

С. Свердлов

Оберон — воплощение мечты Никлауса Вирта

Источник: Компьютерра, 1996, №46.

Название этим заметкам я придумал около года назад, когда в шестом номере журнала "Монитор" за 1995 год прочел статью Сергея Орлика "О чем не мечтали Вирт и Гради Буч, или Новая объектная модель Delphi". И хотя об ушедших ("Монитор" вскоре после выхода того номера закрылся) "или хорошо, или ничего", очень хочется возразить журналу и автору, хотя бы и с опозданием. Оговорюсь сразу, что сама статья никаких возражений не вызывает. Она посвящена языковым новациям фирмы Borland, сделавшей очередные добавления к своей версии языка Паскаль. С. Орлик излагает факты, а против фактов что возразишь? И журнал "Монитор" нравился мне больше многих компьютерных изданий, поскольку не публиковал репортажей с тусовок компьютерных торговцев, а писал все больше о фундаментальных идеях и алгоритмах. На этом, видно, и прогорел.

Протест вызывает заглавие статьи. Вы, наверное, согласитесь, что в этом залихватском заглавии, сквозит намек на то, что Н. Вирт и Г. Буч имеют теперь повод позавидовать крутым парням из Borland, так здорово усовершенствовавшим Паскаль.

Не берусь говорить про Г. Буча, а Никлаус Вирт (Niklaus Wirth) уж точно мечтал совсем о другом. В то время, когда Borland превращала простой и изящный Паскаль в его собственную противоположность, Вирт предложил язык программирования Оберон, который является развитием Модулы-2, но проще и в то же время мощнее Модулы (бывает, оказывается, и так!). Более того, некоторые нововведения, которые появились в Object Pascal в 1995 г., Вирт включил в Оберон еще в 1987 г. А в Турбо Паскале в 1987 г. появились модули. Мне даже почему-то кажется, что разработчики из Borland в середине 1990-х были-таки знакомы с маленьким шедевром великого швейцарца. Вот и решите, кто кому должен завидовать.

Что такое хорошо

Но дело ведь даже не в названии той статьи. (У меня, кстати, есть гипотеза, что первоначально она называлась по-другому. Интересно было бы это проверить. Ау-у! С. Орлик). Огорчает, что явно и неявно по поводу и без повода читателю внушается мысль, что хорошо именно то, что предлагают нам в своих коммерческих разработках монополисты компьютерной отрасли. Но ведь они руководствуются своими, а не отнюдь не вашими, как вы можете считать, интересами. Не выпустив в свое время компилятор Турбо Модула-2, который был готов, а продолжив расширение Паскаля, Borland ориентировалась на прогноз своей, а вовсе не вашей будущей прибыли.

В то же время всегда существовали и существуют разработки, которые создаются не в обстановке сумасшедшей спешки в ходе гонки на выживание, а являются продуктом обстоятельной и вдумчивой работы.

Мне, конечно, возразят. Скажут: "Все решает рынок. То, что имеет успех на рынке — хорошо. Остальным — так им и надо". Полно. Разве вы не помните как, доведенные до иступления, американские домашние хозяйки в ночь на 24 августа 1995 г. стояли в очередях в компьютерные магазины за вожденной коробочкой с белыми облаками? Или, может, вы не мыли голову Head & Shoulders? Реклама может все!

Самое ужасное, что страдают в первую очередь юные. Ведь им же не с чем сравнивать. И когда молодой человек слышит: "Настоящие профессионалы выбирают Си++", что ему остается? Кто ж не хочет стать "настоящим профессионалом"? Сориентироваться же молодым довольно трудно. Прилавки книжных магазинов забиты опусами вида "Освоение системы X версии Y для операционной системы W полными идиотами за Z дней". Когда я стал приносить на занятия и показывать своим ученикам книги Д. Кнута, Э. Дейкстры, Д. ван Тассела, Г. Майерса, Б. Мейера и К. Бодуэна, студенты были поражены в первую очередь самим фактом существования книг подобного рода. Признаюсь, что и сам, когда на книжном прилавке 20 лет назад рядом стояли трехтомник (как я теперь понимаю, классический) Д. Кнута "Искусство программирования для ЭВМ" и том К. Джермейна

“Программирование на IBM/360”, я выбрал Джермейна (издание третье, стереотипное — значит, не я один). Вспомните, сорокалетние.

Что в имени твоём

Язык программирования Оберон создан Н. Виртом в 1987 году в ходе разработки одноименной операционной системы для однопользовательской рабочей станции Ceres. Язык и операционная система названы именем одного из спутников планеты Уран — Оберона, открытого английским астрономом Уильямом Гершелем ровно за двести лет до описываемых событий и носящего имя героя комедии Шекспира “Сон в летнюю ночь”, царя эльфов и фей.

Внимание Н. Вирта к спутнику далекой планеты было привлечено блистательным успехом экспедиции американского космического аппарата “Вояджер-2”, который, стартовав в 1977 году, в январе 1986-го передал на Землю высококачественные снимки пяти известных к тому времени и десяти вновь открытых спутников Урана. Новые спутники некоторое время спустя также были названы именами шекспировских героев. На Вирта произвели впечатление надежность и удивительная живучесть бортового компьютера “Вояджера”, который при своей относительной простоте обеспечил проведение уникального эксперимента.

Оберон, Оккам и Delphi

Может быть вы знаете язык программирования Оккам. Он носит имя английского философа XIV века Уильяма Оккама, который выдвинул принцип, известный как “бритва Оккама”: “сущности не следует умножать без необходимости”. Именно такого подхода к разработке и совершенствованию языков программирования Вирт придерживается на протяжении всей своей деятельности, а вот его оппоненты делают обычно наоборот. Так было, когда вместе с Тони Хоаром Вирт вступил в конфликт с Рабочей Группой IFIP, которая занималась усовершенствованием Алгола-60. Группа создала Алгол-68; Вирт — Паскаль. Кто оказался прав? Так было в конце 1970-х, когда практически одновременно появились Ада и Модула-2 — языки сравнимые функционально, но насколько Модула проще. И, наконец, C++ и Object Pascal с одной стороны и Оберон — с другой. Синтаксис Оберона целиком умещается на одной странице, а исчерпывающе полное описание всего языка составляет около 20 страниц.

Создатели же Object Pascal сплошь и рядом пренебрегают принципом бритвы Оккама. Возьмем самое дорогое, к чему они так трепетно относятся — объекты. Посмотрите сколько слов потребовалось для обслуживания этого понятия: `as`, `class`, `constructor`, `destructor`, `inherited`, `is`, `object`, `property`, `abstract`, `default`, `dynamic`, `message`, `nodefault`, `override`, `private`, `protected`, `public`, `published`, `stored`, `virtual` (я, кажется, еще что-то пропустил). Это при том, что разница между `object` и `class` состоит лишь во внутренних механизмах реализации. А сможете ли вы быстро ответить чем отличаются `public` и `published`, `private` и `protected`?

Как же обстоят дела с объектами в Обероне? А никак! Там этого слова вообще нет! Но есть и наследование, и полиморфизм, и динамический контроль, и приведение типов, и скрытые поля. Сколько для всего этого потребовалось новых зарезервированных слов? Одно — `IS`.

О слове объект хочется сказать особо. Слово это в последние годы превратилось в фетиш, ему придается какое-то магическое значение. Но ведь объекты это не самоцель. Согласитесь, что не существует потребности во что бы то ни стало употребить в программе объекты. А вот, скажем, поддерживать список из разнотипных элементов с индивидуальным поведением — реальная необходимость во многих задачах. Оберон содержит все, что требуется для создания таких списков.

Разные подходы к отбору языковых средств видны и на следующем примере. С самого начала (опубликовано в 1988 году) в языке Оберон предусмотрена возможность проверки во время выполнения фактического типа объекта и безопасный способ манипулирования объектом в соответствии с его фактическим типом (охрана типа). Без этих средств объектная модель языка не может считаться полноценной. В то же время Турбо Паскаль в течение многих лет с 1989 г. (версия 5.5) как-то без этого обходился. И только в Delphi появились необходимые механизмы, названные RTTI (неумеренное использование аббревиатур свидетельствует, на мой взгляд, о неадекватности вводимых в обиход понятий, или/и, по крайней мере, о неспособности их авторов придумать для своих творений название хотя бы из двух слов).

Алгол, Паскаль, Модула, Оберон

Оберон является продолжением линии языков Алгол-60, Паскаль, Модула-2. Как подчеркивает Вирт, Обертон получен исключением из Модулы-2 большого числа средств, а добавлением лишь немногих.

В Обероне отсутствуют записи с вариантами, перечислимые и ограниченные типы, множества общего вида, тип `CARDINAL`, нецелочисленные индексы массивов (нижняя граница индекса всегда 0), локальные модули и модули определений, списки экспорта, оператор `for`, прежний вариант оператора `with` и даже главная программа. Предусмотрены указатели только на записи и массивы. Упрощены списки импорта, в которых теперь перечисляются только названия модулей (в то же время в тексте программы импортированные имена сопровождаются явным указанием модуля-экспортера).

Очень изящно оформляется экспорт модулей. Те, кто пишет программы на Турбо (Borland, Object) Паскале, хорошо знают как громоздко выглядят интерфейсные части модулей, особенно если там располагаются описания объектов. А поскольку все "внутренности" объектов вынужденно выставляются напоказ, возникает потребность в словах `private` и `protected`, `public` и `published`, что еще больше загромождает как текст программы, так и наше восприятие программы и языка. В Обероне экспортируемые имена просто помечаются звездочкой (*), а текст интерфейса формируется автоматически специальным инструментом Оберон-среды — смотрителем (это пушкинское слово мне кажется наиболее подходящим переводом для `browser`).

Основное добавление по сравнению с Модулой-2 — расширение типов (записей). Расширение типов (наследование) делает Оберон объектно-ориентированным языком. При этом, как уже говорилось, предусмотрены необходимые механизмы поддержки полиморфизма — проверка и охрана типа (их аналогом в Delphi является RTTI).

Очень естественно и просто решен в Обероне вопрос о совместимости числовых типов. Они образуют иерархию:

```
SHORTINT <= INTEGER <= LONGINT <= REAL <= LONGREAL.
```

Значения "меньших" типов могут присваиваться "большим".

Открытые массивы в Обероне в отличие от Модулы могут быть многомерными. (В Delphi одномерные открытые массивы проходят по разряду нововведений.)

Оберон и Оберон-2

"Сделай так просто, как возможно, но не проще того", — это высказывание А.Эйнштейна Вирт выбрал эпиграфом к описанию языка Оберон. Удивительно простой и даже аскетичный Оберон является, пожалуй, минимальным универсальным языком высокого уровня.



Рис. 1. Маэстро Никлаус Вирт.

По прошествии некоторого времени, вероятно, не без сопротивления, Вирт согласился с добавлениями к языку, предложенными его коллегами. Язык стал называться Оберон-2. Авторами описания Оберона-2 являются Никлаус Вирт и Ханспетер Мёссенбёк (Hanspeter Mossenböck) — профессор университета Иоганна Кеплера в австрийском городе Линце. Эпиграф Эйнштейна был снят.

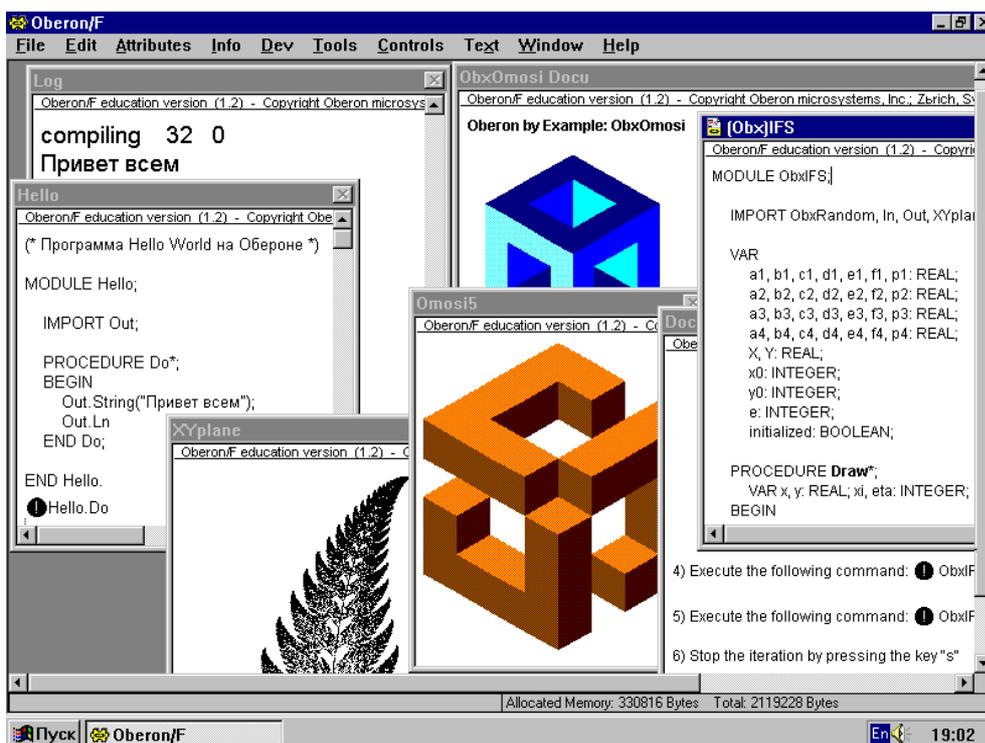


Рис. 2. Интегрированная среда Oberon/F.

В Oberon-2 возвращен оператор FOR, которому незадолго до того Вирт приписывал “маргинальное значение”. (Если бы не зарезервированные слова FOR, TO и BY, вернувшиеся в язык, Oberon-2 можно было бы называть правильным расширением Oberona: любая программа на Oberone была бы и программой на Oberon-2).

Другими нововведениями являются экспорт только для чтения (после имени объекта при таком его экспорте ставится не “*”, а “-”), открытые массивы в качестве базового типа указателей, расширенный оператор with.

Главное же новшество — связанные процедуры (виртуальные методы в терминах Турбо Паскаля). Объясняя их отсутствие в Oberone, Вирт находил достаточно убедительные аргументы, отмечая, что определение действий, применимых к объектам некоторого типа непременно в составе определения самого этого типа представляет собой догму, происходящую из концепции абстрактного типа данных и создающую некоторые препятствия при разработке больших систем. Решение, принятое в Oberon-2 несет черты компромисса — описания процедур-методов помещаются не в объявлении самого типа, а отдельно. Эти процедуры лишь “связаны” с типом.

Oberon-2 является сейчас фактическим стандартом языка и поддерживается большинством Oberon-систем, которых существует довольно много. Это во-первых, семейство систем ETH — Федерального Технического Университета (Цюрих, Швейцария) — организации, где работает Н. Вирт. Это Oberon/F (Oberon microsystems Inc., Цюрих) — коммерческая Oberon-система, снабженная более традиционным, чем системы ETH, пользовательским интерфейсом (в мае 1996-го выпущена версия 1.2). Это реализация университета Кеплера в Линце. Наконец, можно назвать компиляторы Модуль-2 и Oberona-2 (система XDS), созданные в Новосибирске учениками Андрея Петровича Ершова.

И все же “улучшать” создание классика дело неблагодарное. Все, что предлагается Виртом продумано им очень детально и глубоко. Поэтому некоторые новации Oberona-2, по-моему, неоспорны. Заметно усложняется компилятор в части, относящейся к трансляции связанных процедур. Интерфейс модуля теперь перестает быть просто перечнем цитат, взятых из текста реализации модуля, работа зрителя затрудняется. Экспорт только для чтения мог порождать бессмысленные варианты, когда “только для чтения” экспортируется константа, тип или процедура. Лишь совсем недавно эта неточность была замечена и исправлена.

Оберон и Солнце, Кофе и Сок

В последнее время я с интересом наблюдаю как одно за другим в печати появляются признания поклонников Си++, из которых вдруг выясняется, что любимый язык, оказывается, доставляет им массу проблем. Сложен, механизмы контроля несовершенны, провоцирует на всякие опасные трюки (адресная арифметика и т.п.). Отчего такая откровенность? А дело в том, что компания Sun (Солнце), совершила трезвый поступок, предложив язык Java, который является модернизацией Си++, отличаясь от своего предшественника простотой и отказом от потенциально опасных возможностей. Все рукоплещут и каются. Но, не кажется ли несколько неприличным такое запоздалое коллективное прозрение?



Рис. 3. Оберон — самый удаленный от планеты и второй по величине спутник Урана..

Направление модернизации Си++, приведшее к появлению языка Java, находится в русле тех принципов, которые исповедует Н. Вирт на протяжении многих лет и блистательно воплотил в Обероне: простота, неизбыточность, безопасность. Я уж не говорю о таком решении как сборщик мусора, который предусмотрен в Java и который Н. Вирт использовал еще в языке Алгол W, а после того как избавился от записей с вариантами — и в Обероне. Так уж не отраженным ли светом маленького спутника по имени Оберон сияет огромная звезда Sun?

Язык Java (Кофе — на американском жаргоне) является частью технологии распространения приложений в сети Интернет. Но и в этом деле не обошлось без Оберона. В июле 1996 года в телеконференции `comp.lang.oberon` опубликовано сообщение о новой технологии Juice (Сок), предназначенной для распространения исполнимого кода через World Wide Web и основанной на использовании Оберона. Авторы технологии Михаэль Франц и Томас Кистлер отмечают три ключевых компонента Juice:

- архитектурно-независимый формат распространения программ,
- компилятор, транслирующий с Оберона в переносимый формат Juice и
- дополнение (plug-in), которое позволяет Netscape browser'у (смотрителю) исполнять Juice-приложения.

Для пользователя Интернет преимущества технологии Juice проявляются в том, что, во-первых, Juice-версия переданной по Сети программы будет скорее всего выполняться много быстрее аналогичного Java-апплета, поскольку код Java как правило интерпретируется, а Juice всегда компилирует приложение в родной код целевой машины перед его выполнением. Во-вторых, переносимый формат Juice компактнее байт-кода Java, и пользователи с медленным подключением к Сети сэкономят время, загружая приложения Juice. Наконец, использование Juice позволяет избежать многих проблем безопасности, поскольку, в отличие от Java, не требует выполнения потокового анализа, чтобы убедиться, что загруженный код не выполнит запрещенных действий.

Мне нравятся Delphi и Oberon

Вам может показаться, что я люто ненавижу Object Pascal, Borland и Delphi. Ничего подобного. Вот уже около 8 лет я использую Турбо — Borland — Object Паскаль. Причем еще и теперь наряду с Delphi имею счастье благодаря некоторой технической отсталости нашего института и невероятной надежности японских компьютеров “Ямаха” (ни одной сколько-нибудь серьезной поломки у 20 компьютеров за 8 лет!) видеть в работе Турбо Паскаль 2.0. Поверьте, это классная программа. Ну а что касается Borland, то я не знаю другой компании, чьи программы были бы так же надежны, удобны, изящны, быстры и непрехотливы.

Я просто хочу, чтобы мы отдавали себе отчет в том, что направление, которое приняла эволюция языка Паскаль в системах Borland, далеко от идеала. Напротив, качество и проработка среды программирования систем Borland выше всяких похвал. В этом отношении Oberon-система очень трудно (если вообще возможно) конкурировать с Borland. Системы, разработанные в ETH, имеют весьма своеобразный, хотя и очень простой концептуально пользовательский интерфейс. А среда разработки Oberon/F, хотя и оснащена интерфейсом традиционным, но, по крайней мере, внешне и на первый взгляд выглядит заметно беднее Delphi.

Вообще и разработчики, и некоторые аналитики придают очень большое значение самой идеологии построения Oberon-систем, употребляя всякие магические слова вроде “объектная прикладная среда”, “компонентное программное обеспечение”, “документоориентированные системы” и т. п. Мне же кажется, что основная ценность — в самом языке Oberon. Пользовательская парадигма видоизменяется довольно быстро, языки же более консервативны. Когда система навязывает человеку слишком многое в способе его существования в компьютерной среде, это может не устраивать творчески настроенного индивидуума. Утверждать же, что рядовые пользователи непрограммисты когда-либо будут обитать в Oberon-среде, было бы опрометчиво.

Будущее Oberon предвидеть непросто. Мне, например, представляется, что Oberon — идеальный язык для изучения программирования. Сочетание простоты, строгости и неизбыточности предоставляет начинающему программисту великолепную возможность, не заблудившись в дебрях, выработать хороший стиль, освоив при этом и структурное и объектно-ориентированное и модульно-компонентное программирование. Но слишком от многих обстоятельств, часто не имеющих прямого отношения к достоинствам самого языка, зависит его судьба.

Подробности письмом

Тем, кто заинтересуется Oberоном могут узнать о нем подробнее. Вот перечень известных мне публикаций на русском языке.

1. Джонсон Т., Браун Р., Содерблом Л. Спутники Урана. В мире науки. №6, 1987
2. Кауфман В.Ш. Языки программирования. Концепции и принципы — М.: Радио и связь, 1993
3. Супрун А.Д. Oberon — новый продукт создателей Pascal и Modula-2. Компьютеры+Программы №5, 1994
4. Орлов С. Компонентное ПО и объектная прикладная среда Oberon. Computer Week Moscow №37, 1995
5. Богатырев Р. Java и Juice: дуэль технологий ?! Компьютерра #34, 1996

А это работы Н. Вирта и его соратников:

1. N.Wirth. The Programming Language Oberon. Software – Practice and Experience, 18, 7 (July 1988), 671-690.
2. N.Wirth and J.Gutknecht. The Oberon System. Software – Practice and Experience, 19, 9 (Sept 1989).
3. M.Reiser. The Oberon System. Addison-Wesley, 1991
4. Reiser and N.Wirth. Programming in Oberon. Steps beyond Pascal and Modula. Addison-Wesley, 1992.

5. Wirth and J.Gutknecht. Project Oberon. The Design of an Operating System and Compiler. Addison-Wesley, 1992.
6. H.Mossenbock. Object-oriented Programming (in Oberon-2). Springer, 1993
7. N.Wirth. Theory and Techniques of Compiler Construction. Addison-Wesley, 1996.

Хочу обратить внимание отечественных издателей на последнюю книгу. Давно уже ничего подобного в русском переводе не выходило, а как было бы здорово! Сколько можно про "чайников"-то. Вирта у нас хорошо знают, читают и любят (в этом году он даже стал почетным профессором Новосибирского университета), и десяти тысячный тираж книга наверняка выдержит.

Но, конечно, больше всего сведений можно найти в Интернет.

Вначале информация для тех, кто располагает, как и я, доступом только к электронной почте. Чтобы получить описание языка Оберон-2 из первоисточника вам достаточно отправить по адресу ftpmail@relcom.ru письмо такого содержания:

```
open ftp.inf.ethz.ch
cd /pub/Oberon/Docu
get Oberon2.Report.Text
quit
```

(В названии файла и каталогов заглавные и малые буквы различаются.) В ответ вы получите несколько писем, содержащих порезанный на кусочки файл Oberon2.Report.Text, размер которого 69004 байта. Он имеет формат текстового документа ETH-Oberon V4, но превратить его в обычный ASCII не составит большого труда: надо просто отрезать заголовок. Русский перевод этого основополагающего документа можно получить на Steel Rat BBS (Вологда) +7 (817-2) 21-38-39 (файл Oberon2r.zip).

На том же ftp-сервере находятся дистрибутивы нескольких Оберон-систем. В каталоге /pub/Oberon/Windows содержится ETH-Oberon V4 для Windows. Файл OBERON.ZIP имеет размер 955185 байт. Это 32-разрядная система, которая будет работать с Windows 95 и Windows NT, а для запуска под управлением Windows 3.1 и 3.11 требует установки 32-разрядного расширения Win32s. Oberon/F для Windows версии 1.2 (май 1996 года) находится в каталоге /pub/Oberon/OberonF/WinOberonF12. Минимальный комплект состоит из файлов disk1.exe (1268843 байт) и disk2.exe (1292641 байт) и позволяет установить систему в Windows 95 и Windows NT.

Из каталога /pub/Oberon/Sources вы можете получить файл ProjectOberon.V4.tar.gz (178933 байта), содержащий исходный текст (на Обероне, естественно) Оберон-системы (включая компилятор, написанный лично Н. Виртом) для компьютера Ceres. Этот файл представляет собой созданный утилитами UNIX сжатый архив, для распаковки которого вам потребуются соответствующие программы.

Ну, а тем, кто имеет доступ к WWW, сообщаю такие адреса:

- Институт компьютерных систем Федерального Технического Университета (ETH), Цюрих, Швейцария:
<http://www.inf.ethz.ch/department/CS/Oberon.html>
- Отечественная реализация языка Оберон (система XDS, Институт информационных систем, xTech Ltd., Новосибирск):
<http://www.iis.nsk.su/xtech/xds/>
- Информация о технологии Juice (Университет штата Калифорния, США):
<http://www.ics.uci.edu/~juice>

В заключение хочу поблагодарить профессора Н. Вирта (ETH) и аспиранта ВМК МГУ М. Зырянова за содействие в получении сведений, использованных в этой статье.

Об авторе: Сергей Залманович Свердлов — доцент Вологодского педагогического университета. С ним можно связаться по адресу: IVT@VGPI.VOLOGDA.SU