ШАБЛОНИ

гл.ас., д-р. Нора Ангелова

Шаблони на функции

- Шаблоните на функциите - позволяват създаването на функции, използващи неопределени (хипотетични) типове данни за свои параметри или за резултата от обръщението към функцията.
- Чрез тях се описват „обобщени“ функции.

Шаблони на функции

- Функция, която въвежда елементите на масив от цели числа

```
void readIntArr(int n, int* a) {
  for (int i = 0; i < n; i++) {
    cout << "a[" << i << "]= ";
    cin >> a[i];
  }
}
```

- Функция, която въвежда елементите на масив от реални числа

```
void readDoubleArr(int n, double* a) {
  for (int i = 0; i < n; i++) {
    cout << "a[" << i << "]= ";
    cin >> a[i];
  }
}
```

Шаблони на функции

- Дефиниция на шаблон на функция

<шаблон_на_функция> ::=

template <typename <параметър> {, typename <параметър>}$_{опц}$>

<тип> <име_на_шаблон_на_функция> (<формални_параметри>)

<тяло>

Пример:

```
template <typename T>
void testTemplateFunction(int n, T* a) {
  …
}
```

Шаблони на функции

- Използване на шаблон на функция

Използването на дефинираните шаблони на функции се осъществява чрез обръщение към „обобщената“ функция, която шаблонът дефинира, с параметри от конкретен тип.

Компилаторът генерира т. нар. шаблонна функция, като замества параметрите на шаблона с типовете на съответните фактически параметри.

При това заместване не се извършват преобразувания на типове.

Шаблони на функции

- Пример

Да се дефинира шаблон на функция за въвеждане на елементите на едномерен масив (за елементите е дефиниран операторът >>)

```
template <typename T>
void read(int n, T* a) {
  for (int i = 0; i < n; i++) {
    cout << "a[" << i << "]= ";
    cin >> a[i];
  }
}

…
int a[10]; double b[10];
read(n, a);
read(n, b);
```

Шаблони на класове

- Шаблоните на класове - позволяват създаването на класове, използващи неопределени (хипотетични) типове данни за свои член-данни, за параметри на член-функции, за резултати от обръщения към член-функции.
- Чрез тях се описват „обобщени“ класове – класове, зависещи от параметри.
- Използват се за изграждане на общоцелеви класове - контейнери (стекове, опашки, списъци и др.).

Шаблони на класове

- Декларация на шаблон на клас

<декларация_на_шаблон_на_клас> ::=

template < <списък_от_параметри> >

class <име_на_шаблон_на_клас>

<тяло>

<списък_от_параметри> ::=

typename <параметър> {= <тип>}$_{опц}$
{, typename <параметър> {= <тип>}$_{опц}$ }$_{опц}$

Шаблони на класове

- Подразбиращи се стойности

Ако параметър е с подразбираща се стойност, всички параметри след него също трябва да са с подразбиращи се стойности.

Шаблони на класове

Пример:

Шаблон на клас с два параметъра, вторият от които е подразбиращ се.

```
template <typename T, typename S = int>
class example {
  public:
    T func1(T, S);
    S func2(T, S);
  private:
    T a;
    S b;
};
```

Шаблони на класове

- Дефинирането на член-функциите на шаблон на клас се осъществява като:
  - вградени (inline) член-функции;
  - извън тялото на класа.

Шаблони на класове

Пример:

- Вградени член-функции

```
template <typename T, typename S = int>
class example {
  public:
    T func1(T x, S y) {  // вградена член-функция
      cout << "func1 \n";
      return x;
    }
    S func2(T, S);       // невградена член-функция
  private:
    T a;
    S b;
};
```

Шаблони на класове

Пример:

- Извън тялото на класа

Дефиницията се предшества от

`template < <списък_от_параметри> >`

```
template <typename T, typename S>
S example<T, S>::func2(T x, S y) {
  cout << "func2\n";
  return y;
}
```

Шаблони на класове

Ако някой <тип> е пропуснат се използва типът по подразбиране, ако декларацията на

шаблона е с подразбиращи се параметри или се

съобщава за грешка.

При срещане на дефиниция на шаблонен клас компилаторът използва зададените типове и генерира съответен клас.

Шаблони на класове

Пример

```
typedef example<int, double> CL1
```

Дефинира класа CL1, който е специализация на шаблона на класа example при T – int и S – double.

Шаблони на класове

Забележка

Ако и двата параметъра на шаблона на класа example са подразбиращи се, ще е възможна и специализацията:

```
typedef example<> CL3;
```

* Скобите <> трябва да присъстват.

Шаблони на клас

Шаблонът на клас дефинира съвкупност от класове.

Понякога се налага за конкретен тип данни член-функция на шаблона на класа да се реализира по по-различен алгоритъм.

В C++ е възможно предефинирането на член-функция на шаблон на клас за конкретен тип.

Този процес се нарича **специализация на член-функцията за съответния тип**.

КРАЙ