

212-Хачатурова-эссе-templates

Во избежание повторяемости написания кода для разных типов данных, в языке C++ используются шаблоны (templates).

Шаблон класса позволяет оперировать данными разных типов в общем. То есть, нет привязки к некоторому конкретному типу данных (int, float, ...). Вся работа выполняется над некоторым обобщенным типом данных, например, типом с именем T. Фактически, объявление шаблона класса есть только описанием. Создание реального класса с заданным типом данных осуществляется компилятором в момент компиляции, когда объявляется объект класса.

Общая форма объявления шаблонного класса:

```
template <typename T> //T – обобщенное имя типа, используемое методами класса
class ClassName
{
    // тело класса
    // ...
};
```

Преимущества использования шаблонов классов:

- Избежание повторяемости написания программного кода для разных типов данных. Программный код (методы, функции) пишется для некоторого обобщенного типа T. (Далее в программе можно заменять T на int, double, char, и т.д.);
- Уменьшение текстовой части программного кода, и, как следствие, повышение читабельности программ;
- Обеспечение удобного механизма передачи аргументов в шаблон класса с целью их обработки методами класса.

Определение функций вне шаблона класса:

Синтаксис определения функций вне шаблона класса может немного отличаться от их определения внутри шаблона. В частности, определения функций вне шаблона класса должны определяться как шаблон, даже если они не используют параметры шаблона.

Параметры шаблонов могут иметь значения по умолчанию - тип по умолчанию, который будет использоваться. Например:

```
template <typename T=int> //тип по умолчанию - int
class ClassName {
};
```