

## СТАТИЧЕСКАЯ ПРОВЕРКА ТИПОВ ДЛЯ ПРОГРАММ, РАЗРАБОТАННЫХ НА ПЛАТФОРМЕ «1С:ПРЕДПРИЯТИЕ»

*А. С. Балюк, В. А. Попова*

Стремительное развитие информационных технологий позволяет с каждым годом всё больше автоматизировать различные производственные процессы. В связи с потребностью непрерывного внедрения новых функциональных возможностей в информационные системы повышаются риски возникновения ошибок, которые в значительной степени ухудшают качество программных продуктов.

На текущий момент создаются различные механизмы для устранения дефектов программного кода, которые могут возникнуть в результате внедрения новой функциональности или внесения изменений в уже существующие компоненты систем. К таким системам относятся программы, разрабатываемые на платформе «1С:Предприятие», в которых часть ошибок связана с динамической типизацией, применяющейся в языке программирования 1С. Основной характеристикой динамической типизации является присваивание типов переменным в момент выполнения программы. В отличие от языков программирования со статической типизацией, где проверка типов осуществляется в момент компиляции программы и при этом тип присваивается переменной один раз, в языке программирования 1С тип одной и той же переменной может быть изменён в течение всего процесса работы программы. В связи с этим могут возникать ошибки несоответствия типов [1].

Для отслеживания ошибок, возникающих в языках программирования с динамической системой типов, следует использовать механизмы, которые позволяют анализировать код программы без её запуска. На сегодняшний день для подобных задач используются методы статического анализа программного кода. Механизмы отслеживания ошибок уже были разработаны для некоторых языков программирования с динамической типизацией. Например, компания Google создала такой инструмент проверки типов для языка программирования Javascript в рамках проекта Closure Compiler [3].

Сложность разработки механизма статической проверки типов для языка программирования 1С обуславливается большим количеством типов и методов, которые используются для создания конфигураций. Некоторые типы и методы предоставляется разработчикам в качестве инструментов встроенного языка платформы «1С:Предприятие». Информация о таких типах и методах содержится во встроенной справке, а также доступна на портале 1С:ИТС [2]. Ещё необходимо учитывать, что в программах, написанных на 1С, существуют прикладные типы, такие как Справочники, Документы, Перечисления и т. д., которые также необходимо включить в механизм проверки соответствия типов.

Таким образом, была поставлена задача — разработать программный комплекс для выполнения статической проверки типов в программах, разработанных на языке программирования 1С.

Предполагается, что разработанный механизм статической проверки типов для языка 1С позволит выявить существующие ошибки и в дальнейшем избе-

жать проблем с динамической типизацией в любой конфигурации, разработанной на платформе «1С:Предприятие».

Для хранения типов и методов выбран формат XML. Такой выбор объясняется тем, что для XML-документов существуют строгие правила разметки, благодаря которым соблюдается единая структура документа. Также в XML имеется возможность использовать ссылки между элементами, что также являлось важным критерием при выборе формата, так как типы и методы могут быть связаны между собой.

Для того чтобы формировать представление типа или метода в формате XML, была разработана схема XSD, которая описывает следующее содержание XML-документа:

- `heading` — имя типа или метода на русском и английском языках;
- `parameters` — входные параметры и их типы;
- `returnedValue` — множество возвращаемых типов;
- `access` — перечисление режимов работы 1С, в которых доступно использование описываемого типа или метода.

Представление всех типов и методов в формате XML образует базу типов. Для формирования такой базы осуществляется написание обработки на платформе «1С:Предприятие». В дальнейшем будет построена система правил, которая позволит определять, какие типы значений может принимать и возвращать каждый метод конфигурации, к которой применяется механизм статической проверки типов.

### Литература

1. Бронштейн И. Е. Подход к обнаружению ошибок несоответствия типов в коде на динамических языках программирования / И. Е. Бронштейн // Труды института системного программирования РАН. – 2013. – Т. 25. – с. 67-84.

2. 1С. Информационно-технологическое сопровождение [Электронный ресурс]. – URL: <https://its.1c.ru/> (дата обращения: 24.04.2020)

3. Closure Compiler. Google Developers [Электронный ресурс]. – URL: <https://developers.google.com/closure/compiler> (дата обращения: 22.04.2020).