УДК 004.04

ОСОБЕННОСТИ РЕАЛИЗАЦИИ WEB-СЕРВИСОВ, ОТВЕЧАЮЩИХ ЗА УДАЛЕННУЮ КОМПИЛЯЦИЮ И ВЫПОЛНЕНИЕ ПРОГРАММНОГО КОДА

Ю.С. КУВЕЦКИЙ
(Представлено: канд. техн. наук И.Б. БУРАЧЕНОК)

Разработан универсальный синтаксис шаблонных конструкцийдля составления различных шаблонов программ. Представлена структура алгоритма для преобразования полученного шаблона в код на современных высокоуровневых языках программирования: C#, Java и Ruby. Рассмотрены особенности реализации web-сервисов, позволяющих компилировать и выполнять переданный им по сети Internet программный код.

На данный момент разработчикам программного обеспечения для того, чтобы составить и запустить простую программу на одном из популярных, высокоуровневых языков программирования, необходимо потратить много времени и сил, особенно если сталкиваешься с этим впервые. Поэтому и возникает необходимость в создании ПО, позволяющего просто и быстро сформировать код, скомпилировав и выполнив его удаленно. *Отличительная особенность* данного ПО – это возможность составить программный код на одном из высокоуровневых языков программирования без знания их синтаксиса. С целью создания легко расширяемого и достаточно универсального ПО, позволяющего работать с различными языками программирования через общие интерфейсы, необходимо решить следующие задачи. Разработать универсальный синтаксис шаблонных конструкций, который позволил бы составлять шаблоны различных программ, а также разработать алгоритм для преобразования полученного шаблона в код на популярных сегодня языках программирования, например, C#, Java и Ruby.

Вначале подробно разберем *синтаксис шаблонных конструкций*. Сами конструкции позволяют выводить данные, писать часть кода программы вручную, а также позволяют описать цикл и условный блок. Перечисленных возможностей достаточно для составления большинства алгоритмов. Детальный анализ уже существующих синтаксисов для построения шаблонов, например, используемые в таких популярных фреймворках, как DjangoTemplates (Python) [1] и Razor (C#) [2], позволил выделить следующие базовые конструкции разрабатываемого синтаксиса:

- произвольный текст, не заключенный в «кодовые скобки» («{%» и «%}»), выводится без изменений;
- конструкция вида «{% программный код %}» приводит к выполнению соответствующего кода;
- конструкция, заключенная в скобки «{%@ выражение %}» и «{%@%}», выражение должно иметь числовой результат, выполняется количество раз, заданное значением выражения;
- конструкция вида «{%= выражение %}» выводит результат выражения любого типа;
- конструкция, заключенная в скобки «{%? выражение %}» и «{%?%}», выражение должно иметь результат типа boolean, выполняется при условии того, что выражение приняло значение истины – true.

На рисунке 1 приведен пример использования шаблонных конструкций с использованием языка программирования C#.

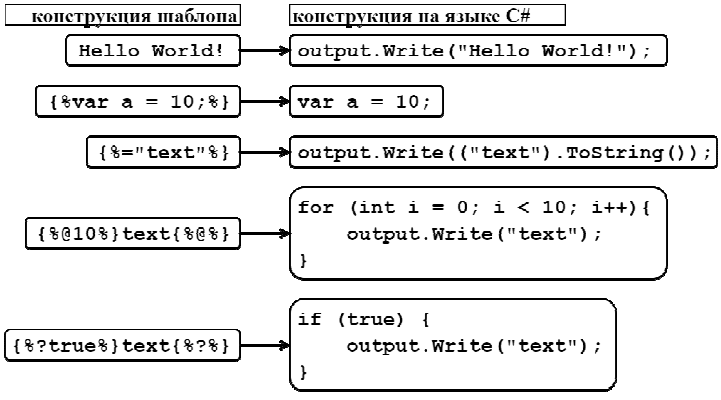

Рисунок 1. – Пример работы шаблонных конструкций

Имея синтаксис шаблона, разработан алгоритм его «парсинга» в программный код. Этот алгоритм основан на использовании регулярных выражений для разбора шаблона по отдельным частям. Плюсы данного способа в том, что из элементарных частей шаблона просто составить программные коды на

различных языках программирования, так как структура программ на выходе алгоритма одна, и их отличия только в синтаксисе и некоторых языковых особенностях.

Изучив все возможности регулярных выражений [3], составление необходимого выражения не представляет сложности. Алгоритм имеет следующую структуру:

1. Шаблон разбивается на части двух типов:
- части, заключенные в функциональные скобки («{%» и «%}»);
- части, вне функциональных скобок.
2. Для каждой из частей определяется ее тип. Это могут быть следующие типы:
- EndChunk – закрывающие конструкции ({%@%}», «{%?%}»);
- TextChunk – части шаблона, которые вне скобок и должны выводиться как есть;
- ChunkWithBraces – основные элементы шаблона (блок для цикла, условный блок, блок вывода, блок кода программы, блок вывода содержимого).

На рисунке 2 приведено регулярное выражение, отвечающее за первый пункт алгоритма.

```
(?<Braces>\{%(?:[\s\S]*?)%\})|(?<Text>(?:(?<=%\})|(?:^))(?:[\s\S]+?)(?:(?=\{%)|(?:$)))
```

Рисунок 2. – Регулярное выражение, используемое для разбора шаблона по частям

После того как описанный алгоритм сформировал набор блоков (chunks) из текста шаблона, свою работу начинают алгоритмы, цель которых составить код программы на одном из языков программирования. Данные алгоритмы на первый взгляд очень похожи, однако отличительные особенности, свойственные отдельному языку программирования, все же имеются.

После составления кода программы необходимо выполнить компиляцию кода и его выполнение. Исходя из того, что данный процесс должен происходить удаленно, рассмотрен протокол SOAP как способ реализации удобного и безопасного общения клиента с web-сервисами по сети Internet. Составленный по шаблону код программы отправляется на web-сервис, где компилируется и выполняется. Так как сами языки программирования разнообразны, то реализации нужных сервисов тоже имеют некоторые отличия.

Сервис для компиляции и выполнения программ на языке программирования Java отличается тем, что для каждой новой полученной программы необходимо создавать новый файл на сервере с уникальным именем. Это связано с тем, что компиляция и выполнение программ на языке Java защищено от постоянного нагрузочного компилирования и выполнения при помощи техники кэширования. Если попытаться изменить содержимое ранее созданного на сервере файла и после этого перекомпилировать его и запустить, то выполнится программа, скомпилированная ранее. Однако при создании новых файлов возникает проблема, связанная с необходимостью удаления ненужных. Ее решением является очистка файлов пользователя при завершении его работы.

Веб-сервис для запуска скриптов на языке программирования Ruby работает немного иначе. Тот факт, что язык Ruby – скриптовый язык (т.е. выполняется построчно) убирает необходимость в компиляции, так как этот термин не используется в отношении к скриптам языка Ruby, также отпадает необходимость в создании каких-либо файлов, так как код будет выполняться в eval-пространстве.

В результате проведенного анализа определен синтаксис для составления универсальных шаблонов, использование которых позволит описывать алгоритмы различной сложности, с последующим представлением этих алгоритмов в виде программного кода на различных языках программирования. Также разработан легко расширяемый, независимый от используемого языка программирования алгоритм для разбора шаблона и преобразования в программный код, который обеспечивает высокую производительность и снижает время формирования кода. Кроме этого рассмотрены некоторые особенности реализации web-сервисов, отвечающих за удаленную компиляцию и выполнение программного кода. Указанные особенности, анализ, подходы позволяют осуществить разработку программного обеспечения, отвечающего современным требованиям, что важно при поддержке нескольких высокоуровневых языков программирования и использовании web-сервисов для передачи программного кода и результатов компиляции или выполнения его по сети Internet.

ЛИТЕРАТУРА

1. Django. Templates [Электронный ресурс] / DjangoSoftwareFoundation. – Режим доступа: https://docs.djangoproject.com/en/1.11/ref/templates/. – Дата доступа: 15.09.2017.
2. W3Schools.com. WS-Security.ASP.NET Razor – C# and VB Code Syntax [Электронный ресурс] / W3Schools. – Режим доступа: https://www.w3schools.com/asp/razor_syntax.asp. – Дата доступа: 15.09.2017.
3. Lattyf. Тонкости регулярных выражений [Электронный ресурс] // Хабрахабр. – Режим доступа: https://habrahabr.ru/post/112016/. – Дата доступа: 16.09.2017.