

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова**

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ISSN 2411-1473

**Современные
информационные технологии
И
ИТ-образование**

Научный журнал

Том 2 (№ 11)

**Москва
2015**

УДК [004:377/378](063)
ББК 74.5(0)я431+74.6(0)я431+32.81(0)я431
С 56

**Современные информационные технологии и ИТ-образование. Т. 2 (№ 11),
2015. - 614 с. (ISSN 2411-1473)**

В данном выпуске журнала представлены доклады X Юбилейной международной научно-практической конференции «Современные информационные технологии и ИТ-образование», прошедшей в Московском государственном университете имени М.В. Ломоносова 20-22 ноября 2015 года.

Журнал «Современные информационные технологии и ИТ-образование» включен в наукометрическую базу «Российский индекс научного цитирования» с размещением полнотекстовых версий в научной электронной библиотеке eLIBRARY.RU. URL: http://elibrary.ru/title_about.asp?id=52785



*Издание осуществлено при финансовой поддержке
Российского фонда фундаментальных исследований
(Грант РФФИ № 15-07-20760_з)*

Учредитель:

Фонд содействия развитию интернет-медиа, ИТ-образования, человеческого потенциала «Лига интернет-медиа»

Издатель:

Фонд содействия развитию интернет-медиа, ИТ-образования, человеческого потенциала «Лига интернет-медиа»

Адрес редакции:

119991, г. Москва, ГСП-1, Ленинские горы, д. 1, стр. 52, факультет ВМК МГУ имени М.В. Ломоносова, каб. 375. E-mail: sukhomlin@mail.ru, тел./факс (495) 939-46-26.

Журнал зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций (Роскомнадзор).

Свидетельство о регистрации средства массовой информации ПИ № ФС77-61433 от 10 апреля 2015 г.

Издается с 2005 года. Выходит 1 раз в год.

Редакционная коллегия журнала:

Главный редактор:

Сухомлин В.А. - доктор технических наук, профессор, заведующий лабораторией ОИТ факультета ВМК МГУ имени М.В. Ломоносова, Президент Фонда «Лига интернет-медиа»;

Члены редакционной коллегии:

Веремей Е.И. - доктор физ.-мат. наук, профессор, СПбГУ;

Гергель В.П. - доктор физ.-мат. наук, профессор, ННГУ им. Н.И. Лобачевского;

Самуйлов К.Е. - доктор физ.-мат. наук, профессор, РУДН;

Калиниченко Л.А. - доктор физ.-мат. наук, профессор, вед. н.с. ИПИ РАН ФИЦ ИУ РАН;

Лугачев М.И. - доктор экономических наук, профессор, МГУ имени М.В. Ломоносова;

Любецкий В.А. - доктор физ.-мат. наук, профессор, ИППИ РАН им. А.А. Харкевича;

Нечаев В. В. - доктор технических наук, профессор, МИРЭА;

Посыпкин М.А. - доктор физ.-мат. наук, вед. н. с. ИППИ РАН им. А.А. Харкевича;

Язенин А.В. - доктор физ.-мат. наук, декан факультета ПМиК, профессор, ТвГУ;

Намиот Д.Е. - кандидат физ.-мат. наук, с.н.с. факультета ВМК МГУ имени М.В. Ломоносова;

Зубарева Е.В. - кандидат пед. наук, доцент, н.с. факультета ВМК МГУ имени М.В. Ломоносова;

Сотникова М.В. - кандидат физ.-мат. наук, доцент СПбГУ.

Статьи, поступающие в редакцию, рецензируются. За достоверность сведений, изложенных в статьях, ответственность несут авторы публикаций. Мнение редакции может не совпадать с мнением авторов материалов. При перепечатке ссылка на журнал обязательна.

Материалы публикуются в авторской редакции. При перепечатке и цитировании материалов ссылка на журнал «Современные информационные технологии и ИТ-образование» обязательна.

Содержание

ИССЛЕДОВАНИЯ И РАЗРАБОТКИ В ОБЛАСТИ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ИХ ПРИЛОЖЕНИЙ.....	8
<i>Manfred Sneps-Sneppe</i> Telecommunications and sensors in robotic war.....	9
<i>Pyushin E., Namiot D.</i> Javascript memory management.....	20
<i>Намиот Д.Е.</i> Интерфейсы прикладного уровня в SDN.....	26
<i>Вахов А.Н., Зотова Е.А., Коломоец И.В., Рыжов А.П., Шварц А.Ю.</i> Платформа Uchi.ru: опыт разработки и перспективы развития.....	31
<i>Денисов В.С.</i> Управление настройками Java-приложений с использованием механизма аннотаций.....	37
<i>Денисов В.С.</i> Функциональные требования к библиотеке управления настройками Java-приложений.....	41
<i>Леонов М.В., Киселева Е.А.</i> Опыт программирования электронных справочников в условиях «планшетной революции»	45
<i>Игнатенков А.В., Ольшанский А.М.</i> Применение искусственной нейронной сети для построения расписаний процессов на при- мере графика движения поездов.....	50
<i>Ромасевич Е.П.</i> Проблемы миграции современных сетей на протокол IPv6.....	56
<i>Ромасевич П.В.</i> Оценка общего объема памяти ввода-вывода телекоммуникационной системы в условиях самоподобного трафика и потери пакетов.....	61
<i>Емельченков Е.П., Макаров А.И., Мунерман В.И.</i> Объектно-ориентированный подход к представлению баз данных в не первой нормальной форме.....	66
<i>Копытов В.В., Шульгин А.О., Федоров С.А.</i> Разработка архитектуры интеграционной среды кроссплатформенных мобильных прило- жений с корпоративной информационной системой.....	71
<i>Кейно П.П.</i> Предпосылки формирования новой методологии разработки веб-узлов BlockSet и деклара- тивного языка VML.....	78
<i>Гарфутдинова А.Р., Макаровских Т.А.</i> Автоматизированная система для организации и проведения конференции.....	85
<i>Мунерман В.И., Симакова А.А.</i> Алгебраический подход к формализации семантики задач.....	92
<i>Орлова Н.В.</i> Об интеграции технологий дополненной реальности, 3D-туров и 3D-моделирования для обучения IT-специалистов.....	98
<i>Туракулова А.И.</i> Возможности нового объектно-ориентированного языка программирования ObjectScript	105
<i>Актаева А.У., Баймуратов О.А., Галиева Н.Г., Илипбаева Л.Б., Искоджаева И.</i> Квантовая информатика: безопасность информации.....	109
<i>Ярмухаметов Ф.Ф., Кейно П.П.</i> Практика использования веб-шаблонизаторов в программном комплексе интерпретатора на примере СТТР2.....	117
<i>Пивнева С.В., Купцов Н.А.</i> Математическое моделирование процесса обучения и самообучения на основе мультиэври- стического подхода.....	121
<i>Макашов П.Л., Романенко Н.А.</i> Сервис-ориентированный подход к управлению IT проектами на примере использования программного продукта «Jira».....	127

<i>Ощурков В.А., Макашова В.Н.</i>	
Оперативное планирование производства в MES-системах с использованием методов и алгоритмов искусственного интеллекта.....	133
<i>Сапегина В.С., Куренев А.В.</i>	
Создание эффективной системы лояльности клиентов посредством внедрения платформы TradeLine.....	140
<i>Соколова А.А.</i>	
Разработка системы управления процессом создания программного продукта.....	146
ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ	150
<i>Головешкин В.А., Жукова Г.Н., Ульянов М.В., Фомичев М.И.</i>	
Сравнение ресурсных характеристик традиционного и модифицированного метода ветвей и границ для TSP.....	151
<i>Абаев П.О., Бесчастный В.А., Гайдамака Ю.В.</i>	
О применении пространственных точечных процессов в решении оптимизационных задач для беспроводных сетей с установлением прямых соединений.....	160
<i>Абаев П.О., Хачко А.В., Бесчастный В.А.</i>	
О задаче оптимизации работы сервера установления соединения с механизмом локального контроля перегрузки.....	166
<i>Самуйлов К.Е., Гайдамака Ю.В., Сопин Э.С., Горбунова А.В.</i>	
Алгоритм вычисления преобразования Лапласа-Стилтьеса для времени отклика системы облачных вычислений с гистерезисным управлением.....	172
<i>Самуйлов К.Е., Гудкова И.А., Острикова Д.Ю.</i>	
Анализ вероятности прерывания обслуживания в модели сети LTE с совместным использованием ресурсов.....	178
<i>Вихрова О.Г., Сопин Э.С.</i>	
Анализ показателей качества сети LTE с помощью систем массового обслуживания с ограниченным ресурсом и случайными требованиями.....	185
<i>Медведева Е.Г., Гайдамака Ю.В.</i>	
К анализу параметров качества передачи мультимедийного потокового трафика в одноранговой сети.....	192
<i>Мокров Е.В., Гудкова И.А.</i>	
Модель для расчета уровня снижения мощности в сети 3GPP LTE с совместным использованием частот телеметрии аэропорта.....	199
<i>Белякова Ю. В., Михалкович С. С.</i>	
Концепт-параметры как механизм развития средств обобщенного программирования в языке C#.....	205
<i>Исаченко А.Н., Исаченко Я.А.</i>	
О некоторых характеристиках матроидов и свойствах гамильтоновых матроидов.....	214
<i>Козлов С.В.</i>	
Использование соответствия Галуа как инварианта отбора контента при проектировании информационных систем.....	220
<i>Пеленицын А.М.</i>	
Концепты C++17 в их отношении к концептам C++0x.....	226
<i>Рамиль Альварес Х., Владимиров Ю.С.</i>	
Об извлечении квадратного корня в тричной симметричной системе.....	233
<i>Рамиль Альварес Х.</i>	
Алгоритм извлечения квадратного корня из больших целых чисел.....	239
<i>Фёдоров И.Г.</i>	
Синтаксис и семантика исполняемых моделей бизнес-процессов.....	242
<i>Афанасьевский Л.Б., Горин А.Н., Фадин А.Г.</i>	
Реализация аналитической и имитационной моделей системы массового обслуживания..	253
НАУЧНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ И НАУКЕ	260
<i>Лазовская Т.В., Тархов Д.А.</i>	
Использование асимптотических решений при построении нейросетевой модели в жесткой задаче.....	261
<i>Васильев А.Н., Тархов Д.А., Шемякина Т.А.</i>	

Многоуровневые модели окружающей среды в мегаполисах.....	267
<i>Васильев А.Н., Тархов Д.А., Шемякина Т.А.</i>	
Модель неизотермического химического реактора на основе параметрических нейронных сетей. Гибридный метод.....	271
<i>Васильев А.Н., Тархов Д.А., Шемякина Т.А.</i>	
Мезо-уровневая нейросетевая модель загрязнения атмосферного воздуха Санкт-Петербурга по данным мониторинга.....	279
<i>Зыкина А.В., Запорожец Д.Н.</i>	
Пакет прикладных программ для моделирования и решения процессов с использованием аппарата вариационных неравенств.....	284
<i>Бабичева Т.С.</i>	
Транспортные потоки: математическое и имитационное моделирование	290
<i>Идрисова Д.И., Каверзнева Т.Т., Тархов Д.А., Лазовская Т.В.</i>	
Моделирование распределения опасного вещества в тупиковом тоннеле с использованием нейросетевого подхода.....	297
<i>Пишкинас А.О., Оносов И.А., Корчагин С.А., Романчук С.П., Терин Д.В.</i>	
Разработка программных средств моделирования композитных наноматериалов.....	301
<i>Семенов М.Г.</i>	
Модель Марковица: математические аспекты и компьютерная реализация.....	306
<i>Сенчилов В.В.</i>	
О решении одной видоизмененной краевой задачи типа Неймана в классах метааналитических в круге функций с применением системы компьютерной математики Maple.....	310
<i>Скрипачев В.О., Пирхавка А.П., Полушковский Ю.А., Суворцева И.В., Жуков А.О., Яковлев О.В.</i>	
Аспекты создания информационной системы для обработки ионосферных данных.....	316
<i>Юмагулов М.Г., Беликова О.Н., Исанбаева Н.Р.</i>	
Моделирование областей устойчивости точек либрации ограниченной задачи трех тел с помощью систем компьютерной математики.....	321
<i>Лесников С.В., Булыгина Д.С., Лесников А.В., Лесников Г.С.</i>	
Конструирование гипертекстового информационно-поискового тезауруса метаязыка лингвистики.....	326
<i>Понятский В.М., Егоров Д.Б.</i>	
Программный комплекс моделирования последовательности методов видеообработки для задач управления.....	334
<i>Тархов Д.А., Симакина А.А., Суднева А.И.</i>	
Обработка данных методом треугольных приближений.....	341
<i>Тарасенко Ф.Д., Тархов Д.А.</i>	
Сравнительный анализ применения различных базисных функций в алгоритмах последовательного сглаживания данных.....	350
<i>Тархов Д.А., Шаньшин И.К., Шаханов Д.О.</i>	
Сравнение нейросетевого и классического подхода к задаче идентификации миграционных процессов.....	355
<i>Васильев А.Н., Кузнецов Е.Б., Леонов С.С.</i>	
Нейросетевой метод идентификации и анализа модели деформирования металлических конструкций в условиях ползучести.....	360
<i>Бурцев А.А., Сидоров С.А.</i>	
Программный комплекс ДССП-ТВМ для структурированного программирования троичной [виртуальной] машины.....	371
ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ ПРОГРАММИРОВАНИЕ, ГРИД-ТЕХНОЛОГИИ, ПРОГРАММИРОВАНИЕ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ.....	380
<i>Курочкин И.И., Гуменный Д.Г.</i>	
Безопасность сетей SDN. Классификация атак.....	381
<i>Захаров В.Н., Мунерман В.И.</i>	
Параллельный алгоритм умножения многомерных матриц.....	384
<i>Мунерман В.И., Мунерман Д.В.</i>	

Алгебраический подход к построению программно-аппаратных комплексов для повышения эффективности массовой обработки данных.....	391
<i>Орлов Ю.В., Посыпкин М.А.</i>	
Среда комплексного анализа производительности алгоритмов балансировки в параллельном методе ветвей и границ.....	397
<i>Тархов Д. А., Радченко Д.С.</i>	
Распределённые алгоритмы оптимизации.....	404
<i>Артамонов Ю.С., Востокин С.В.</i>	
Применение облачного сервиса Templet Web при проведении лабораторных практикумов на суперкомпьютере «Сергей Королев».....	409
<i>Ефимов А.И.</i>	
Виртуалтрединг: новая архитектура для вычислительных систем с прямым тонко гранулированным аппаратным мультипрограммированием.....	415
<i>Голубева Я.В.</i>	
Разработка и исследование алгоритмов балансировки нагрузки в параллельной реализации метода ветвей и границ.....	423
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СИСТЕМАХ УПРАВЛЕНИЯ	430
<i>Веремей Е.И.</i>	
Математические модели и методы в цифровых технологиях управления движением морских судов.....	431
<i>Веремей Е.И., Сотникова М.В.</i>	
Алгоритмы оптимизации движения по заданной траектории с учетом прогноза погодных условий.....	440
<i>Жгун Т.В., Липатов А.В., Лемешова Д.Д.</i>	
Применение метода главных компонент для построения объективных показателей изменения качества системы.....	446
<i>Понятский В.М., Грубенко А.Д., Кислинский И.В., Федорищева В.Г.</i>	
Использование ИТ на основе ПК Matlab для моделирования канала обмена CAN для систем управления.....	456
<i>Понятский В.М., Федорищева В.Г., Богданова Л.А., Игумнова Т.</i>	
ИТ на основе Polyspace Matlab для отладки программного обеспечения встроенных микропроцессоров систем управления	462
<i>Понятский В.М., Яковлев А.Е., Корзунов О.В., Лужинский А.И., Кислинский И.В.</i>	
Использование ИТ при командной организации разработки модели системы управления	474
<i>Смирнова М.А., Смирнов М.Н.</i>	
Астатическая коррекция цифровых законов управления.....	481
<i>Смирнов М.Н., Смирнова М.А.</i>	
Вопросы синтеза цифровых законов управления морскими судами с учетом неопределенностей.....	487
<i>Жук А.П., Бурмистров В.А., Гавришев А.А.</i>	
Система передачи информации с использованием стохастических ортогональных ансамблей дискретных многоуровневых сигналов.....	493
<i>Фараонов А.В.</i>	
Принятие решений в условиях неопределенности как способ подготовки специалистов транспортной логистики.....	499
<i>Джамбеков А.М.</i>	
Разработка системы управления процессом каталитического риформинга на базе нечеткой логики.....	506
<i>Коваленченко А.С.</i>	
К вопросу об оценке эффективности функционирования беспроводных сенсорных сетей.	513
<i>Пусная О.П., Зайцева Т.В., Лихонина А.В.</i>	
Автоматизированная система управления спортивными достижениями.....	517
<i>Губарев А.В.</i>	
Модель угроз системы передачи данных между управляющим программным обеспечением и аппаратным средством.....	524
<i>Марценюк М.А., Селетков И.П.</i>	

Использование нечёткого клеточного автомата для моделирования динамических процессов в средах с памятью.....	532
<i>Селезнева О.В.</i>	
Сравнение методов прогнозирования траектории морских судов.....	541
<i>Плужник Е.В., Никульчев Е.В., Лукьянчиков О.В., Ковшов Е.Е.</i>	
Программное обеспечение для управления распределенными приложениями в облачных вычислительных инфраструктурах.....	547
<i>Беляков Д.В.</i>	
Задача об автоколебаниях пластинки в потоке среды.....	552
<i>Коваленко С.Ю., Кондаков С.А.</i>	
Комплексы для проведения лабораторных работ.....	556
<i>Павлов А.В.</i>	
Фильтрация и прогноз смещенных нестационарных последовательностей.....	560
БОЛЬШИЕ ДАННЫЕ И ИХ ПРИЛОЖЕНИЯ.....	564
<i>Королев С.А., Селиверстов А.В., Зверков О.А., Любецкий В.А.</i>	
Классическая аттенюаторная регуляция, зависящая от концентрации триптофана, у актинобактерий.....	565
<i>Любецкий В.А., Селиверстов А.В.</i>	
О решении одной NP-полной задачи.....	569
<i>Зверков О.А., Селиверстов А.В., Любецкий В.А.</i>	
О транскрипционных факторах, кодируемых в пластидах родофитной ветви.....	571
<i>Гагарин А.П.</i>	
Оценка структурной сложности некоторых типовых объектных моделей данных в молекулярной генетике	576
<i>Рубанов Л.И., Селиверстов А.В., Зверков О.А., Любецкий В.А.</i>	
Ультраконсервативные элементы у простейших из надтипа Alveolata.....	581
<i>Рубанов Л.И., Селиверстов А.В., Любецкий В.А.</i>	
Широкомасштабный поиск ультраконсервативных элементов в полных геномах.....	586
<i>Истомина С.Н.</i>	
Сравнительный анализ связанных рядов: длин ортологичных белков и их приращений....	594
<i>Горбунов К.Ю., Любецкий В.А.</i>	
Реконструкция предковых хромосомных структур.....	600
<i>Королев С.А., Селиверстов А.В., Любецкий В.А.</i>	
О трансляции рибосомного белка L16 в пластидах цветковых растений.....	606
<i>Фролов А.С., Семенов А.С.</i>	
Использование Charm++ для решения графовых задач в масштабах эксафлопсных вычислений.....	608

Пеленицын А.М.

Южный федеральный университет, Институт математики, механики и компьютерных наук
им. И.И. Воровича, ассистент кафедры информатики и вычислительного эксперимента,
apel@sfedu.ru

КОНЦЕПТЫ C++17 В ИХ ОТНОШЕНИИ К КОНЦЕПТАМ C++0X

КЛЮЧЕВЫЕ СЛОВА

Обобщённое программирование, концепты C++, шаблоны C++, контроль типов

АННОТАЦИЯ

В статье проводится анализ текущего проекта концептов для C++ 2017 на основе выявления проблем концептов C++0x, которые планировалось включить в стандарт языка 2011 года. Показано, что неудача концептов C++0x напрямую повлияла на конкретные проектные решения, вошедшие в новый дизайн концептов.

Неограниченный параметрический полиморфизм, как он представлен в механизме шаблонов C++98, имеет ряд известных недостатков, главным из которых является невозможность генерации разборчивых сообщений об ошибках. На протяжении 2000-х и 2010-х рассматриваются средства введения ограниченного параметрического полиморфизма в язык C++. Мы исследуем наиболее актуальное из таких предложений — способ определения и применения ограничений на параметры шаблонов C++, так называемые концепты C++1z (также: Concepts Lite), планируемые к включению в стандарт языка 2017 года.

Все рассмотренные нами источники по дизайну концептов C++1z имеют общий недостаток, а именно, в описании нового механизма практически не говорится о том, какие факторы повлияли на принятие конкретных проектных решений. На наш взгляд, важнейшие из этих решений продиктованы проблемами, возникшими в старом дизайне концептов C++0x. Анализ этой связи между двумя проектами позволяет лучше понять основные черты готовящихся изменений в языке C++. Именно этот анализ предлагается в нашей статье.

В рассмотрении проекта концептов C++1z мы используем ряд документов комитета по стандартизации C++: некоторые примеры взяты из неформального введения в тему [1]; детали синтаксиса уточнены по более свежему формальному документу-описанию изменений в текст стандарта языка [2]; изменения в стандартной библиотеке пока ещё не пересмотрены в свете последней редакции концептов, но некоторое представление о них можно получить по отчёту [3]. Мнение автора языка C++, Бьярне Страуструпа приводится в основном по статье [4].

Статья состоит из пяти частей. В первой части приводится пример, который должен иллюстрировать основную проблему неограниченных шаблонов C++98 и преимущества, которые предполагается получить после введения концептов в новый стандарт языка. Во второй части приводится краткий обзор старого проекта концептов C++0x и основные причины его неудачи — невключения в стандарт 2011 года. В третьей части описываются главные черты нового проекта концептов и их связь с проблемами старого проекта. В четвёртой части рассматривается один из важных элементов ограниченного параметрического полиморфизма: перегрузка на основе концептов, и то, как авторам нового проекта удалось реализовать её в условиях отсутствия явных моделей концептов. В пятой части даны замечания по текущему состоянию нового проекта — деталям синтаксиса и доступности его реализации.

1. Пример и мотивация

Одна из главных проблем шаблонов C++98 состоит в длинных ошибках компиляции. Происходят они от того, что несоответствие типа неявным требованиям выясняется посреди процесса подстановки этого типа в определение шаблона. В результате текст ошибки ссылается на детали реализации шаблона, которым не смог удовлетворить тип-аргумент. Однако клиент шаблона, как правило, ничего не знает о его реализации.

Типичный пример такого несоответствия можно составить с помощью распространённой ошибки, которая состоит в попытке отсортировать список общим алгоритмом. В стандартной библиотеке C++ общий алгоритм сортировки подразумевает произвольный доступ к элементам сортируемого контейнера, в то время как тип списка такой доступ не предоставляет. Приведём

пример заголовка шаблона функции сортировки, который несколько проще стандартного алгоритма и принимает не пару итераторов, а контейнер целиком — в трёх вариантах (в своей реализации такой шаблон может, к примеру, вызывать стандартный алгоритм `std::sort`).

Листинг 1: Функция сортировки контейнера: неограниченный полиморфизм

```
template<typename Cont>
void sort(Cont & c);
```

В листинге 1 ничто не указывает на предположения о типе-параметре кроме его имени — типичная ситуация для неограниченного полиморфизма.

Листинг 2: Функция сортировки контейнера: ограниченный полиморфизм, полная и сокращённая версии

```
template<typename Cont>
  requires Sortable<Cont>()
void sort(Cont & c);

template<Sortable Cont>
void sort(Cont & c);
```

В листинге 2 приведены две эквивалентных версии в синтаксисе концептов, предлагаемом для C++17, причём вторая, сокращённая, совпадает также и с вариантом, предлагавшимся для концептов C++0x. В ней видно использование явного ограничения `Sortable` — это и есть именованный набор требований к шаблонному параметру `Cont`, который называется *концептом*. В данном случае набор требований должен обеспечивать произвольный доступ к элементам контейнера и наличие операции «меньше» для этих элементов.

Наиболее важным свойством концептов является то, что проверка соответствия аргумента заданным требованиям проводится до того, как начинается подстановка этого аргумента в тело шаблона. Таким образом, при возникновении несоответствий компилятор сообщит о них незамедлительно и в терминах ограничений, представленных в интерфейсе обобщённой компоненты, а не её реализации. Имеющиеся прототипы реализаций концептов C++0x и C++1z дают существенно более доступные сообщения об ошибках в подобных случаях, чем это было с неограниченными шаблонами C++98. Проиллюстрируем это на примере с сортировкой списка..

Сообщение об ошибке с вызовом стандартного алгоритма `sort` для стандартного списка в компиляторе GCC 4.9 занимает 67 строк, непосредственно текст ошибки плохо отражает существо проблемы, выглядит следующим образом:

```
no match for 'operator-' (operand types are 'std::_List_iterator<int>' and 'std::_List_iterator<int>')
```

Ошибка сопровождается большим количеством «замечаний» (англ. *note*), которые должны помочь программисту понять причину ошибки, но на деле имеет малую ценность. Аналогичная ошибка с концептами C++1z занимает около 6 строк, и выглядит следующим образом:

```
error: cannot call function 'void sort(const C&) [with C = std::_cxx11::list<int>]'
```

```
note: constraints not satisfied
```

```
void sort(C const &) {}
```

```
note: concept 'Sortable<std::_cxx11::list<int, std::allocator<int> >>' was not satisfied
```

В данном случае замечания совершенно точно отражают суть дела.

Концепты это новая сущность языка программирования, которая не укладывается в систему типов C++98. Это отличает механизм ограниченного параметрического полиморфизма, предлагаемый для C++, от ограниченного параметрического полиморфизма, основанного на подтипировании и характерного для чистых объектно-ориентированных языков, таких как Java и C#. (Отметим, что последний упомянутый подход неоднократно критиковался с предложением улучшений [5–7].) В связи с этим мы считаем важным всесторонний анализ дизайна концептов для C++ и предлагаем вариант такого анализа на основе сравнения текущего проекта концептов с проектом, предлагавшимся для включения в C++11. На наш взгляд основные черты нового проекта прямо вытекают из проблем, возникших в проекте концептов C++0x.

2. Концепты C++0x и их проблемы

Обсуждение концептов C++1z следует начать с анализа целей и причин неудачи концептов C++0x. Кратко перечислим основные элементы проекта концептов C++0x, которые одновременно являются достаточно общими требованиями к механизму ограниченного параметрического

полиморфизма.

1. Синтаксис для определения именованного набора ограничений — концепта.
2. Синтаксис для наложения ограничения (концепта) при объявлении шаблона класса или функции.
3. Способ проверки соответствия использования шаблонных параметров внутри шаблона класса или функции наложенным ограничениям.
4. Способ указания соответствия конкретных типов тем или иным концептам — так называемые *модели концептов* (concept map).

Оказалось, что основные сложности доставляли пункты 3 и 4. Первый из них требовал существенной модификации алгоритма поиска имён при обработке шаблонов компилятором. Такая модификация была выполнена в рамках проекта ConceptGCC [8], работала она весьма медленно и нестабильно. Мнение разработчиков этой версии состоит в том, что при необходимой доработке, проблем с производительностью компилятора быть не должно. Однако сам масштаб изменений, которые требовалось внести в распространённые компиляторы в отношении обработки шаблонов, негативно воспринимался членами комитета по стандартизации.

Второй проблемный пункт (4-й в списке выше) с моделями концептов неоднозначно воспринимался даже основным идеологом концептов C++0x, автором языка C++ Бьярне Страуструпом. Дело в том, что свидетельства соответствия конкретных типов некоторым концептам во многих случаях тривиальны и могут быть автоматически установлены компилятором. Однако такой автоматизм может плохо сочетаться с другими возможностями обновлённых шаблонов, как например, перегрузкой на основе концептов. Более того, в комитете по стандартизации высказывались мнения, что получение моделей концептов без участия, а значит, без явного намерения программиста просто неверно идеологически.

В целом, по состоянию на 2009 год проект концептов C++0x оценивался Б. Страуструпом как переусложнённый, рассчитанный на программистов-экспертов, в то время как изначальная идея состояла в создании механизма, облегчающего жизнь «среднему» программисту. Автор языка C++ предложил пять радикальных шагов по упрощению концептов, однако комитет по стандартизации не видел возможности обсуждать дополнительные предложения до принятия нового стандарта: сроки для внесения изменений подошли к концу. В итоге концепты были оставлены для будущих редакций стандарта языка.

3. Концепты C++1z в их отношении к концептам C++0x

Проект концептов для стандарта 2017 года был создан с нуля, но в соответствии с опытом, полученным при разработке концептов C++0x. Основные задачи, стоявшие перед авторами нового проекта, можно сформулировать следующим образом.

- Простота и понятность для «среднего» программиста.
- Отказ (возможно, временный) от средств, породивших наибольшие споры при обсуждении концептов C++0x.
- Учёт объективной сложности адаптации промышленных компиляторов.
- Обратная совместимость со стандартом языка 1998 года, в том числе, с неограниченными шаблонами (как и у концептов C++0x).

В этом разделе мы укажем две черты концептов C++1z, который вытекают из этих целей и отличают их от концептов C++0x: они связаны в первую очередь с двумя проблемными пунктами списка, приведённого в прошлом разделе. В следующих двух разделах обсуждаются другие черты концептов C++1z, важные с точки зрения обобщённого программирования.

Отсутствие контроля типов внутри шаблонов — возможно, самая неожиданная черта концептов C++1z, если смотреть на них без учёта изложенной выше предыстории. Контроль типов внутри шаблонов представлял потенциально особую сложность для адаптации промышленных компиляторов, как упоминалось в предыдущем разделе. Связано это, в частности, с тем, что новые ограниченные шаблоны должны были успешно сосуществовать и взаимодействовать со старыми, неограниченными шаблонами C++98. Если, к примеру, из шаблона первого типа вызывался шаблон второго, то проконтролировать корректность требований внешнего шаблона было достаточно сложно: фактически, задача свелась бы, как и прежде, поиску ошибок путём подстановки шаблонных аргументов. Обработка подобных сочетаний существенно усложняла компилятор, не принося никакого выигрыша по сравнению со старым подходом стандарта 1998 года.

Таким образом, в концептах C++1z было решено полностью исключить проверку соответствия требований, наложенных в заголовке шаблона, и определения этого шаблона. То

есть проверяется лишь соответствие аргументов шаблона в месте его использования, в то время как в теле шаблона параметры обрабатываются компилятором так же, как это было с шаблонами C++98. Данный подход считается разумным компромиссом в связи с тем, что автор обобщённой компоненты зачастую более квалифицированный специалист, чем её пользователь: он должен сформулировать набор ограничений к аргументам своей компоненты и написать её определение так, чтобы оно соответствовало этим требованиям, не рассчитывая на помощь компилятора в этой задаче. Далее, многочисленные использования однажды написанной компоненты будут уже проверяться именно компилятором.

Этим изменением объясняется различие двух проектов концептов с точки зрения модулярного контроля типов. Модулярный контроль типов означает, что если определение обобщённой компоненты было однажды принято компилятором, то при любой подстановке аргументов, удовлетворяющих ограничениям, повторная проверка определения не требуется. В концептах C++0x это условие удовлетворялось лишь частично по причине возможного использования неограниченных шаблонов внутри ограниченных. В концептах C++1z вначале проверяется соответствие аргументов требованиям, однако во время подстановки могут произойти ошибки в том случае, если автор шаблона неточно описал эти требования.

Полностью автоматическая проверка соответствия типов наложенным ограничениям или отказ от моделей концептов — вторая важная черта, отличающая концепты C++1z. Как упоминалось в прошлом разделе, механизм моделей концептов вызывал много вопросов, связанных с запретом или разрешением генерации таких моделей. В проекте для C++1z было решено максимально упростить использование концептов с помощью удаления самого понятия моделей концептов.

Таким образом, если тип не может быть однозначно и автоматически отображён на нужный набор требований, то он будет отвергнут компилятором. Имеющийся способ решения этой проблемы — модифицировать рассматриваемый тип нужным образом, зачастую не представляет трудностей, но разрушает совместимость с кодом, который уже использует старое определение данного типа. Эта проблема непосредственно связана с одним из важных свойств обобщённого программирования, известным как ретроактивное моделирование. Ретроактивное моделирование предполагает возможность определения того, каким образом тип удовлетворяет концепту, без изменения типа. С удалением моделей концептов говорить о поддержке этого свойства если и можно, то только в очень ограниченном объёме, а именно: всегда можно добавлять перегруженные версии свободных функций, работающих с объектами данного типа. Однако наличие свободных функций является лишь одним из нескольких типов требований, которые могут накладывать концепты.

Завершая сравнение концептов C++0x и концептов C++1y, отметим, что в последних по сравнению с первыми были исключены также:

- аксиомы — средство описания семантических ограничений,
- синтаксическая поддержка уточнения (refinement) или, попросту, «наследования» концептов.

Аксиомы это достаточно высокоуровневые правила, которые в общем случае невозможно проверить с помощью компилятора. Пример аксиомы приведён в листинге 3: здесь показано, что полугруппой является пара из типа T и бинарной операции над элементами этого типа, причём операция должна быть ассоциативной — это условие сформулировано в виде аксиомы.

Аксиомы предполагались в помощь компилятору для проведения оптимизирующих преобразований. Соответствующие исследования проводились несколькими исследователями, поддерживавшими эту идею, однако число их было слишком мало, чтобы обеспечить вхождение механизма с таким нечётким назначением в новый упрощённый проект концептов.

Листинг 3: Аксиома ассоциативности — часть определения концепта полугруппы

```
concept Semigroup<typename Op, typename T> :  
    CopyConstructible<T> {  
    T operator()(Op, T, T);  
    axiom Associativity(Op op, T x, T y, T z) {  
        op(x, op(y, z)) == op(op(x, y), z);  
    }  
}
```

Пример уточнения также приведён на листинге 3: тип T в концепте полугруппы должен иметь конструктор копий (CopyConstructible<T>). Как замечено Дугом Грегором, одним из авторов концептов C++0x, в предложении по их упрощению [11], в отсутствии моделей концептов уточнения полностью эквивалентно ограничениям, которые можно накладывать на тип внутри определения концепта.

4. Перегрузка на основе концептов и связанная с ней проблема

В связи с удалением из нового дизайна концептов явных моделей возникает следующая проблема.

В техническом отчёте 2005 года ведущие специалисты в области обобщённого программирования, Дуг Грегор и Джереми Сик, высказали идею о том, что для успешной реализации перегрузки на основе концептов необходимо наличие механизма явного сопоставления концепта и типа, который должен ему удовлетворять. Пример Грегора и Сика достаточно прост и интересен, чтобы его рассмотреть и понять, как решают возникающую в нём проблему авторы концептов C++1z. Отметим сразу, что хотя упомянутое решение можно найти в описании проекта изменений в стандартную библиотеку в связи с запланированным добавлением концептов C++1z, само описание изменений библиотеки весьма обширно, а авторы новых концептов не вынесли это важное, по нашему мнению, решение ни в одну из известных нам статей или докладов, посвящённых новому элементу языка.

Пример, по существу, ставит проблему концептов, которые различаются только семантически. В качестве примера таких концептов можно взять широко распространённые в стандартной библиотеке C++ категории итераторов, которые планируется конвертировать в концепты в будущем стандарте. Хорошо известно, что итераторы ввода и прямые итераторы отличаются лишь семантикой операции инкремента.

Листинг 4: Перегрузка на основе концептов: конструктор типа vector с разной эффективностью

```
// O(lg n) allocations
template<Input_iterator Iter>
vector(Iter first, Iter last)
{
    while (first != last) push_back(*first++);
}

// 1 allocation
template<Forward_iterator Iter>
vector(Iter first, Iter last)
{
    auto n = distance(first, last);
    reserve(n);
    while (n--) push_back(*first++);
}
```

В листинге 4 приведён пример перегрузки функций-членов, а именно, конструкторов типа vector, принимающих итераторы начала и конца диапазона, элементы которого надо загрузить в создаваемый вектор. Перегрузка осуществляется на основе категории итератора, который задаётся концептом: для итератора ввода необходимо многократно выделять ($O(\lg n)$ выделений, если расширение происходит с помощью кратного увеличения резерва памяти), в то время как прямой итератор позволяет заранее вычислить количество элементов в полученном диапазоне и обойтись одним выделением.

Проблема состоит в том, что, как отмечалось, выше любой тип, (синтаксически) входящий в категорию итератора ввода, будет принадлежать также и категории прямого итератора. Как показал анализ предложения по редактированию стандартной библиотеки в связи с планируемым добавлением концептов, семантическое различие между разными типами авторы планируют, как и в старом стандарте кодировать заранее определёнными типами-метками (tag), для которых внутри типов-итераторов задаётся стандартный псевдоним iterator_category. На листинге 5 показано, как примерно должен выглядеть при таком подходе концепт итератора ввода.

Из листинга 5 видно, что данный концепт уточняет (refines) концепт WeakInputIterator, который включает некоторые элементарные свойства итераторов, такие как наличие операций разыменования и продвижения. Ограничения на тип-параметр I состоят в том, что объекты этого

типа должны быть сравнимы на равенство (EqualityComparable) и должны содержать вложенный синоним типа под названием `iterator_category` (его значение возвращает выражение `IteratorCategory<I>`), который наследуется (в частности, как часто бывает, равен) от стандартного типа-метки `input_iterator_tag`.

Листинг 5: Концепт итератора ввода

```
template<typename I>
concept bool InputIterator =
    WeakInputIterator<I>() && EqualityComparable<I>() &&
    Derived<IteratorCategory<I>, input_iterator_tag>();
```

Заметим, что строгое равенство типов обеспечивается аналогичным `Derived` отношением `Same`. Оба они вычисляются с помощью средств заголовочного файла `type_traits` стандартной библиотеки, появившегося в C++11, и, таким образом, не требуют существенной модификации существующих компиляторов.

5. Заключительные замечания относительно синтаксиса и реализации концептов C++1z

В предыдущих разделах мы не останавливались на специальном обсуждении синтаксиса для концептов, отдавая основное внимание семантике. Синтаксис концептов продолжает меняться и любые конкретные определения здесь могут быстро устареть. Однако для понимания языка описания концептов (пункт 1 из списка в разделе 2) следует сделать общее замечание относительно идеи введения новых синтаксических конструкций в язык программирования. Идея эта в целом заключалась в том, чтобы максимально использовать новые возможности из последних стандартов языка: 2011 и 2014 года.

Ключевое слово `concept` может появляться в одном из двух контекстов:

- в определении шаблона `constexpr`-функции (появились в C++11) — такой пример даётся в листинге 6,
- в определении шаблона переменной (появились в C++14) — такой пример приводится в листинге 5.

Листинг 6: Концепт Аллокатора (менеджера памяти)

```
template<typename A>
concept bool Allocator()
{
    return requires () {
        typename A::pointer; // (1)
        requires Pointer<typename A::pointer>; // (2)
        // ...
    };
}
```

В обоих случаях это означает наложение некоторых ограничений на соответствующий шаблон функции или переменной. К примеру:

- тип возвращаемого значения шаблона функции, как и тип шаблона переменной должен быть логическим типом;
- у шаблона функции не должно быть параметров;
- в теле функции должен быть только оператор `return`, аргументом которого выступает логическое выражение, вычисляемое на этапе компиляции, возможно, включающее `requires`-подвыражение;
- и т. п.

Наиболее существенное добавление к синтаксису C++ происходит в связи с `requires`-выражением, оно может включать два вида предложений: обращение к вложенным типам (таким образом требуется их существование) или допустимые выражения. Пример второго вида требований приведён на листинге 7: `requires` позволяет вводить псевдопеременные параметра-типа `T` (в этом примере их имена `a` и `b`) и требовать, чтобы выражения сравнения на равенство и на неравенство были допустимыми для этих переменных и возвращали тип `bool`.

```

template<typename T>
concept bool Equality_comparable()
{
    return requires (T a, T b) {
        {a == b} -> bool;
        {a != b} -> bool;
    };
}

```

Приведённый синтаксис имеет ряд очевидных мелких недостатков. Например, указание типа `bool` в обоих видах концептов является избыточным. Кроме того: для использования концепта, определённого как функция, требуется указание круглых скобок (операции вызова функции), в то время как для переменной они не нужны и приведут к ошибке компиляции, если их добавить — это создаёт ненужное синтаксическое различие между разными концептами в то время как семантически они абсолютно эквиваленты.

Вполне возможно, что указанные недостатки будут сняты в будущих редакциях проекта нового стандарта [12].

В августе 2015 года прототипная реализация концептов на основе компилятора GCC была добавлена в основную ветвь разработки этого компилятора [13]. Это изменение планируется зафиксировать в версии 6 компилятора, которая запланирована к выходу весной 2016 года. Однако уже сейчас можно самостоятельно собрать данную версию компилятора из исходных кодов либо воспользоваться онлайн-ресурсами, предоставляющими веб-интерфейс к такой сборке [14]. Для включения механизма концептов при компиляции программ следует указывать ключ `-std=C++1z`.

Литература

1. *Sutton A., Stroustrup B., Dos Reis G.* Concepts Lite: тех. отч. — 28 июня 2013. — ISO/IEC JTC1/SC22/WG21 N3701.URL: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3701.pdf>
2. *Sutton A.* C++ Extensions for Concepts PPTS: тех. отч. — 9 февр. 2015. — ISO/IEC JTC1/SC22/WG21 N4377.URL: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4377.pdf>
3. A Concept Design for the STL: тех. отч. — 13 янв. 2012. — ISO/IEC JTC1/SC22/WG21 N3351. URL: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2012/n3351.pdf>
4. *Stroustrup B.* The C++0x “Remove Concepts” Decision // Dr. Dobb’s Journal. — 2009.URL: <http://www.drdoobs.com/cpp/the-c0x-remove-concepts-decision/218600111>
5. *Järvi J., Willcock J., Lumsdaine A.* Associated types and constraint propagation for mainstream object-oriented generics // OOPSLA '05: Proceedings of the 20th annual ACM SIGPLAN conference on Object-oriented programming systems languages and applications. — ACM Press. New York, NY, USA : ACM Press, 2005. — С. 1–19. — ISBN 1-59593-031-0. — DOI: 10.1145/1094811.1094813.
6. *Belyakova J., Mikhalkovich S.* Pitfalls of C# Generics and Their Solution Using Concepts // Proceedings of the Institute for System Programming. — Moscow, Russia, 2015. — Июнь. — Т. 27, No 3. — С. 29–45. — ISSN 2079-8156.
7. Lightweight, flexible object-oriented generics / Y. Zhang [и др.] // 36th ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI). — Июнь 2015. — С. 436–445. URL: <http://www.cs.cornell.edu/andru/papers/genus>
8. ConceptGCC. — URL: <http://www.generic-programming.org/software/ConceptGCC/>
9. Concepts: Linguistic Support for Generic Programming in C++ / D. Gregor [и др.] // SIGPLAN Not. — New York, NY, USA, 2006. — Окт. — Т. 41, No 10. — С. 291–310. — ISSN 0362-1340. — DOI: 10.1145/1167515.1167499.
10. *Tang X., Järvi J.* Axioms as generic rewrite rules in C++ with concepts // Science of Computer Programming. — 2015. — Т. 97, Part 3. — С. 320–330. — ISSN 0167-6423. — DOI: 10.1016/j.scico.2014.05.006. — Object-Oriented Programming and Systems (OOPS 2010) Modeling and Analysis of Compositional Software (papers from EUROMICRO SEEA'12).
11. *Gregor D.* Simplifying C++0x Concepts: тех. отч. — 9 апр. 2013. — ISO/IEC JTC1/SC22/WG21 N3629. URL: <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2013/n3629.pdf>
12. *Brown W.* Tweaks to Streamline Concepts Lite Syntax: тех. отч. — 10 апр. 2015. — ISO/IEC JTC1/SC22/WG21 N4434. URL: <http://open-std.org/JTC1/SC22/WG21/docs/papers/2015/n4434.pdf>
13. *Merrill J.* Huge C++ PATCH to merge c++-concepts branch — Mail archive for the GCC project. URL: <https://gcc.gnu.org/ml/gcc-patches/2015-08/msg00377.html>
14. Wandbox: Social Compilation Service. — URL: <http://melpon.org/wandbox>