

Šablony

Šablony jsou generické funkce nebo třídy, ve kterých jsou parametrem datové typy nebo celočíselné hodnoty.

Šablony funkcí

Zápis šablony:

```
template<..parametry..> definice_funkce
```

Je-li parametrem datový typ, lze pro jeho označení použít klíčová slova:

```
class  
typename
```

Pro identifikátory parametrů, které označují datové typy, se obvykle používají velká písmena (T apod.).

Příklad. Funkce počítající absolutní hodnotu.

```
template<typename T>  
inline T abso(T a) { return a>=0 ? a : -a; }  
  
template<class T>  
inline T abso(T a) { return a>=0 ? a : -a; }
```

Volání funkce:

```
jméno_funkce<..skutečné_datové_typy..>(..skutečné_parametry..)
```

Pokud lze skutečné datové typy odvodit ze skutečných parametrů, lze jejich uvedení ve volání funkce vynechat.

Příklad. Volání funkce z předchozího příkladu.

```
abso<int>(-3)          abso(-3)  
abso<double>(-3.1)    abso(-3.1)  
abso<float>(-3.1)     abso(-3.1f)
```

Příklad. Šablona funkce třídění Quicksort.

```
template<class T>  
void quicksort(T a[],int k,int l)  
{  
    T x=a[(k+l)/2];  
    int i=k,j=l;  
    do { while (a[i]<x) ++i;  
          while (x<a[j]) --j;  
          if (i>j) break;  
          T w=a[i];  
          a[i]=a[j];  
          a[j]=w;  
    } while (i<j);  
}
```

```

        ++i; --j; } while (i<=j);
    if (k<j) quicksort(a,k,j);
    if (i<l) quicksort(a,i,l);
}

float a[] = { 3.5,1,8,2.2,5,7 };
quicksort(a,0,5);

```

Příklad. Šablona funkce, která má dva parametry datových typů.

```

template<typename T,typename S>
inline bool jeVIntervalu(T x,S a,S b) { return a<=x && x<=b; }

jeVIntervalu<double,int>(-3.1,0,10)

jeVIntervalu<double>(-3.1,0,10)

jeVIntervalu(-3.1,0,10)

```

Vedle parametrů reprezentující datové typy můžeme mít parametry reprezentující celočíselné hodnoty. Patří sem všechny datové typy, které mají charakter celočíselných hodnot:

- celočíselné datové typy (`int` apod.)
- výčtové typy
- ukazatelé
- reference

Příklad. Šablona funkce, která transponuje matici a řádu $m \times n$ a výsledek ukládá do matice b řádu $n \times m$.

```

template<class T, int m, int n>
void transp(const T a[m][n], T b[n][m])
{
    for (int i=0;i<m;++i) for (int j=0;j<n;++j) b[j][i]=a[i][j];
}

float a[2][3]={ {7,4,1},{2,5,3}}, b[3][2];

transp<float,2,3>(a,b);

```