УДК 370+000

ОПЫТ ОБУЧЕНИЯ ОБЪЕКТНООРИЕНТИРОВАННОМУ ПРОГРАММИРОВАНИЮ И C++11

Дончев Ивайло, Тодорова Эмилия

Университет им св. Кирилла и Мефодия г. Велико Тырново, Болгария

Аннотация

Язык C++ является самым распространенным для введения в программирование и программирование среднего уровня в болгарских университетах. В последние годы этот язык очень интенсивно развивается. Его абстракции еще более гибкие и удобные чем прежде. Реализованы столько дополнительных возможностей в новом стандарте, что его можно считать новым языком. Все эти перемены должны найти свое отражение в обучении и вынудят нас полностью реорганизовать наши курсы программирования. В настоящей работе представлены результаты наших усилий в это направление.

Abstract

C++ is the most commonly used language in introductory and intermediate programming courses in Bulgarian universities. This language had its rapid development in recent years. Its abstractions are more flexible and affordable than ever before. There are so many new features in the new standard, known as C++11 that it may be considered a new language. All these changes should find their place in teaching and will force us to utterly reorganize our programming courses. In this paper we share the results of our efforts in this direction.

Введение

Программирование является очень важным умением, которое все, изучающие компьютерные науки, должны усовершенствовать [2]. Однако обучение программированию традиционно сопутствуется немалым числом трудностей. Несмотря на большой опыт, приобретенный в течении более 50 лет, все еще это считается вызовом [3].

C++ самый распространенный язык программирования для вводных и промежуточных курсов в болгарских университетах. Этот выбор не случаен: C++ гибридный язык и позволяет разработчику воспользоваться преимуществами нескольких программных парадигм. С одной стороны, это дает преподавателям свободу в выборе методологии, с другой - на раннем этапе обучения студенты могут овладеть как минимум двумя программными парадигмами [1]. Нельзя отрицать что C++ тяжелый для начинающих. Его большие возможности идут рука об руку со сложным синтаксисом. Потому, учитывая трудности, встречаемые нашими студентами, интенсивное развитие языка в последние годы, а также современные требования о надеждном, безопасном, эффективном программировании, в Великотырновском университете мы начали реорганизацию CS1 и CS2 курсов с целью максимально облегчить начинающих, не снижая при этом качество обучения. Главным толчком для этого оказался новый стандарт ISO/IES для C++. В нем показаны важные изменения, которые рано или поздно должны найти отражение в учебных программах.

Наш реорганизованный курс объектноориентириванного программирования

Самый первый элемент, который мы использовали, был тип rvalue-ссылки. Эта возможность имелась у нас в среде Microsoft Visual Studio 2010 и естественным путем нашла свое место в CS2 курсе ООП в летнем семестре 2009/2010 года. С помощью rvalue-ссылок легко реализовать семантику переноса (move semantics). Можно утверждать что с точки зрения эффективности это самая важная особенность C++11. Семантика переноса позволяет компилятору заменить очень дорогую операцию копирования объектов более эффективными "state movement"операциями. В учебном курсе по ООП для C++ для семантики копирования выделено особое место потому что одна из основных целей обучения приобрести умение писать (т.к. большая часть студентов будущие профессиональные программисты) надёжный, защищенный и эффективный код [1], а это невозможно без реализации методов семантики копирования. По тем же причинам нельзя пренебрегать семантикой переноса. Rvalue-ссылочный тип может быть также использован для реализации идеальной пересылки - проблема, нерешенная до сих пор средствами генерического программирования в C++. Мы вводим студентов в проблематику с помощью классического примера генерической factory-функции, которая возвращает shared_ptr для нового конструированного генерического типа. Показываем недостатки реализации средствами, доступными до появления C++11 и следуя аргументации подобной в [5], достигаем до современного решения.

В 2011/2012 к семантики переноса и идеальной пересылки добавили три новые возможности языка, которые имелись в VC10 - автоматическое определение типа от инициализатора при декларации (auto keyword), decltype и новый suffix return type синтаксис. С 2012 года мы уделяем внимание и шаблонам с переменным числом аргументов (variadic templates). В следующем учебном году курс из элементов C++11 включим еще inilializer-list конструкторы, defaulted и deleted функции, наследованные конструкторы, in-class инициализаторы членов как и больше подробностей по шаблонам с переменным числом аргументов.

Верификация и валидирование программ имеют важное место в учебных планах специальностей Информатика и Компьютерные науки. Поэтому в следующем году нужно уделить соответствующее внимание этим темам в соответственно модели образовательной рамки, предложенной в [6].

Установленные проблемы и рекомендации

Студенты справляются сравнительно хорошо с семантикой переноса и с идеальной пересылкой. Вначале у нас были проблемы с пониманием работы функций std::move() и std::forward() и потому показывали

активирование семантики переноса через явное приведение типа до rvalue-ссылки. Удачной оказалась практика демонстрировать новые возможности расширением тех же классов, на которых до тех пор показывали только семантику переноса. Хороший пример должен включать класс с член-данными, указывающими на динамические ресурсы. Для этой цели подходят дефинированные программистом умный (smart) массив и цепочечный класс с перегрузкой операций (+, +=, =). Обязательно нужно показать и реализацию семантики переноса в наследственной йерархии и композиции классов.

Список использованных источников:

1. ACM/IEEE CS2008 Review Taskforce, Computer Science Curriculum 2008: An Interim Revision of CS 2001, Technical report, ACM/IEEE CS, 2008
2. ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 2001, Journal on Educational Resources in Computing, Volume 1 Issue 3es, Fall 2001, ACM New York, NY, 2001.
3. Caspersen, M., Bennedsen, J., Instructional design of a programming course: A learning theoretic approach, Proceedings of the Third International Workshop on Computing Education Research, September 15-16, Atlanta, Georgia, USA, 2007, pp. 111-122
4. ISO/IEC, International Standard ISO/IEC 14882:2011(E) Information technology - Programming languages - C++, Third edition, 2011-09-01
5. Hinnant, H., Bj. Stroustrup, and Br. Kozicki, A Brief Introduction to Rvalue References, The C++ Source (a web publication). March 10, 2008, available at http://www.artima.com/cppsource/rvalue.html
6. Todorova, M., Kanev, K., Educational framework for verification of object-oriented programs, in Proceedings of the 2012 Joint International Conference on Human-Centered Computer Environments, ACM New York, 2012, pp. 23-27