

## **АВТОМАТИЧЕСКИЙ ВЫВОД ТИПА ПЕРЕМЕННЫХ В СОВРЕМЕННЫХ ЯЗЫКАХ ВЫСОКОГО УРОВНЯ**

Трудоемкость, материалоемкость и дороговизна натурных экспериментов, их ограниченность, многомерность и нелинейность исследуемых процессов и явлений, а также современное развитие вычислительной техники и программного обеспечения (ПО) значительно актуализировало математическое моделирование, как один из наиболее доступных методов исследования объектов и процессов, происходящих во всех сферах человеческой деятельности.

Эффективность численных методов, используемых при моделировании, во многом зависит от языков и систем программирования. Повышение их эффективности тесно связано с совершенствованием информационного обеспечения, организацией обработки и управления данными, процедурами, моделями, декомпозицией больших программ, семантическими, синтаксическими и морфологическими возможностями языка, технологиями программирования и т.п.

Прогресс компьютерных технологий сопровождается созданием новых и совершенствованием существующих средств взаимодействия программистов с ЭВМ – языков программирования (ЯП).

В настоящее время известно большое количество работ, посвященных общему анализу ЯП. В то же время актуальной остается проблема систематизации ЯП в соответствии с методологией хранения и обработки данных.

Авторами настоящей работы проведено исследование особенностей вывода типов переменных в наиболее популярных (рейтинговых) на текущий момент ЯП для правильного выбора необходимого ПО при математическом моделировании.

Вывод типов (англ. *type inference*) – в программировании возможность компилятора самому логически вывести тип значения выражения. Это не только сокращает размер исходного кода и повышает его лаконичность, но и нередко повышает повторное использование кода.

Одним из наиболее часто используемых в настоящее время, рейтинговых языков высокого уровня является C++ – компилируемый ЯП общего назначения, поддерживающий объектно-ориентированную и процедурную парадигмы. C++ – статически типизированный ЯП, т.е. тип любой переменной известен на этапе компиляции. Более того, программист обязан указать тип используемой переменной. В некоторых случаях имена оказываются очень громоздкими. Традиционно для решения этой проблемы использовались псевдонимы типов (*typedef*), позволяющие сократить длину идентификатора типа и избавиться от потенциальных проблем несовместимости типов. Это способ работает и в C++11, но появился и новый – если переменная инициализируется в объявлении, то в качестве ее типа можно указать *auto*. В этом случае компилятор автоматически выведет тип переменной из типа инициализатора.

После появления шаблонных типов и техник шаблонного метапрограммирования тип некоторых значений, в особенности возвращаемых значений функций, не может быть легко задан. Это приводит к сложностям при хранении промежуточных данных в переменных, иногда может потребоваться знание внутреннего устройства конкретной библиотеки метапрограммирования.

Поэтому в C++11 были добавлены два способа вывода типов: *auto* – для создания переменных с типом на основе присваиваемого выражения; *decltype* – для определения результирующего типа произвольного выражения. Однако типы, выводимые *auto* и *decltype*, отличаются между собой. В частности, *auto* всегда выводит не-ссылочный тип, тогда как *auto&&* всегда выводит ссылочный. Тем не менее, результатом *decltype* может быть как ссылочный тип, так и не-ссылочный, в зависимости от обрабатываемого выражения:

В C++14 добавлен синтаксис *decltype(auto)*. Этот синтаксис позволяет использовать правила *decltype* для объявлений *auto*.

Синтаксис *decltype(auto)* также можно использовать для вывода типов возвращаемых значений, если указать *decltype(auto)* вместо *auto* на месте типа возвращаемого значения функции.

Выполнено исследование особенностей вывода типов переменных рейтинговых ЯП для правильного выбора необходимого ПО при решении задач математического моделирования.