

Развитие языка в реальном мире и для него: C++ 1991–2006

Бьярне Страуструп

Перевод — Пеленицын А. М.

4. Стандартная библиотека: 1991–1998

4.1 STL

4.1.8 Точка зрения Степанова

Описание STL здесь (естественным образом) сфокусировано на вопросах языка и библиотеки в контексте C++. Чтобы получить полную картину, я попросил Александра Степанова высказать свое мнение:

В октябре 1976 я отметил, что конкретный алгоритм — параллельная редукция — был связан с монойдами: наборами элементов с ассоциативной операцией. Это наблюдение вселило в меня уверенность, что любой полезный алгоритм можно связать с некоторой математической структурой и что такая связь одновременно приведет к наиболее широкому применению и позволит создать выразительную классификацию. Также как математики научились «поднимать» свои теоремы в наиболее общие условия, так и я хотел поднимать алгоритмы и структуры данных. Вам редко требуется знать точный тип данных, на котором работает алгоритм, так как большинство алгоритмов работает сразу на многих схожих типах. Чтобы написать алгоритм, вам нужно знать только свойства операций над данными. Я называю набор типов со схожими свойствами, на которых алгоритм имеет смысл, лежащей в основе алгоритма концепцией. Также, чтобы надлежащим образом выбрать эффективный алгоритм, вам нужно знать сложность этих операций. Другими словами, сложность это неотъемлемая часть интерфейса концепции.

В конце 70-х я познакомился с работой Джона Бэкуса по функциональному программированию. В то время как его идея программирования в функциональных формах произвела на меня впечатление основополагающей, я осознавал, что его попытка жестко зафиксировать количество функциональных форм была изначально неверной. Число функциональных форм — или, как я называю их теперь, обобщенных алгоритмов — постоянно растет с открытием новых алгоритмов. В 1980 вместе с Дэвидом Мёссером и Дипаком Капуром я начал работу над языком Тектон, чтобы описать алгоритмы, определенные на алгебраических структурах. Язык был функциональным, ибо я не представлял в то время, что память и указатели являются фундаментальной составляющей программирования. Я также уделил время изучению Аристотеля и его последователей, что привело меня к лучшему пониманию основных операций над объектами, такими как сравнения на равенство и копирование, и отношению между частью и целым.

В 1984 я стал сотрудничать с Ароном Кершенбаумом, который был экспертом по алгоритмам на графах. Ему удалось убедить меня принять массивы всерьез. Я смотрел на последовательности, как на рекурсивно определяемые объекты, поскольку это традиционно воспринималось как «научнообразный» подход. Арон показал мне, что многие фундаментальные алгоритмы полагаются на произвольный доступ. Мы сконструировали большое множество компонент на языке Scheme и были в состоянии реализовать в обобщенном виде сложные графовые алгоритмы.

Работа на Scheme привела к гранту на создание обобщенной библиотеки на Ada. Дэйв Мёссер и я создали обобщенную библиотеку, работавшую со связными структурами. Мои попытки реализовать алгоритмы, которые работали бы с любой последовательной структурой (одновременно со списками и массивами), потерпели неудачу в силу ограничений компиляторов Ada, существовавших на тот момент. У меня было эквиваленты многих STL алгоритмов, но я не мог скомпилировать их. Основываясь на этой работе, Дэйв Мёссер и я опубликовали статью, где мы вводили понятие обобщенного программирования, настаивая на выведении абстракций из полезных эффективных алгоритмов. Наиболее важной вещью, которую я вынес из Ada, была статическая типизация как инструмент проектирования. Бьярне Страуструп усвоил этот урок из Simula.

В 1987 в Bell Labs Энди Кёниг научил меня семантики C. Абстрактная машина, стоявшая за C, стала откровением. Я также прочитал много кода UNIX и Plan 9: код Кена Томпсона и Роба Пайка, конечно, повлиял на STL. Но в любом случае, в 1987 C++ не был готов для STL, а мне нужно было двигаться дальше.

К тому времени я открыл для себя работы Эйлера, и моё представление о природе математики подверглось радикальному изменению. Я был «де-Бурбакизован», перестал верить в множества и был изгнан из Канторовского рая. Я все еще верю в абстракцию, но знаю, что тот, кто заканчивает абстракцией, начинает не с неё. Я усвоил также, что необходимо адаптировать абстракцию к реальности, а не наоборот. Математика перестала быть наукой о структурах, но переродилась для меня в науку о числах и формах.

В 1993 после пяти лет работы над сторонними проектами, я вернулся к обобщенному программированию. Энди Кениг предложил мне написать план для включения моей библиотеки в Стандарт C++, Бьярне Страуструп с энтузиазмом принял этот план, и менее чем через год STL была принята в стандарт. STL это результат 20 лет размышлений, но менее чем двух лет разработки.

STL достигла ограниченного успеха. Хотя она стала широко используемой библиотекой, ее центральная идея не прояснена. Люди путают обобщенное программирование с использованием (и злоупотреблением) шаблонов C++. Обобщенное программирование имеет дело с абстрагированием и классификацией алгоритмов и структур данных. Оно черпает свой дух у Кнута, а не в теории типов. Оно имеет своей целью постепенное конструирование систематических каталогов полезных, эффективных и абстрактных алгоритмов и структур данных. Этот предприятие все еще остается мечтой.