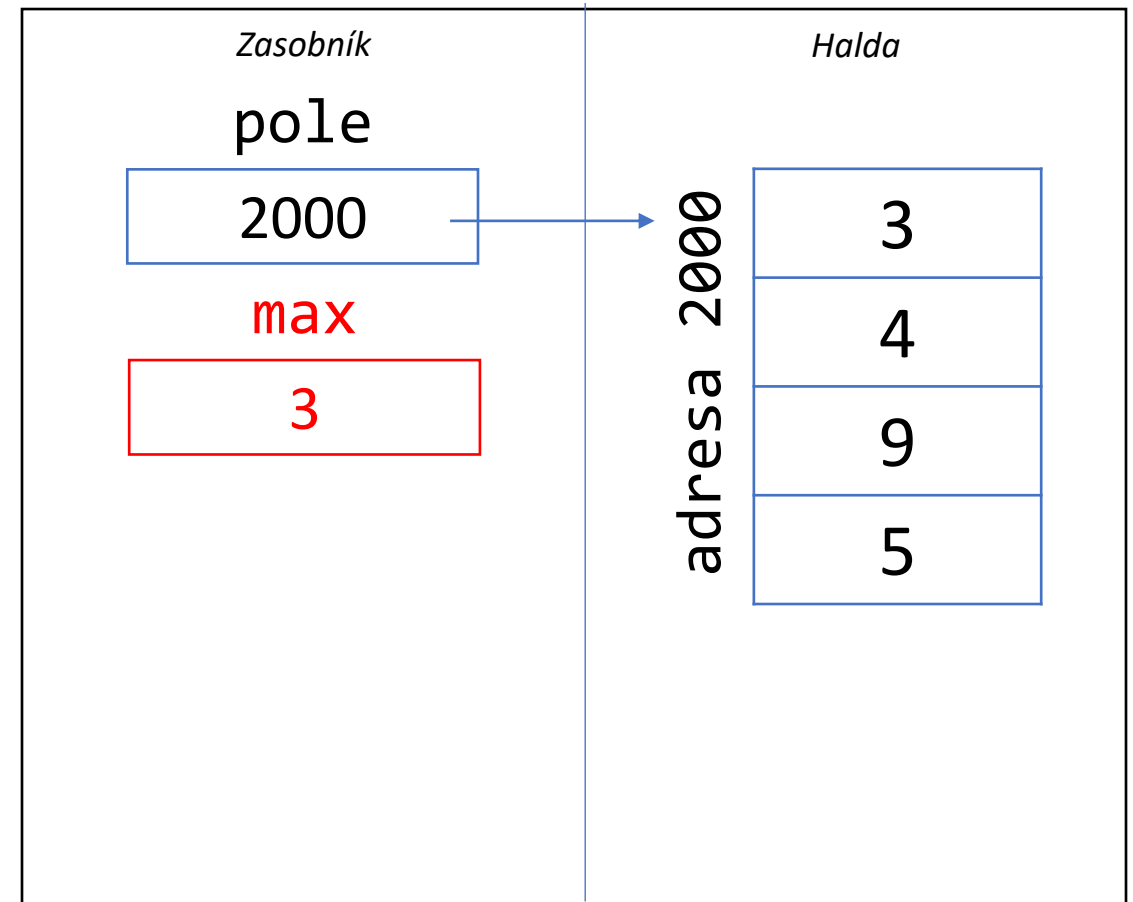
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
int max = pole[0];
```

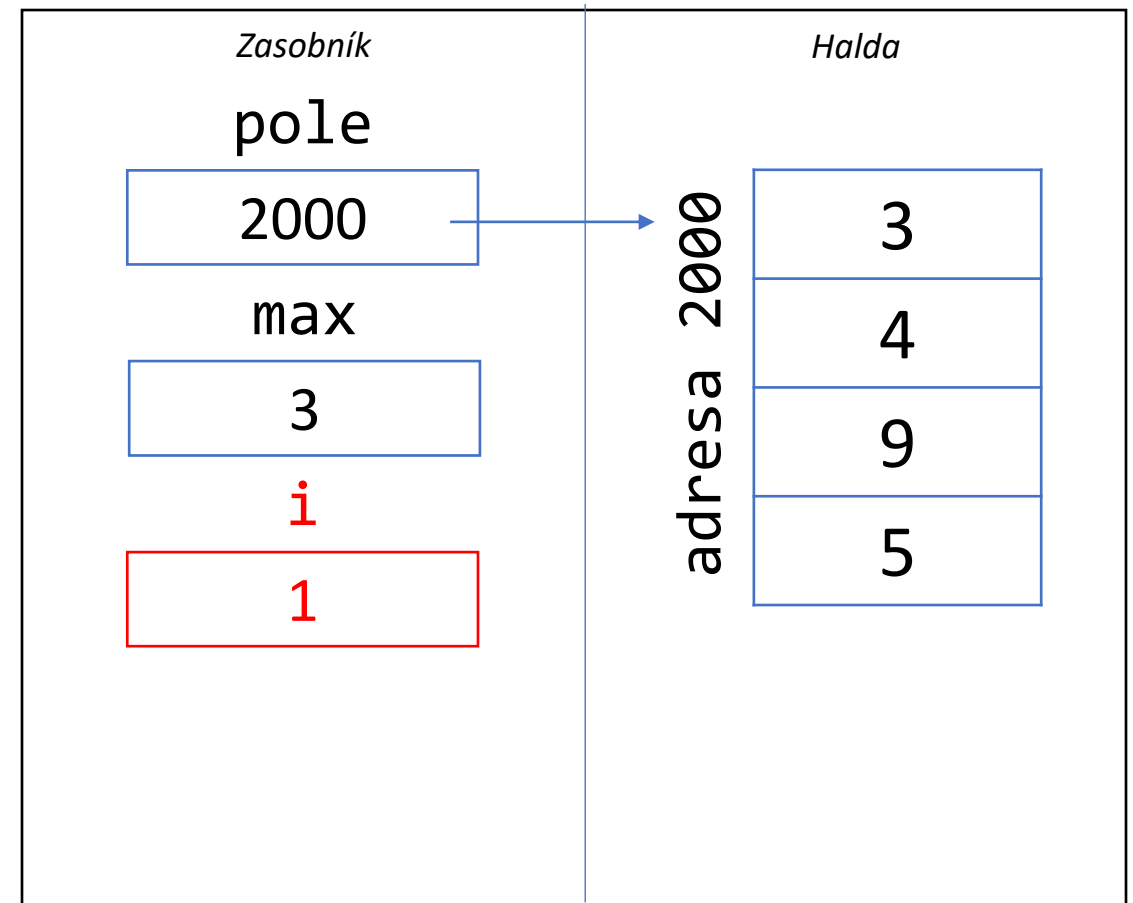
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
  
}
```

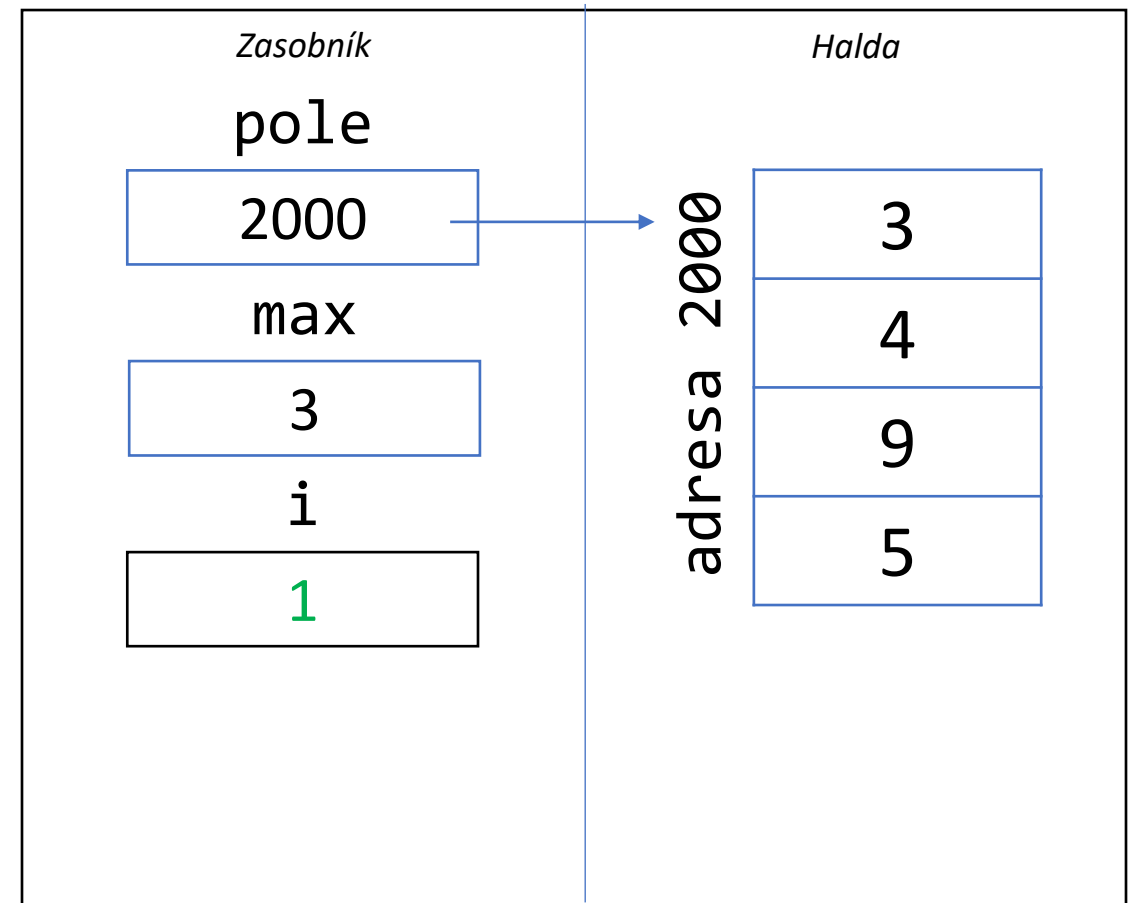
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
for (int i = 1; i < pole.Length; i++)  
{  
  
}
```

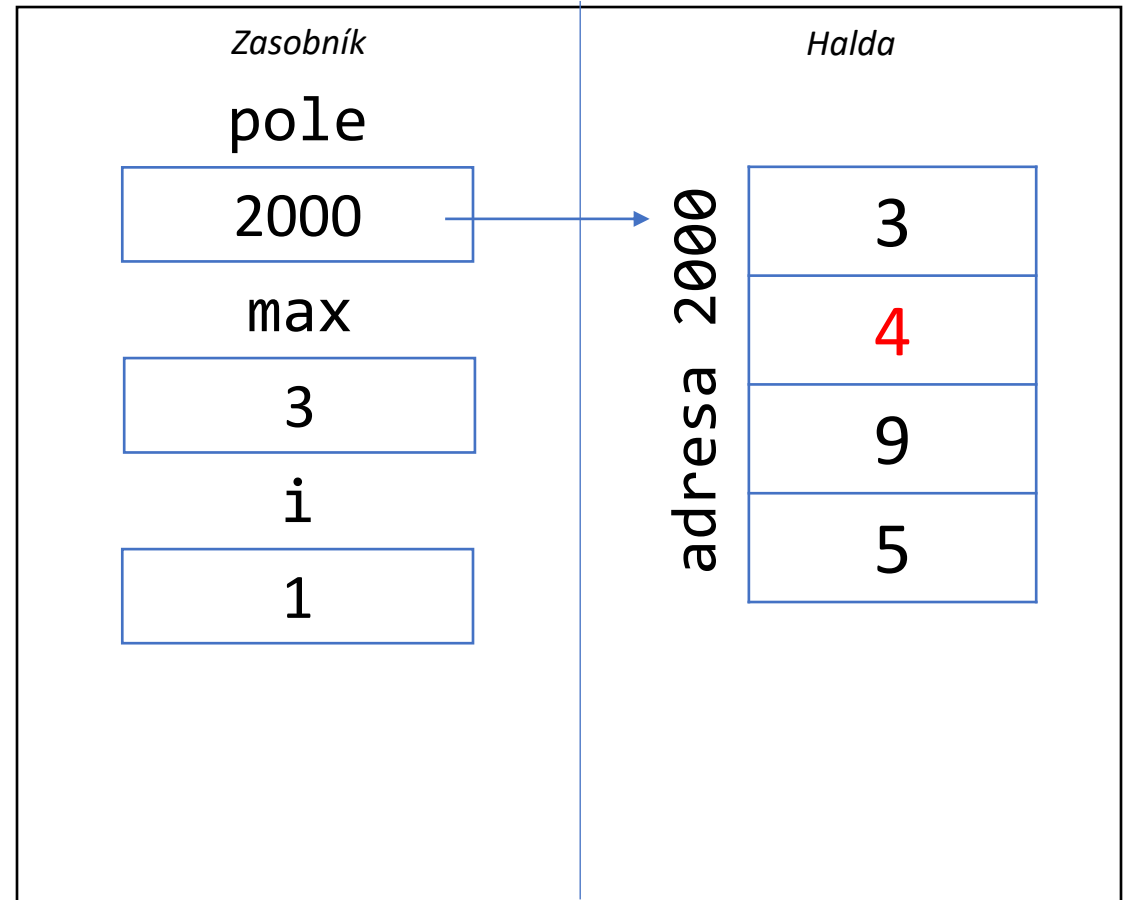
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
}
```

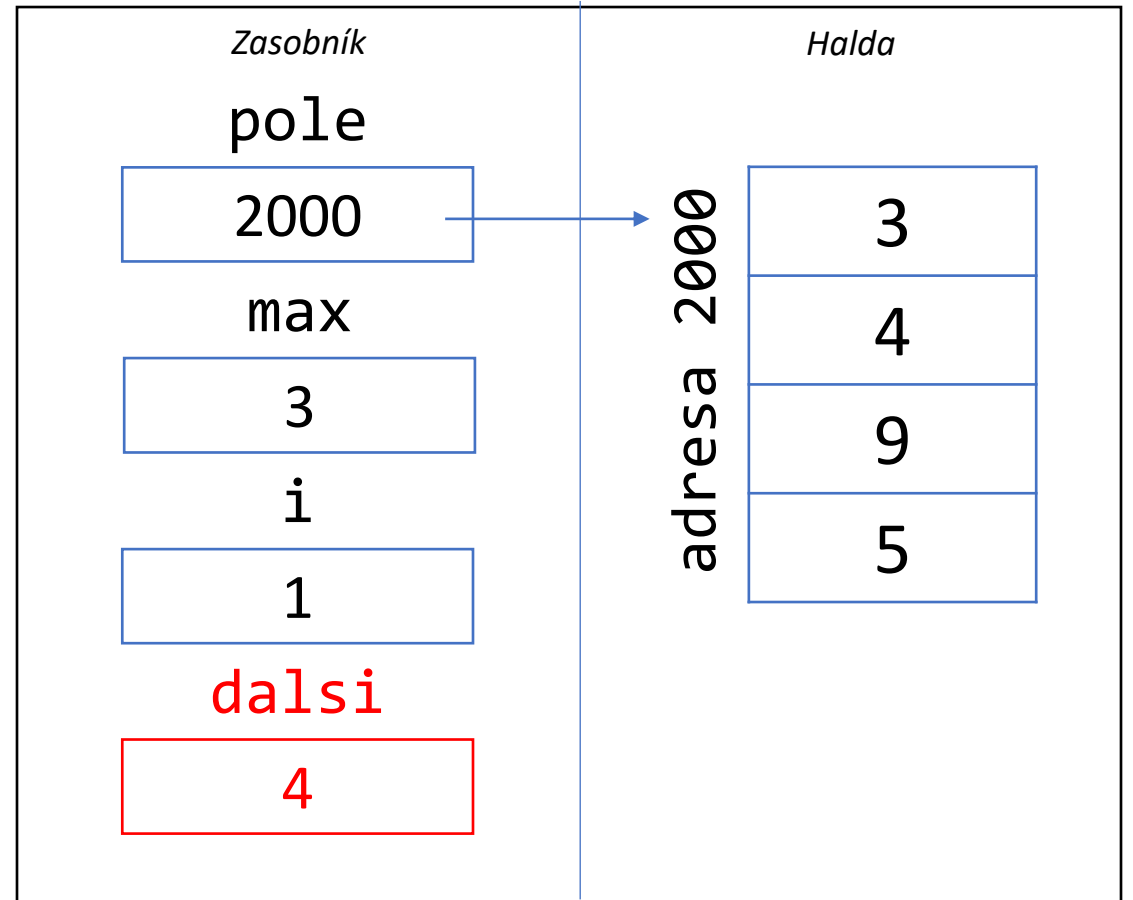
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
}
```

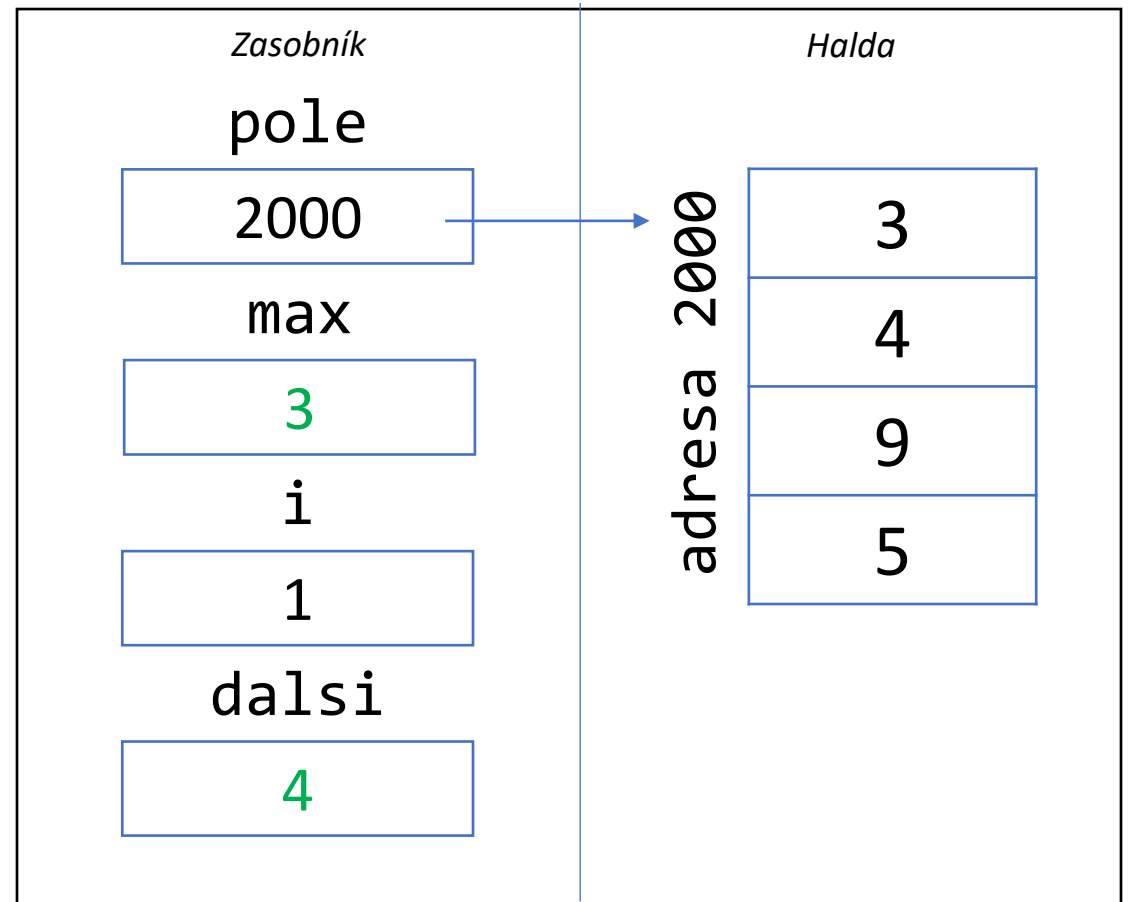
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
    if (dalsi > max)  
    {  
    }  
}
```

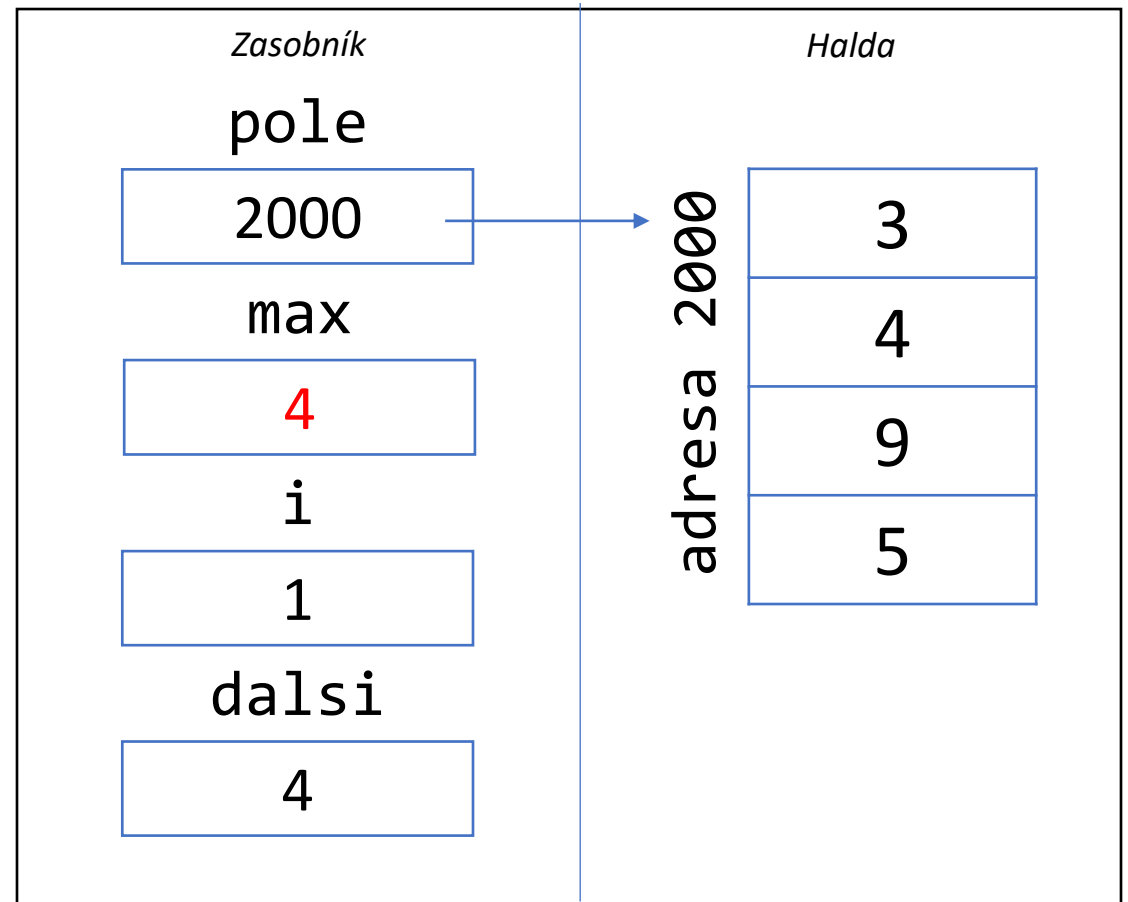
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

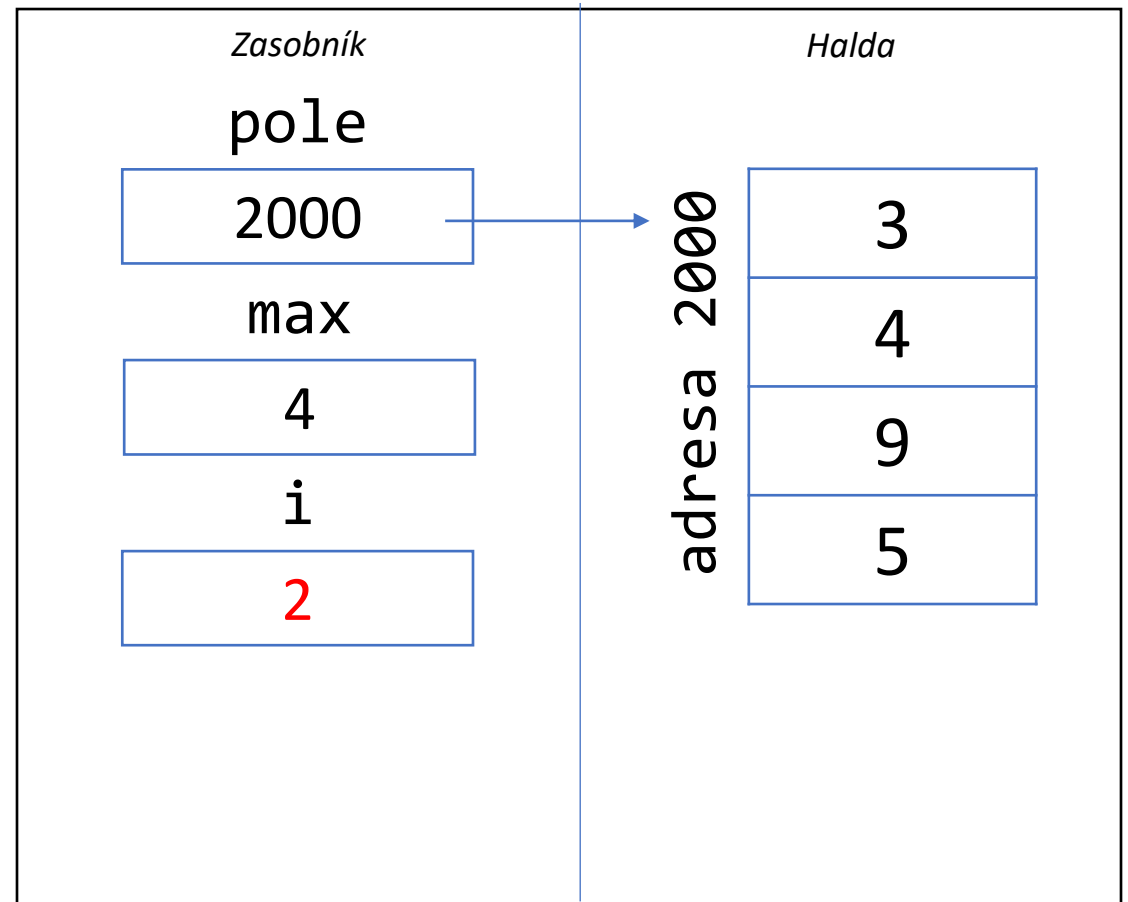
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

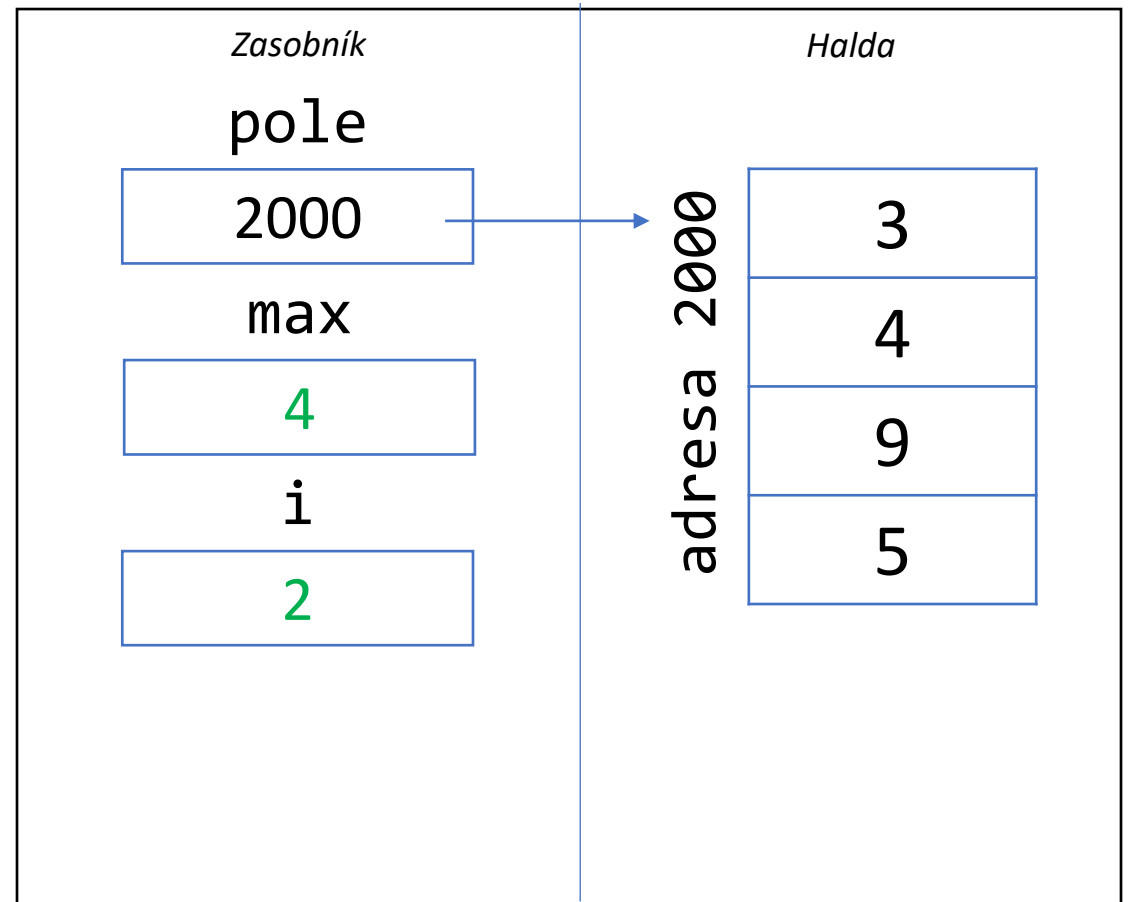
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

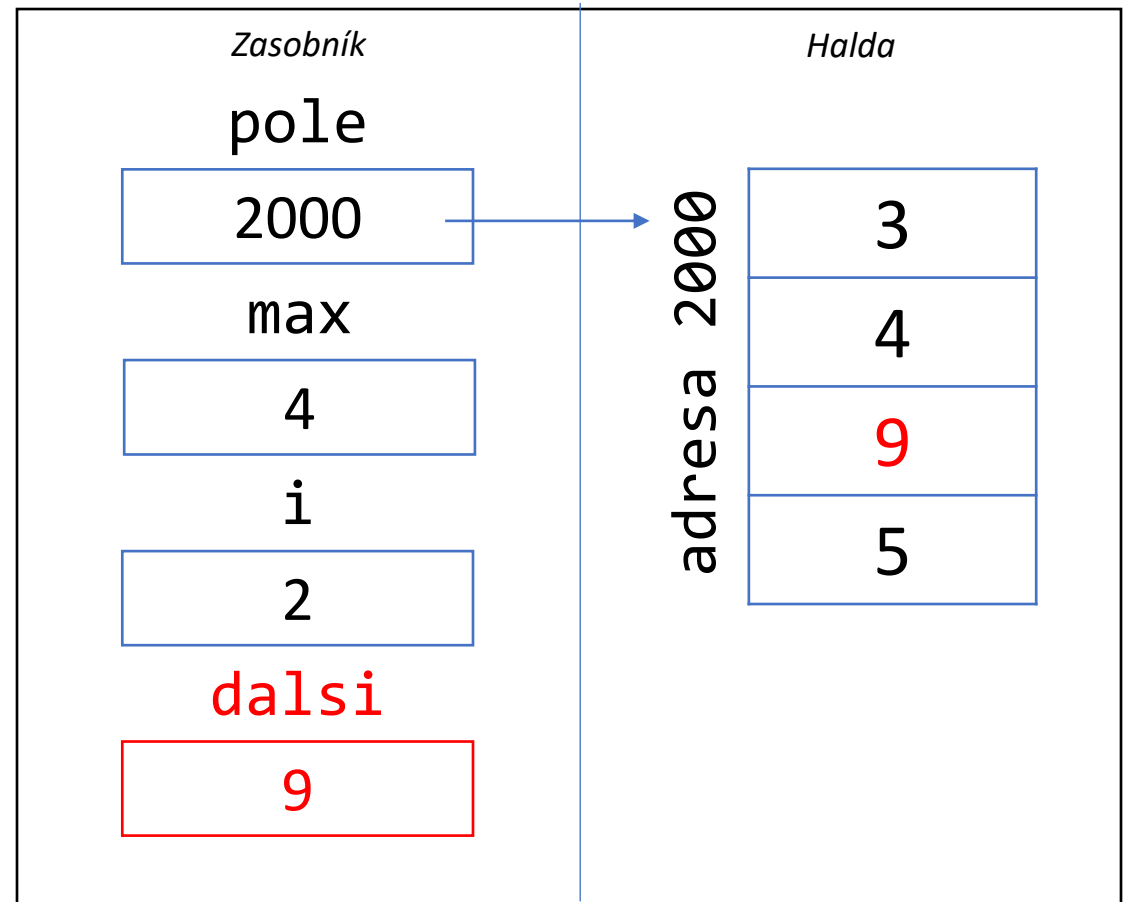
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

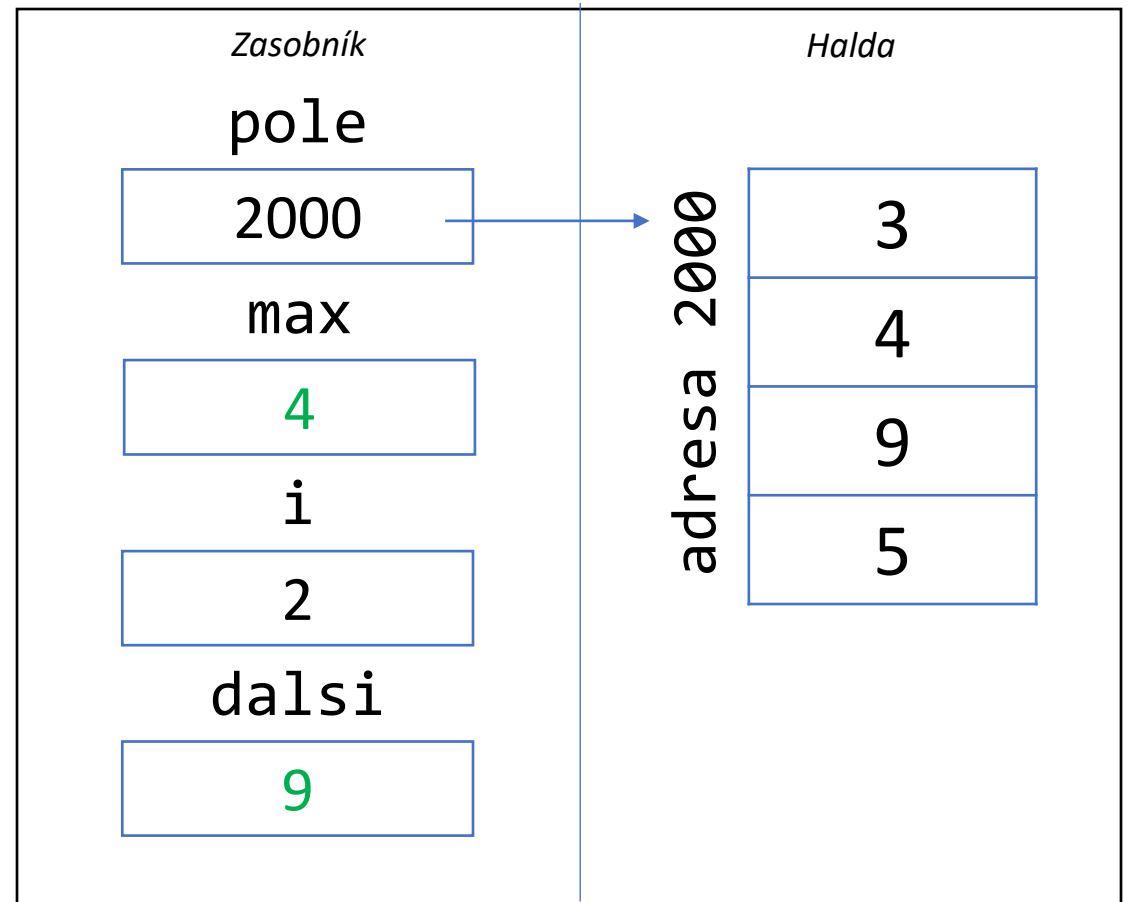
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

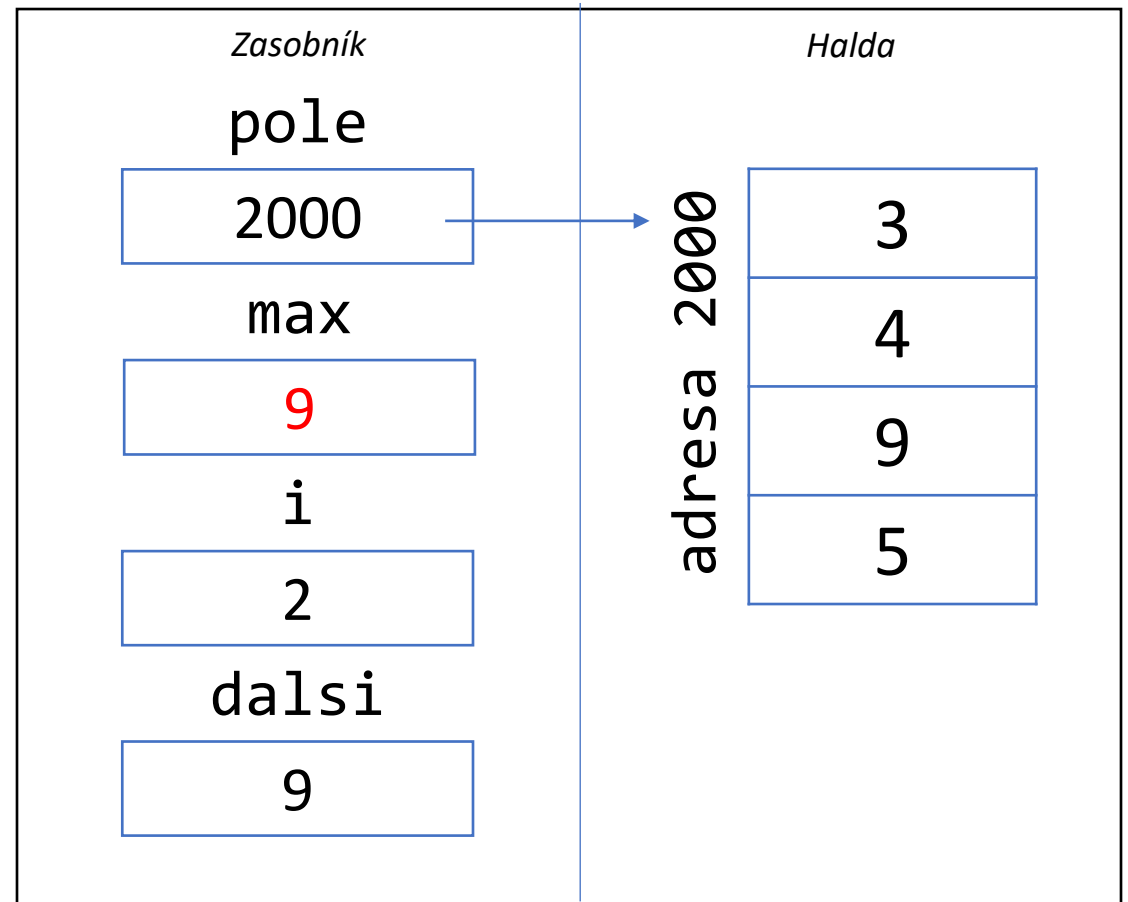
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

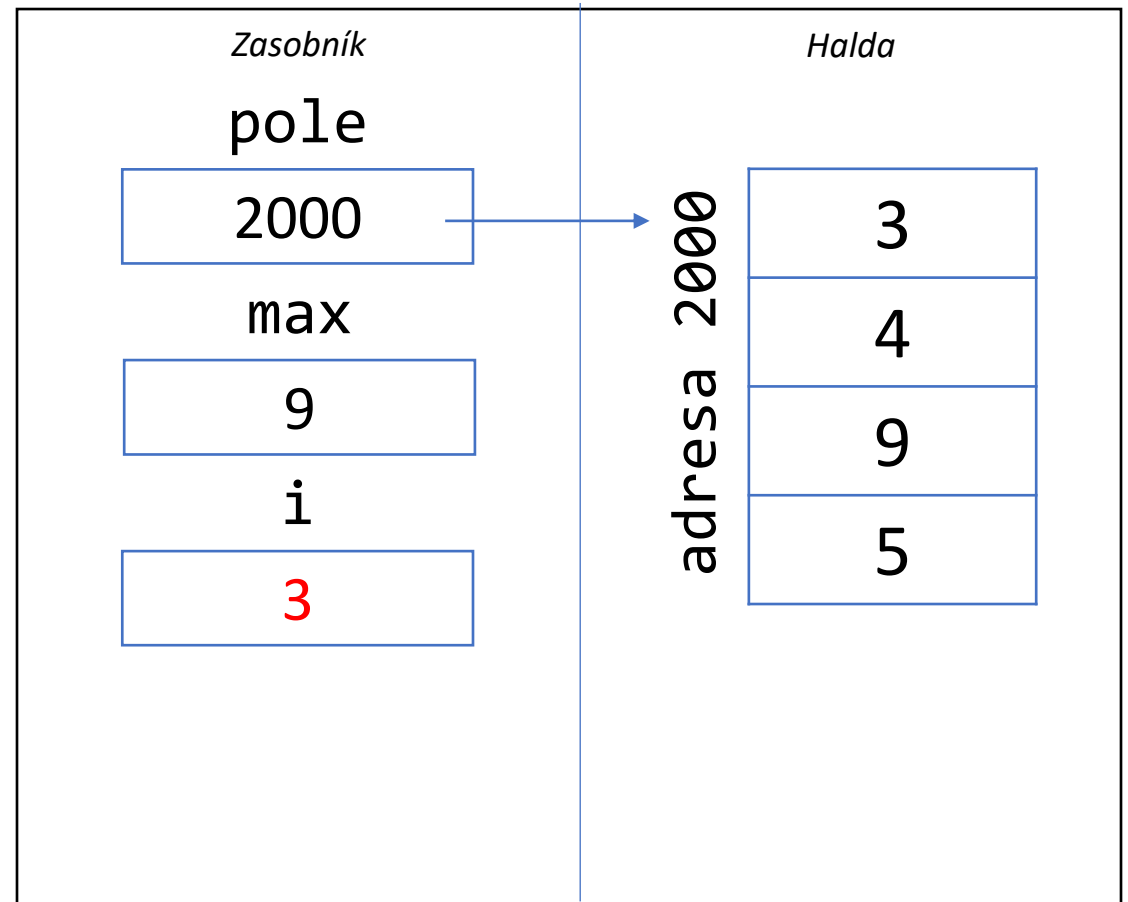
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

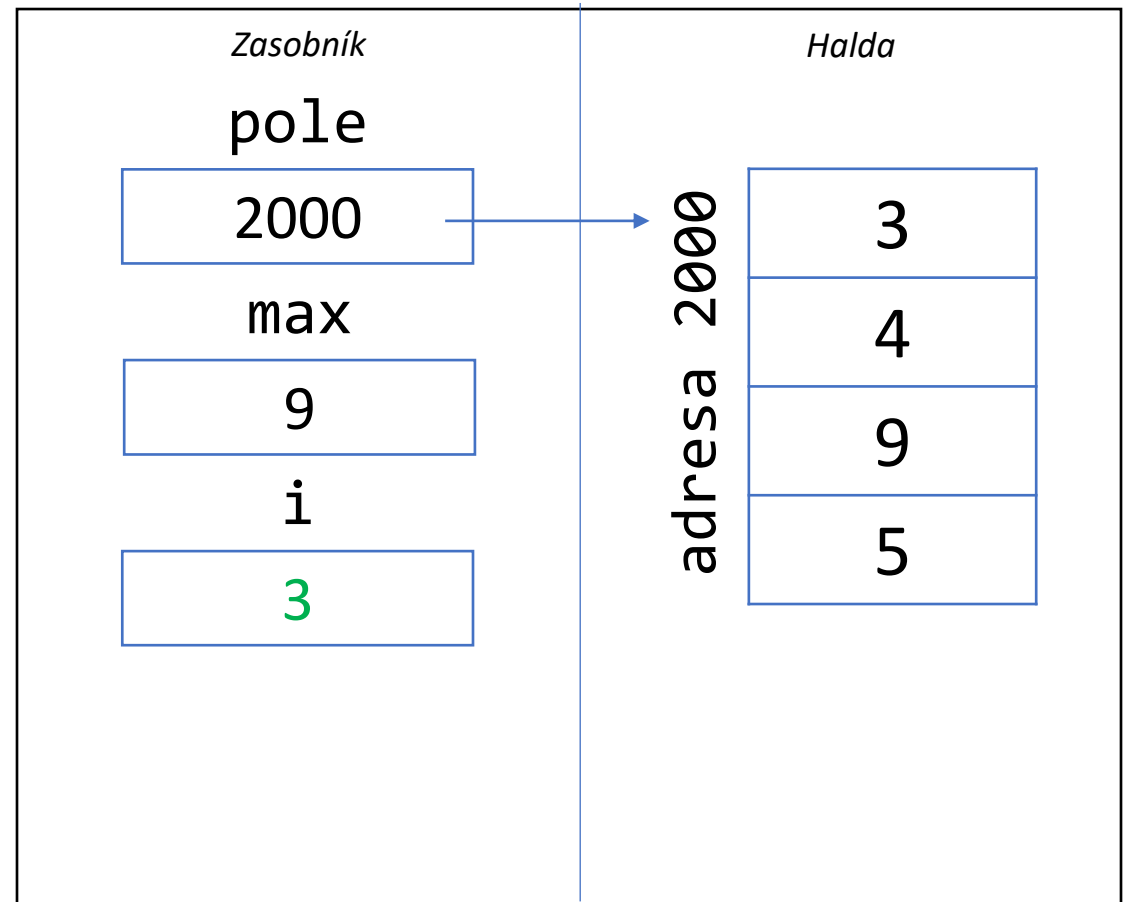
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

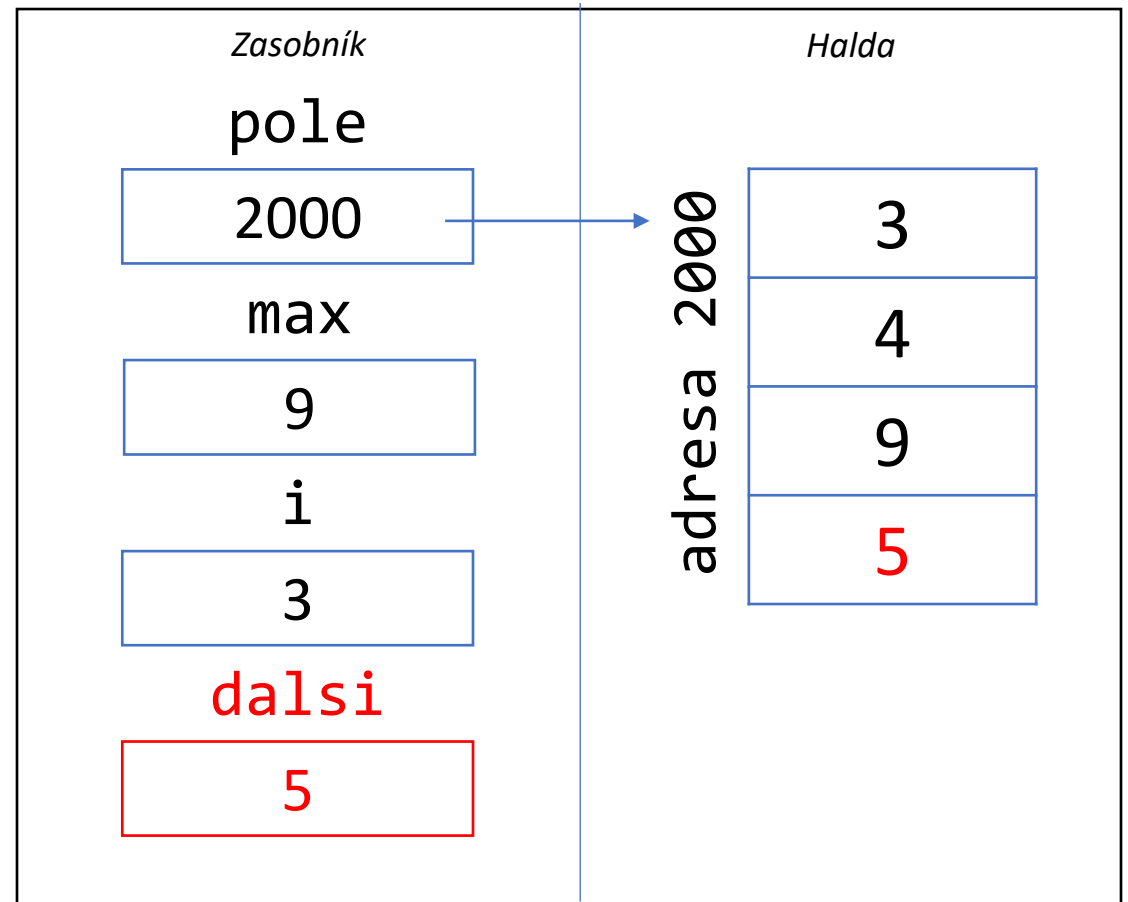
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

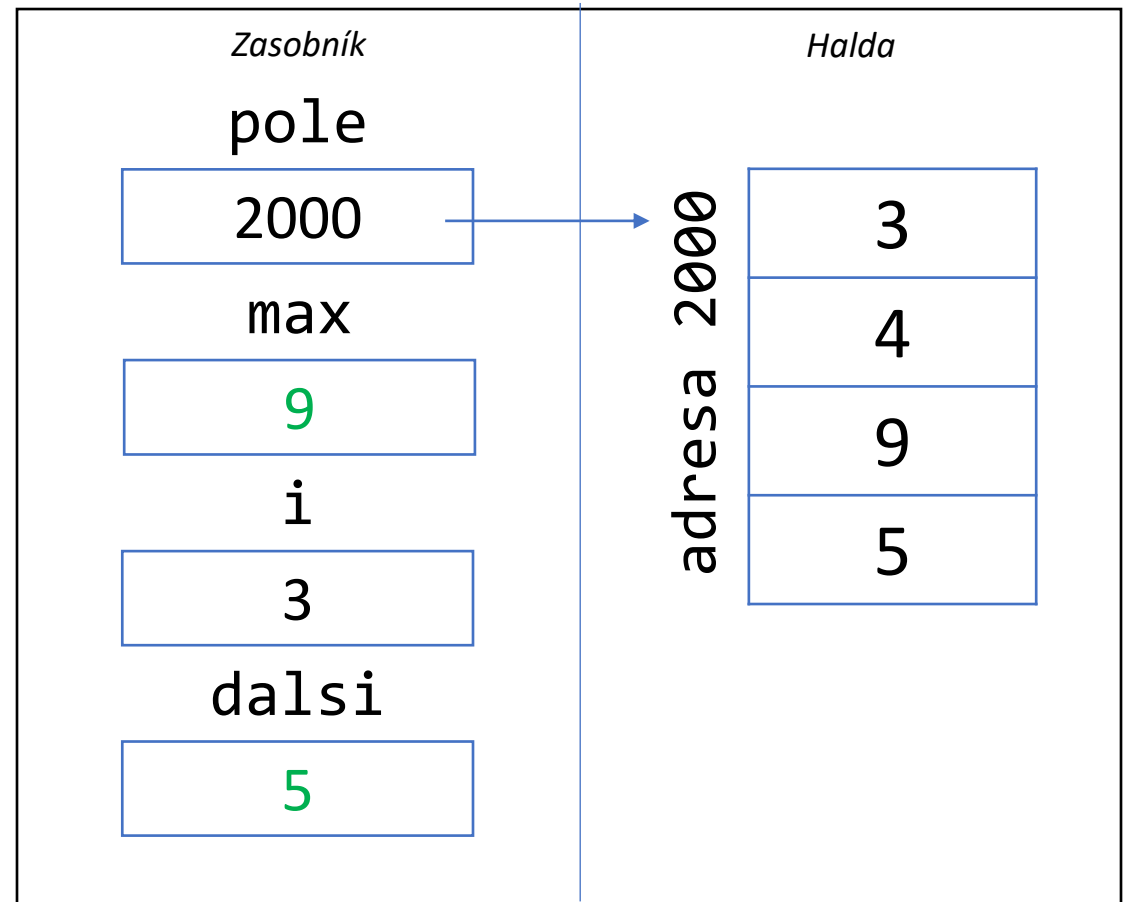
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

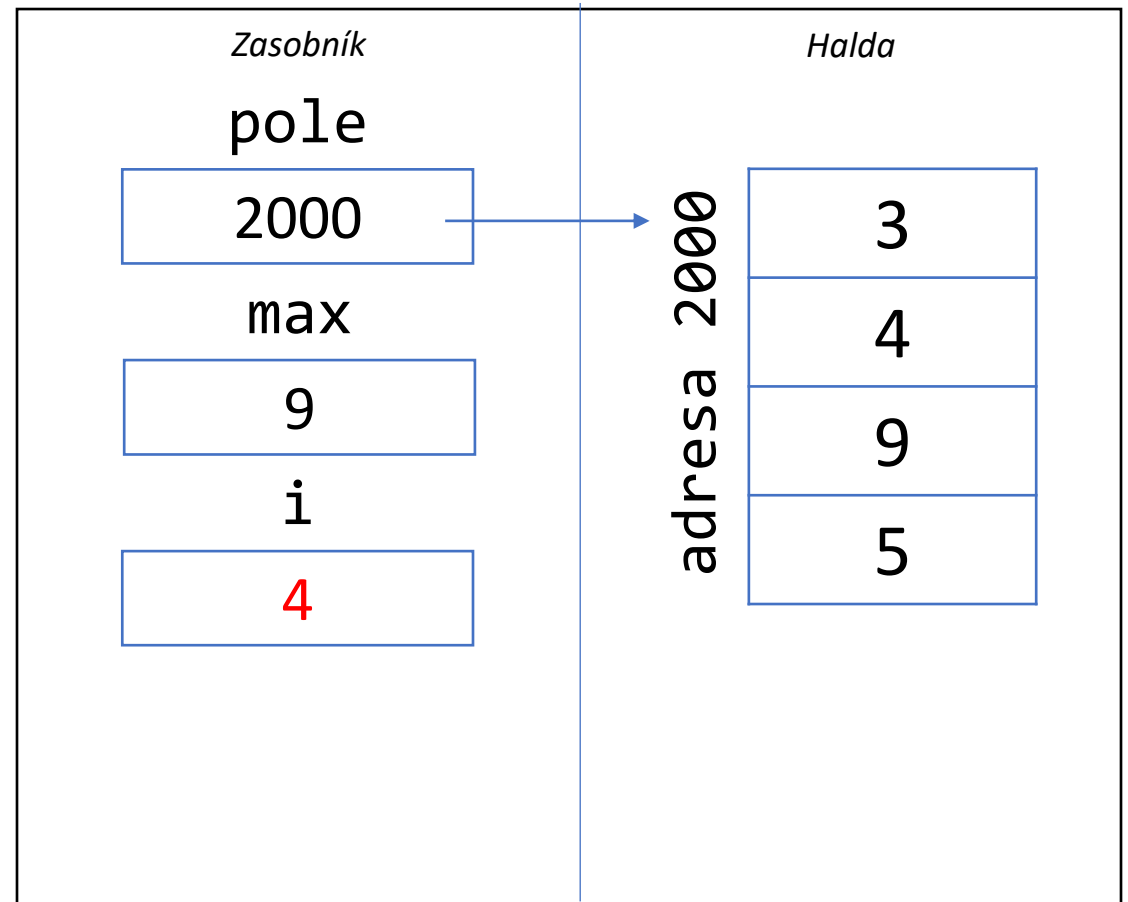
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

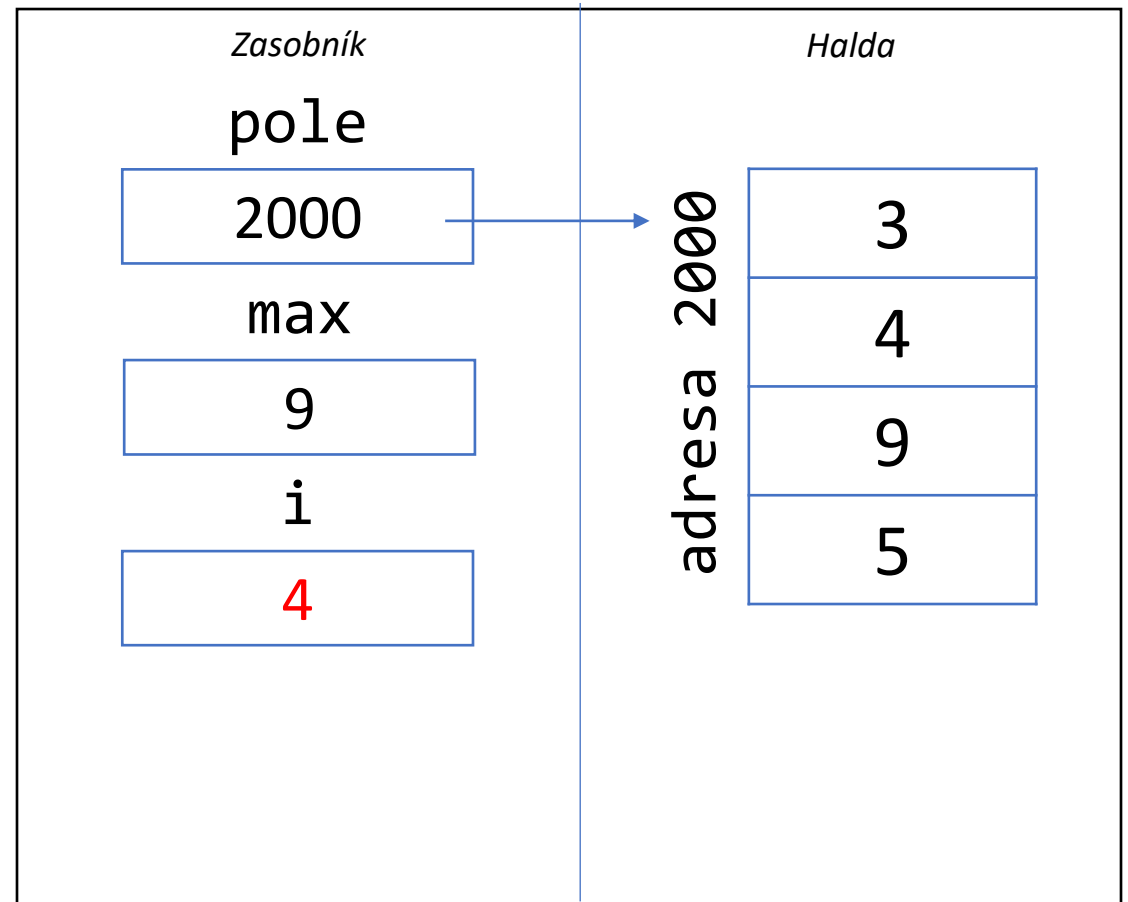
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}
```

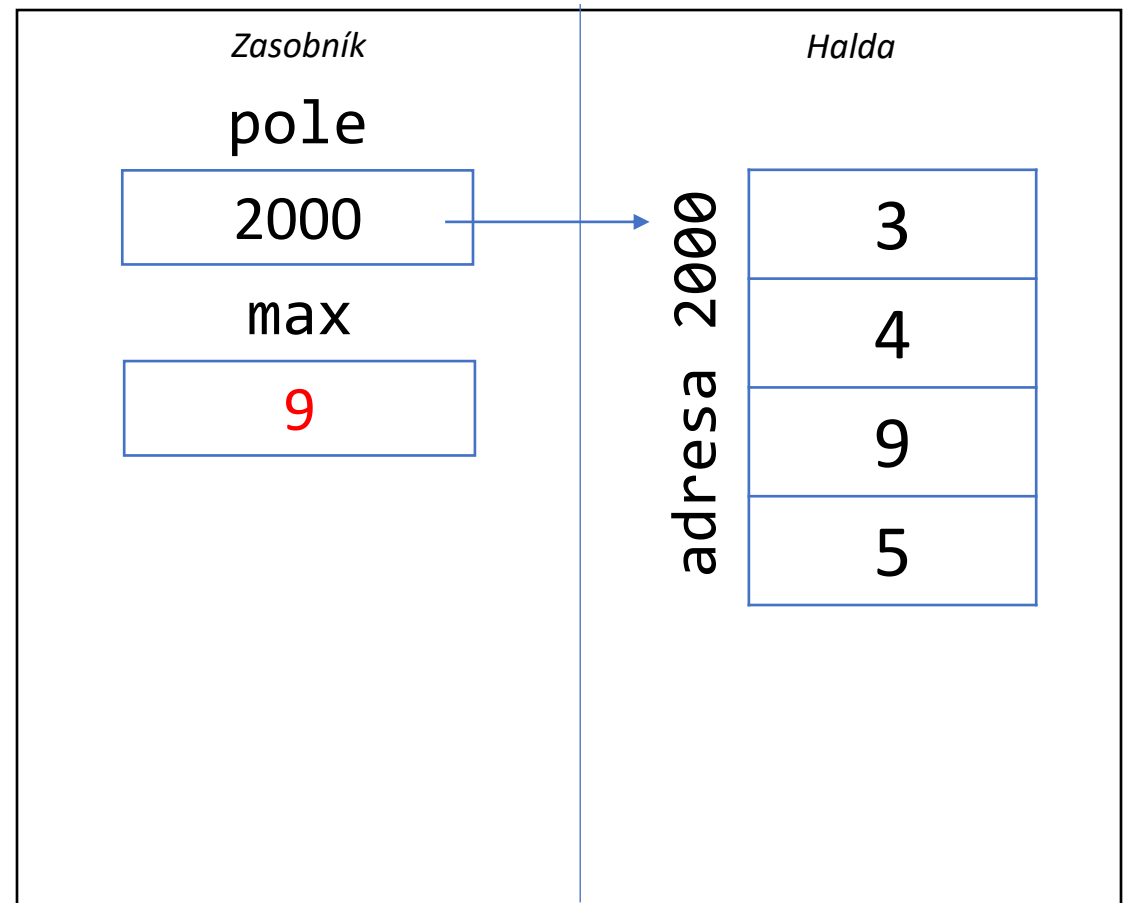
Paměť RAM



Maximální hodnota v poli

```
int[] pole = new int[] { 3, 4, 9, 5 };  
  
int max = pole[0];  
  
for (int i = 1; i < pole.Length; i++)  
{  
    int dalsi = pole[i];  
  
    if (dalsi > max)  
    {  
        max = dalsi;  
    }  
}  
  
Console.WriteLine(max);
```

Paměť RAM



Poznámka k proměnné *dalsi*

- V příkladu je opět znázorněné, že proměnná *dalsi* je definovaná znovu při každé iteraci cyklu *for* a bylo by vhodnější ji nadefinovat před cyklem. Takto se to dříve dělalo například v jazyce C.
- Ve skutečnosti **kompilátor** tento **kód zoptimalizuje** a není nutné proměnnou *dalsi* definovat před cyklem *for* a dáváme přednost **lepší čitelnosti kódu**.

Použité zdroje

[1] Single-Dimensional Arrays - C# Programming Guide | Microsoft Docs. [online]. Copyright © Microsoft 2021 [cit. 02.02.2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/single-dimensional-arrays>



EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání



Programování a algoritmizace

Děkuji za pozornost

Strategický projekt UTB ve Zlíně, reg. č. CZ.02.2.69/0.0/0.0/16_015/0002204



Ing. et Ing. Erik Král, Ph.D.
FAI, ÚPKS