



ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ТҰҢҒЫШ ПРЕЗИДЕНТІ - ЕЛБАСЫНЫҢ ҚОРЫ

«ҒЫЛЫМ ЖӘНЕ БІЛІМ – 2017»

студенттер мен жас ғалымдардың
XII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ

СБОРНИК МАТЕРИАЛОВ

XII Международной научной конференции
студентов и молодых ученых
«НАУКА И ОБРАЗОВАНИЕ – 2017»

PROCEEDINGS

of the XII International Scientific Conference
for students and young scholars
«SCIENCE AND EDUCATION - 2017»



14th April 2017, Astana



**ҚАЗАҚСТАН РЕСПУБЛИКАСЫ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ
Л.Н. ГУМИЛЕВ АТЫНДАҒЫ ЕУРАЗИЯ ҰЛТТЫҚ УНИВЕРСИТЕТІ**

**«Ғылым және білім - 2017»
студенттер мен жас ғалымдардың
XII Халықаралық ғылыми конференциясының
БАЯНДАМАЛАР ЖИНАҒЫ**

**СБОРНИК МАТЕРИАЛОВ
XII Международной научной конференции
студентов и молодых ученых
«Наука и образование - 2017»**

**PROCEEDINGS
of the XII International Scientific Conference
for students and young scholars
«Science and education - 2017»**

2017 жыл 14 сәуір

Астана

УДК 378

ББК 74.58

Ғ 96

Ғ 96

«Ғылым және білім – 2017» студенттер мен жас ғалымдардың XII Халықаралық ғылыми конференциясы = The XII International Scientific Conference for students and young scholars «Science and education - 2017» = XII Международная научная конференция студентов и молодых ученых «Наука и образование - 2017». – Астана: <http://www.enu.kz/ru/nauka/nauka-i-obrazovanie/>, 2017. – 7466 стр. (қазақша, орысша, ағылшынша).

ISBN 978-9965-31-827-6

Жинаққа студенттердің, магистранттардың, докторанттардың және жас ғалымдардың жаратылыстану-техникалық және гуманитарлық ғылымдардың өзекті мәселелері бойынша баяндамалары енгізілген.

The proceedings are the papers of students, undergraduates, doctoral students and young researchers on topical issues of natural and technical sciences and humanities.

В сборник вошли доклады студентов, магистрантов, докторантов и молодых ученых по актуальным вопросам естественно-технических и гуманитарных наук.

УДК 378

ББК 74.58

ISBN 978-9965-31-827-6

©Л.Н. Гумилев атындағы Еуразия
ұлттық университеті, 2017

$$F = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \\ 2 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 3 & 3 & 4 & 4 & 5 & 4 & 4 & 5 \\ 3 & 4 & 6 & 4 & 5 & 5 & 6 & 6 & 4 & 5 & 5 & 6 & 6 & 5 & 6 & 6 \end{vmatrix} \quad (6)$$

Для определителя матрицы узловых проводимостей справедливо выражение [5]:

$$\det Y(p) = \frac{1}{Z_1 Z_2 Z_3} + \frac{1}{Z_1 Z_2 Z_4} + \frac{1}{Z_1 Z_2 Z_6} + \frac{1}{Z_1 Z_3 Z_4} + \frac{1}{Z_1 Z_3 Z_5} + \frac{1}{Z_1 Z_4 Z_5} + \frac{1}{Z_1 Z_4 Z_6} + \frac{1}{Z_1 Z_5 Z_6} +$$

$$+ \frac{1}{Z_2 Z_3 Z_4} + \frac{1}{Z_2 Z_3 Z_5} + \frac{1}{Z_2 Z_4 Z_5} + \frac{1}{Z_2 Z_4 Z_6} + \frac{1}{Z_2 Z_5 Z_6} + \frac{1}{Z_3 Z_4 Z_5} + \frac{1}{Z_3 Z_4 Z_6} + \frac{1}{Z_3 Z_5 Z_6} = 0 \quad (7)$$

Не трудно убедиться в том, что условие равенства нулю определителя матрицы узловых проводимостей выполняется при условии:

$$D(p) = Z_4 Z_5 Z_6 + Z_3 Z_5 Z_6 + Z_3 Z_4 Z_5 + Z_2 Z_5 Z_6 + Z_2 Z_4 Z_6 + Z_2 Z_3 Z_6 + Z_2 Z_3 Z_5 + Z_2 Z_3 Z_4 +$$

$$Z_1 Z_5 Z_6 + Z_1 Z_4 Z_6 + Z_1 Z_3 Z_6 + Z_1 Z_3 Z_5 + Z_1 Z_3 Z_4 + Z_1 Z_2 Z_6 + Z_1 Z_2 Z_5 + Z_1 Z_2 Z_4 = 0 \quad (8)$$

Подставляя исходных данных в топологическое выражение (8) и после несложных преобразований получим искомое характеристическое уравнение:

$$D(p) = (R_1 R_2 + R_1 R_3 + R_2 R_3 + R_1 R_6 + R_2 R_6) L C p^2 + ((R_2 + R_3 + R_6) L + (R_1 R_2 R_3 + R_1 R_6 R_3 +$$

$$R_2 R_6 R_3) C) p + R_1 R_2 + R_1 R_3 + R_1 R_6 + R_2 R_6 + R_3 R_6 = 0 \quad (9)$$

Список использованных источников

1. Бессонов Л. А. Теоретические основы электротехники. М.: Высшая школа, 1973. 752 с.
2. Мельников Н.А. Матричный метод анализа электрических цепей. М.: Энергия, 1972. 232с.
2. 3 Ионкин П.А., Соколов А.А. Топологический анализ электрических цепей // Электричество. 1964.- №4. - С. 59-66.
3. 4 Гераскин О.Т. Топологическое содержание узлового и контурного определителей электрической сети и расчет их величин при помощи ЦВМ // Известия АН СССР. Энергетика и транспорт. 1966.- №2. - С. 59-70.
4. Ахметбаев Д.С. Использование топологического метода к анализу стационарных режимов сложных электрических сетей // Электрооборудование: эксплуатация и ремонт. Москва. 2010., №9, С.19-26.

УДК 378.72

ЯЗЫК ПРОГРАММИРОВАНИЯ СИ++: ЭТАПЫ ЭВОЛЮЦИИ И СОВРЕМЕННОЕ СОСТОЯНИЕ

Токашева Мейрамгуль

ЕНУ им. Л.Н.Гумилева, Астана, Казахстан

Ерболова Асель Серикановна

Казахский национальный университет имени аль-Фараби, Алматы, Казахстан

Первые версии языка программирования Си++ (тогда он назывался "Си с классами") были разработаны в начале 80-х годов Бьярном Страуструпом, сотрудником знаменитой AT&T Bell Labs, где ранее были разработаны такие шедевры программирования, как операционная система UNIX и язык программирования Си.

По признанию самого автора языка, Си++ никогда не разрабатывался на бумаге. Проектирование, реализация и документирование новых возможностей происходили

фактически одновременно. Единственной целью разработки было создание языка, на котором было бы удобно программировать автору и его друзьям. Если вспомнить историю создания языка Си, то прослеживаются явные аналогии. За основу был взят популярный в среде профессиональных разработчиков язык программирования Си. Первыми средствами, которыми был расширен Си, стали средства поддержки абстракций данных и объектно-ориентированного программирования.

Как это принято в AT&T, описание нового языка не было опубликовано сразу. Первыми его пользователями стали сами сотрудники Bell Labs. В 1993 впервые была реализована коммерческий транслятор, и сам язык был назван "C++", что можно (имея в виду операцию инкрементирования языка Си) трактовать как увеличенный или расширенный язык Си.

Первым транслятором языка был препроцессор cfront, транслирующий программу на Си++ в эквивалентную программу на Си. И только в конце 80-х годов были реализованы прямые трансляторы, не использующие Си в качестве промежуточного языка. Пионером среди таких трансляторов стал GNU CC.

Если не считать документацию к транслятору cfront, первой книгой с описанием языка стала "The C++ Programming Language" (Addison-Wesley, 1985), переведенная на русский язык и изданная в 1991 году (Страуструп Б. Язык программирования C++. М.: Радио и Связь, 1991). С этого момента началось его бурное распространение и создание многочисленных реализаций.

Модель реализации ООП была частично позаимствована из языка программирования Simula67 и ориентировалась в основном на возможность эффективной реализации на вычислительных машинах со стандартной архитектурой. Некоторые возможности языка Simula были отклонены, так как, по мнению автора Си++, подталкивали разработчика к плохому стилю программирования. Так, в первых версиях Си++ полностью отсутствовала возможность динамической идентификации типа объекта (run-time type identification, rtti). Основные концепции поддержки ООП в Си++ были изложены Страуструпом в статье "What is Object Oriented Programming".

С 1985 года в язык были введены новые возможности: множественное и виртуальное наследование, шаблоны функций и классов, обработка исключительных ситуаций. Кардинально изменена семантика совместного использования оператора new, изменен синтаксис для вложенных классов.

С момента опубликования и до настоящего момента язык постоянно совершенствовался и расширялся. Важным этапом в его развитии стала публикация в 1990 году подробного и достаточно строгого описания языка [1]. Сокращенно эту книгу часто называют ARM. Фактически одновременно с этим началась стандартизация языка.

Инициатором стандартизации выступил не автор языка. Более того, Страуструп всегда довольно прохладно относился к попытке его полной стандартизации и выступал за реализации, в которых базовые возможности языка расширялись бы средствами и библиотеками, характерными только для данной реализации.

История стандартизации

Объединенный ANSI-ISO (ANSI X3J16; ISO WG21/N0836) комитет начал функционировать в конце 1989 года. Целью его работы является создание единого стандарта для языка Си++ и его библиотечных средств. За основу проекта стандарта было взято описание языка, данное в ARM, и книга [2].

В работе объединенного комитета, функционирующего и по сей день, значительное место занимает изучение (с последующим принятием или отказом от) возможных изменений текста проекта стандарта, а также уточнение различных правил языка. Позволим себе напомнить, что непосредственный предшественник Си++ - язык Си прошел успешно процесс стандартизации. Работа по его стандартизации завершилась в 1989 году, и стандартизованный вариант сейчас известен под именем ANSI Си.

Работа по стандартизации Си++ осложнялась тем, что язык долгое время был открыт

для расширений. Сами формулировки правил ARM были недостаточно строги и часто требовали уточнения. Си++ стал довольно громоздким языком (сопоставимым разве что с языком Ada), и ни один человек сейчас не в состоянии точно помнить все его детали и тонкости. С момента начала стандартизации несколько изменилась и сама идеология Си++.

Изначально автор отвергал возможность использования в языке средств динамического определения типов (rtti), однако в текущем проекте стандарта такие средства имеются. Аналогично, в Си++, описанном в ARM, есть довольно жесткие ограничения на возможность определения виртуальных функций, которые сейчас ослаблены. Характерно, что некоторые изменения, требующие пересмотра самой идеологии языка, вносились самим Страуструпом [4]. Изначально планировалось, что окончательная редакция проекта стандарта будет опубликована в 1994 году. Эти сроки были безнадежно провалены. Можно сказать, что последние годы процесс стандартизации постоянно находится в состоянии "2 года до завершения". Так, согласно текущему расписанию международный стандарт Си++ должен быть опубликован в конце 1998 года. Даже теперешние, на редкость подробные и громоздкие формулировки семантических правил и ограничений языка явно не дотягивают до математической строгости и оставляют простор для различных трактовок.

В ранних версиях проекта стандарта не было раздела, описывающего стандартные библиотеки. Не было описаний библиотеки и в ARM. В то же время реализация библиотеки потокового ввода/вывода, предложенная Andrew Koenig, была повторена в нескольких реализациях и стала стандартом "де-факто". В 1993-1994 годах в проекте стандарта было введено около семи новых разделов для описания библиотеки.

Принципиально важным событием в истории развития стандарта стандартной библиотеки стало включение библиотеки STL (Standard Template Library) разработанной нашим бывшим соотечественником, сотрудником Hewlett-Packard Александром Степановым. В своей статье об истории STL он упоминает, что изначально стремился использовать в Си++ только возможности шаблонов, аналогичные родовым (generic) пакетам и процедурам языка Ada, но после обсуждений со Страуструпом существующих возможностей Си++ изменил свое мнение. Комитет по стандартизации пошел навстречу этим двум гуру, вплоть до того, что в семантику шаблонов были внесены изменения. Этим был создан интересный прецедент в истории языков программирования: не библиотека написана для языка, а сам язык претерпел изменения под влиянием библиотеки, причем разработанной не автором языка.

Согласно расписанию работы комитета по стандартизации, проект стандарта принятый в апреле 1995 года, был предложен для публичного обсуждения и сделан доступен пользователям Internet. Утверждается, что с этого момента никаких серьезных изменений в текст стандарта вноситься не будет. Предполагалось, что в сентябре 1996 года новая редакция проекта стандарта будет вынесена на публичное обсуждение, но не так давно этот этап был перенесен на конец года.

Современное состояние языка

Приведем описание новых возможностей языка, введенных с момента публикации ARM и до публикации проекта стандарта.

1. Ведены новые ключевые слова-синонимы для операций (and, and_eq, bitand, bitor, compl, not, or, or_eq, xor, xor_eq).

2. В языке появился булевский тип данных bool и литералы этого типа true и false.

3. Появился механизм пространств имен (namespace), облегчающий совместную реализацию проектов группами программистов и позволяющий избегать проблем при использовании нескольких библиотек (ключевые слова namespace и using).

4. Новое ключевое слово explicit позволяет запретить нежелательное использование конструкторов как функций преобразования типов.

5. Изменены синтаксис и семантика для изменения прав доступа к членам классов. Новый механизм позволяет использовать единый синтаксис для использования членов пространств имен и членов классов. При этом несколько изменились правила выбора

наиболее подходящей из набора совместно используемых функций (на основе использования ключевого слова `using`).

6. Добавлен механизм явного использования `rtti` (включающий операцию с ключевым словом `type id` и класс `type_info` стандартной библиотеки).

7. Добавлены новые и скорректированы старые способы явного преобразования типов (`static_cast`, `dynamic_cast`, `const_cast` и `reinterpret_cast`).

8. Объявления переменных теперь возможны не только в заголовке `for`-цикла, но и в операторах `if`, `while`, `do-while`, `switch`.

9. Более точно определено время жизни временных объектов в выражении. Теперь время их жизни ограничено полным выражением, а не концом текущего блока, как сказано в ARM.

10. Полностью переработано определение шаблонов в Си++. Теперь уже нельзя сказать, что шаблоны Си++ являются лишь слегка замаскированными синтаксическими подстановками. Для них обязателен синтаксический разбор и контроль семантики (в максимально возможной степени). Неоднозначности внутри тел шаблонов, вызываемые неизвестной природой типовых параметров, явно разрешаются посредством ключевого слова `typename`.

11. Допускаются шаблонные функции-члены нешаблонных классов, вложенные шаблонные классы и шаблоны - параметры шаблонов.

12. Виртуальные функции могут возвращать тип, отличный от типа подменяемой функции базового класса, если эти типы являются указателями или ссылками на производный и базовый класс.

13. Перечислимый тип (`enum`) окончательно утвердился как самостоятельный тип, не являющийся ни одним из целочисленных типов. Теперь разрешено совместное использование функций, основанное на этом различии; константа 0 перечислимого типа более не считается целочисленным 0, запрещено ее неявное преобразование к указательному типу.

14. Ослаблено ограничение на тип, возвращаемый операцией `->`. Теперь это может быть практически произвольный тип.

15. Добавлено (на редкость бессмысленное) ключевое слово `mutable`, позволяющее допускать изменение членов объекта константного класса. Более подробно некоторые из этих изменений рассмотрены в статье [3].

Перспективы

Си++ допускает использование в нескольких вариантах. Во-первых, он содержит язык программирования Си, точнее, его стандартизированный диалект ANSI C почти целиком. При этом Си++ обладает более мощными и строгими правилами проверки типов. Программист волен использовать лишь часть возможностей Си++, не сталкиваясь при этом с неприятностями, связанными с неполным знанием языка. Можно использовать только механизм классов и наследования, не пользуясь возможностями обобщенного (`generic`) программирования, основанного на шаблонах. Наконец, многим разработчикам приглянулась возможность обработки исключительных ситуаций, не зависящая от других механизмов языка. Взятые вместе, эти возможности предлагают разработчику современный, достаточно гибкий и мощный язык программирования высокого уровня с поддержкой ООП.

Список использованных источников

1. The Annotated C++ Referen Manual, Margaret A. Ellis and Bjarne Stroustrup, 1990, 453 pages. (Русский перевод М. Эллис, Б. Струоструп. Справочное руководство по языку программирования C++ с комментариями. М.: Мир. 1992)
2. P.J. Plauger, The Draft Standard C++ Library; Prentice-Hall, 1995.
3. Е Зуев, А Кротов "Новые возможности языка Си++", PC Magazine/Russian Edition, #7, 1994.
4. Bjarne Stroustrup "New Casts Revisited", ANSI Doc X3J16/93-142.