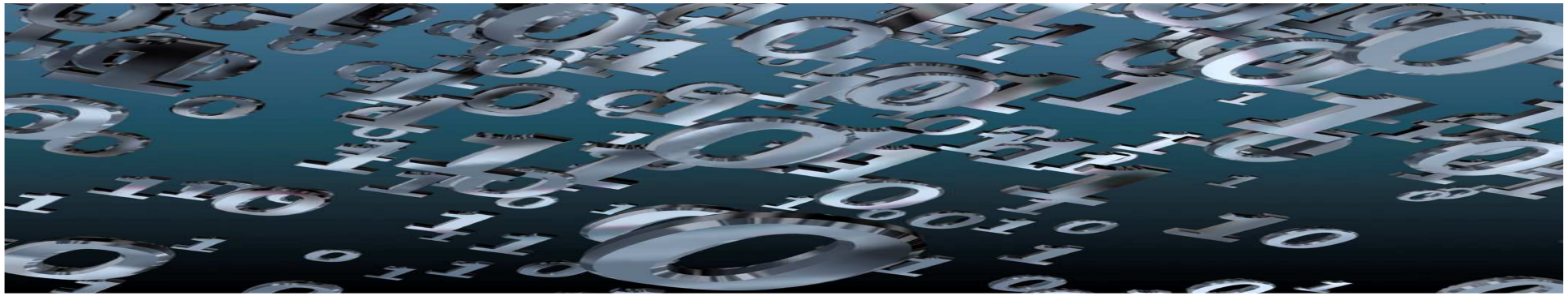
ВЫБОР ЯЗЫКА ПРОГРАММИРОВАНИЯ ДЛЯ ЗАЩИЩЕННЫХ СРЕДСТВ СВЯЗИ.

С. И. Киркоров,

Научно-производственное предприятие МедиаСкан,

Республика Беларусь



Понятие аппаратно-программной платформы расширяется

- Желание абстрагироваться от технических деталей и сосредоточиться специфике задачи привело к бурному росту, в том числе и узко прикладных языков программирования. Характерной чертой, которых является их жесткая привязка к аппаратно-программной платформе (АПП). Таким образом, целесообразно определить аппаратно-программную платформу как связку: аппаратура-ОС-среда выполнения.

«Модные» сегодня аппаратно-программные платформы

- В технической и экономической сфере наиболее популярные платформы: виртуальная машина (VM) Java, GCC, .NET, WEB-вычисления (сервер Apache, MS IIS).
- Достижения физико-математических наук аккумулируются в таком инструментарии, как Mathematica (фирма WOLFRAMRESEARCH), Mathcad (фирма PTC), Statistica (фирма STATSOFT), Matlab (фирмы MATHWORKS).

Преимущества и недостатки специализированного инструментария

- Преимущества:
  - Удобство и простота использования
  - Готовые решения
  - Наработанные методики для преподавания
  - Быстрые в разработке решения для прикладной сферы не требующей защиты информации и высокой надёжности кода
- Недостатки:
  - Высокая стоимость
  - Готовые решения – не новые решения
  - Узкоспециализированные языки программирования
  - Генераторы исполняемого кода дают медленный и плохо читаемый код
  - Сложность или невозможность проведения сертификационных испытаний на безопасность

Инструментарий удобен в сфере образования или на самом раннем этапе подготовки к проведению вычислительного эксперимента при создания реальных систем.

Высокая его стоимость сужает область использования.

Трудности в создании реальных систем или в проведении вычислительного эксперимента

- Завышенные требования к производительности ЭВМ
- Ограничения в объеме обрабатываемых данных
- Низкая скорость обработки данных
- Отсутствие механизмов обеспечивающих защиту обрабатываемой информации
- Трудности при переходе с одной платформы на другую, более производительную и безопасную

Выбор аппаратно-программной платформы для компьютерных реализаций математических моделей

- Одной из причин ограничений применения такого инструментария это сложность или высокая стоимость обеспечения соответствующего уровня гарантий конечного продукта в области технической защиты информации по всем трем составляющим – конфиденциальность, целостность и доступность. Это связано с различиями оптимизированного под прикладную область алгоритма и его математической моделью, реализованной такими пакетами.

Трудности в создании реальных систем

- Сложность и стоимость возрастет, когда стоит задача переноса модели алгоритма и ее реализации на другую доверенную платформу. Под доверенной платформой здесь подразумевается собственный вычислитель, например специализированный процессор с другой операционной системой или без нее или программируемые пользователями вентильные матрицы. Для тестирования доверенной платформы и параллельных вычислений реализуемые ее, как правило, создается стендовое оборудование. Стендовое оборудование как средство измерения также требует соответствующего уровня гарантий правильности своего функционирования.

Правильный выбор - использовать инструментальное средство, которое:

- Само могло бы пройти сертификационные испытания в области технической защиты информации, то есть, как минимум имела открытые спецификации, коды программ и так далее.
- Соответствовало стандарту, строго его выполняло и имело стандарт, регламентирующий эти проверки. Было реализовано (или могло быть адаптировано) для целевых платформ.
- Обеспечивало поддержку многозадачности (для моделирования алгоритмов, реализующих параллельные вычисления) на уровне языка высокого уровня. Этим достигается стабильность семантики, разработчик избавляется от необходимости использования разнородных внешних библиотек или собственных решения обеспечивающих многозадачность.
- Влияет, конечно, и экономический фактор. Стоимость приобретения и совокупного владения такого инструментария не должна превышать 1/3 от всей стоимости разработки.

Стандарты безопасности ПО

- TCSEC (Оранжевая книга)
  - Критерии оценки безопасности высоконадежной компьютерной системы
- Общие критерии оценки безопасности в Информационных технологиях (ИТ) (ISO/IEC 15408-1, в РБ СТБ 34.101.1)
  - Критерии оценки безопасности ИТ
  - 7 уровней оценки безопасности

Уровни оценки безопасности (EALs)

EAL	Ограничения на разрабатываемое ПО
EAL7	Формально доказанная корректность + тестирование
EAL6	Использование доказательства корректности при проектировании + тестирование
EAL5	Проектирование с использованием формальных методов + тестирование
EAL4	Методологическая проектирование, тестирование и исправление
EAL3	Проведены методологические тесты и проверки
EAL2	Проведен структурный тест
EAL1	Оттестирована функциональность

Значима ли надёжность программного обеспечения?

- Несомненно значима! На маркетинговом уровне ☺
  - Ни один поставщик не скажет, что его программное обеспечение ненадёжно
  - Ни одна команда разработчиков не сообщит, что разрабатывает ненадёжное ПО
- В действительности, есть огромное количество ПО, ошибки в котором нас никак не задевают
- Не все программы нуждаются в том, что бы требование надёжности ставилось на первый план
- Сбой полезных, но некритичных программ все еще приемлемо 😐
  - Если произойдет сбой во время этой презентации – достаточно просто перегрузить компьютер
  - Если Ваш текстовый редактор зависнет во время набора важного документа, это не принесет Вам ощутимого вреда, если Вы часто сохраняли результаты своей работы

ПО и Критичность

- Критичность по отношению к бизнес-процессам
  - Сбой программного обеспечения может привести к значительным финансовым потерям и даже к полной остановке бизнеса
  - Например, система межбанковских платежей

- Критичность по отношению к решаемой задаче
  - Сбой программного обеспечения может привести к невыполнимости поставленной задачи
  - Например, спутник для исследования Марса

- Критичность по отношению к безопасности
  - Сбой программного обеспечения может привести к человеческим жертвам или большим разрушениям
  - Например, самолет

Сбои ПО в средствах связи и последствия

- Январь 15, 1990: на 9 часов остановлена общенациональная телефонная сеть США
  - месяц ранее AT&T обновила ПО на 114 коммутируемых телефонных станциях
  - Причина: 1 неуместный оператор “break” в программе на языке C
- Январь 2001: отзывается 230,000 единиц новых мобильных телефонов с доступом в Интернет
  - Пользователи сообщают, что их телефоны зависают после посещения некоторых web-узлов, а после перезапуска телефона все сохраненная на нем информация (адреса, ссылки, записи) теряется
- Matracom 6500 PABX (телефонный коммутатор)
  - Искажение случайных телефонных разговоров
  - Внезапное прерывание длинных телефонных звонков

Международные стандарты на критичное к безопасности ПО

- RTCA/EUROCAE DO-178B
  - Международный стандарт на критичное для безопасности ПО в области авиастроения
- IEC 880
  - Стандарт на ПО для атомных электростанций
- IEC61508 / DEF STAN 00-55/56
  - Европейский стандарт безопасности
- Руководство разработчика ПО для транспортных средств
  - Стандарт безопасности, предложенный Ассоциацией разработчиков безотказного ПО для автомобильной промышленности MISRA (Motor Industry Software Reliability Association)

Уровни критичности ПО согласно DO-178B

Уровень критичности	Последствия от ошибки/сбоя ПО
Уровень A	Катастрофические *(Продукты уровня A сообщают экипажу самолета о его положении в пространстве и предотвращают его от падения, н.п. системы управления полетом, авиационные картографические базы, некоторые дисплеи)*
Уровень B	Опасные/Значительные *(Системы уровня B: слежение за движением и уклонение от столкновений)*
Уровень C	Большие *(Системы уровня C: связь и управление каналами связи)*
Уровень D	Незначительные *(Системы уровня D: системы обеспечения комфорта)*
Уровень E	Без последствий *(Системы уровня E: развлекательные системы)*

IEC61508 Уровни безопасности-сложности-целост-ности SCIL (Safety-Complexity-Integrity Levels)

Уровень SCIL	Последствия от ошибки/сбоя ПО
SCIL 4	Смерть одного или нескольких людей, существенные финансовые потери *(Область: аэрокосмическая, медицинские системы, системы управления движением, системы управления опасными процессами, системы торможения)*
SCIL 3	Серьезные телесные повреждения или финансовые потери *(Область: управление силовыми установками средств передвижения)*
SCIL 2	Неудобство или недовольство *(Область: кассовые терминалы в супермаркетах, аппараты выдачи сигарет/напитков)*
SCIL 1	Без последствий *(Область: студенческие проекты, исследования)*

Уровни целостности предложенные MISRA

Уровень целост-ности	Возможность контроля со стороны водителя	Оценка допустимости отказа	*Примеры возможных последствий при сбое ПО автомобиля*
4	Не поддается контролю	Абсолютно недопустимо	*Обесточивание усилителя рулевого управления*
3	Сложно контролируется	Чрезвычайно редко	*Отказ тормозной системы*
2	Утомляет	Редко	*Неработоспособность механизма очистки лобового стекла*
1	Отвлекает	Не желательно	*Неработоспособность стеклоподъемника*
0	Только вызывает неудобство	Допускается	*Неработоспособность радио/CD плеера*

Содержимое Лицензии Windows 2000

ЗАМЕЧАНИЕ ПО ПОДДЕРЖКЕ JAVA

ДАНОЕ ПРОГРАММНОЕ ИЗДЕЛИЕ МОЖЕТ СОДЕРЖАТЬ ПОДДЕРЖКУ ПРОГРАММ, НАПИСАННЫХ НА JAVA .

ТЕХНОЛОГИЯ JAVA - НЕ УСТОЙЧИВАЯ К СБОЯМ И НЕ РАЗРАБОТАНА, ИЗГОТОВЛЕНА, ИЛИ ПРЕДНАЗНАЧЕНА ДЛЯ ИСПОЛЬЗОВАНИЯ ИЛИ ПЕРЕПРОДАЖИ КАК ИНТЕРАКТИВНОЕ ОБОРУДОВАНИЕ УПРАВЛЕНИЯ В ОПАСНЫХ СРЕДАХ, ТРЕБУЮЩИХ ОТКАЗОУСТОЙЧИВОЙ РАБОТЫ, ТАКИХ КАК СИСТЕМЫ УПРАВЛЕНИЯ ЯДЕРНЫМ ОБОРУДОВАНИЕМ, СИСТЕМЫ НАВИГАЦИИ САМОЛЕТА ИЛИ СИСТЕМЫ СВЯЗИ, СИСТЕМЫ УПРАВЛЕНИЯ ВОЗДУШНЫМ ДВИЖЕНИЕМ, МАШИНЫ ПОДДЕРЖАНИЯ ЖИЗНЕОБЕСПЕЧЕНИЯ ИЛИ ОРУЖЕЙНЫЕ СИСТЕМЫ, В КОТОРЫХ СБОЙ В ТЕХНОЛОГИИ JAVA МОЖЕТ ПРИВЕСТИ НЕПОСРЕДСТВЕННО К СМЕРТИ, ТЕЛЕСНОМУ ПОВРЕЖДЕНИЮ, ИЛИ СЕРЬЕЗНОМУ ФИЗИЧЕСКОМУ ИЛИ ЭКОЛОГИЧЕСКОМУ УЩЕРБУ.

Sun Microsystems, Inc письменно обязал Microsoft делать эту оговорку.

Ada: использовать для систем, связанных с безопасностью

- Требования безопасности рекомендуют использование языка Ada для самых высоких уровней целостности

- Даже документ MISRA-C рекомендует использование Ады:

  Рекомендации по использованию языка С для создания ПО для транспортных средств:

  –"...очевидно, что есть и другие языки, которые в многом лучше подходят для создания систем, связанных с безопасностью, обладающие (к примеру) большей надежностью и лучшим контролем соответствия типов . Примером подобных языков, в целом значительно превосходящих С, есть Ada и Модула 2. Если эти языки доступны для предлагаемых систем, то их применение, в сравнении с С, считается более предпочтительным." стр.3.

Язык программирования и надежность - философия (1).

- Создание программы (программной услуги) – не написание кода, а определение и использование моделей и абстракций, соответствующих объектам и процессам в решаемой задаче или проблемной области.

- Уровень и разнообразие базовых абстракций должен соответствовать уровню и разнообразию решаемых задач («принцип сундука»)
  - готовые и удобные решения для часто встречающихся технологических потребностей;
  - удобные средства создания проблемно-ориентированных абстракций;
  - высокая производительность на полном жизненном цикле программной услуги;

Язык программирования и надежность - философия (2).

- «Все, что не разрешено – запрещено!» Язык реализует жесткую дисциплину прогнозирования (определения свойств абстракции при ее создании) и контроля * (использования абстракции в соответствии с определенными для нее свойствами);

- Чем раньше обнаружится ошибка (несоответствие использования абстракции определенным для нее свойствам), тем проще, быстрее и дешевле оказывается ее устранение;

---

* Кауфман В.Ш - Языки программирования. Концепции и принципы М.:Радио и связь, 1993

Язык программирования и надежность - философия (3).

- Программист в основном занят сопровождением и модификацией чужого кода, а не созданием своего
  - ясность и хорошая структурированность кода многократно важнее возможности быстро его написать!
- Сложный инструмент редко бывает надежным
  - предоставляя больше возможностей, чем Си++, Ada оказывается существенно более простым языком для изучения и понимания;
- Все, что может быть определено – должно быть явно определено!
  - умолчания и надежда на «здравый смысл» - один из основных источников недоразумений и ошибок!
  - определение Ada «замкнуто». По сравнению с Си/Си++, например:
    - правила видимости не требуют привлечения мифического понятия «пространства имен», а вполне обходятся синтаксическими конструкциями языка;
    - правила модульности и раздельной компиляции не требуют привлечения внеязыкового понятия «файл», а также обходятся синтаксическими конструкциями языка;

Язык программирования и надежность - философия (4).

- Опережающая стандартизация
  - Ada возникла и развивалась как ***стандарт*** языка программирования;
  - Средства контроля стандарта Ada были разработаны к моменту принятия первого стандарта языка – все индустриальные реализации Ada достаточно точно соответствуют той или иной версии стандарта, диалектов Ada в индустрии не было и нет;
- Классификация ошибок в программе на уровне стандарта языка. Стандарт четко подразделяет все содержащиеся в нем требования на следующие группы:
  - проверяемые во компиляции отельного модуля (нарушение требования означает, что компиляция не является успешной, в ее результате не будет создан объектный код);
  - проверяемые при сборке программы из успешно скомпилированных модулей (нарушение требования означает, что исполняемый файл создан не будет);
  - проверяемые при выполнении программы (нарушение требования приводит к возбуждению предопределенного исключения);
  - правила, нарушение которых реализация проверять не обязана.

  Средства контроля стандарта проверяют, что поведение реализации соответствует этой классификации!

Сравнение языков программирования отвечающий требованиям реализаций ПО средств связи

Требования	C#	Java	C/C++	Ada
Применение в различных аппаратно-программных платформах в текущий момент времени	+	+	+	+
Принят ISO стандарт языка программирования	-	?	+	+
Развитие языка должно происходить в рамках стандартизации и обеспечиваться инструментальными средствами разработки для всех версий стандартов языка	-	-	-	+
Интеграция с современными технологиями ИТ	-	-	+	+
Язык программирования должен быть перспективным в будущем	?	?	+/?	+
Имеется версия транслятора языка сертифицированная по требованиям стандартов технической защиты информации	-	-	+/?	+

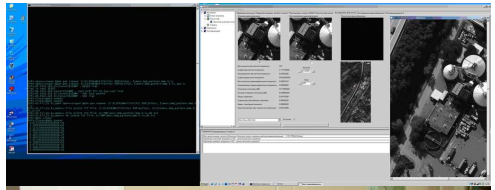
Обоснование выбора языка

- Встроенная поддержка многозадачности является уникальной и широко известной особенностью языка программирования Ada, которая выгодно отличает его от большинства современных языков программирования
- Одним из решением задачи – интеграция подходов и АПП на самом раннем этапе подготовки к проведению вычислительного эксперимента и при создания реальных систем, для которых необходимо проводить сертификационные испытания и аттестацию по требованиям информационной безопасности. Своеобразным "программным клеем" целесообразно использовать язык программирования Ada. Реализацией идеи интеграции является расширяемая платформа OEM-2011 for Windows разработанная на языке программирования Ada в УП МедиаСкан.

Примеры разработок на языке Ada

1. Подсистема управления въездом-выездом подземного паркинга ТЦ «СТОЛИЦА» г. Минск - 2005 г.

2. Программное обеспечение испытательного стенда аппаратуры сжатия видеоинформации г. Минск – 2008 г.



3. За рубежом начиная с 1983 г. : системы связи, управление транспортом, бортовые системы в авиации и космических аппаратов, и в других критичных приложениях.

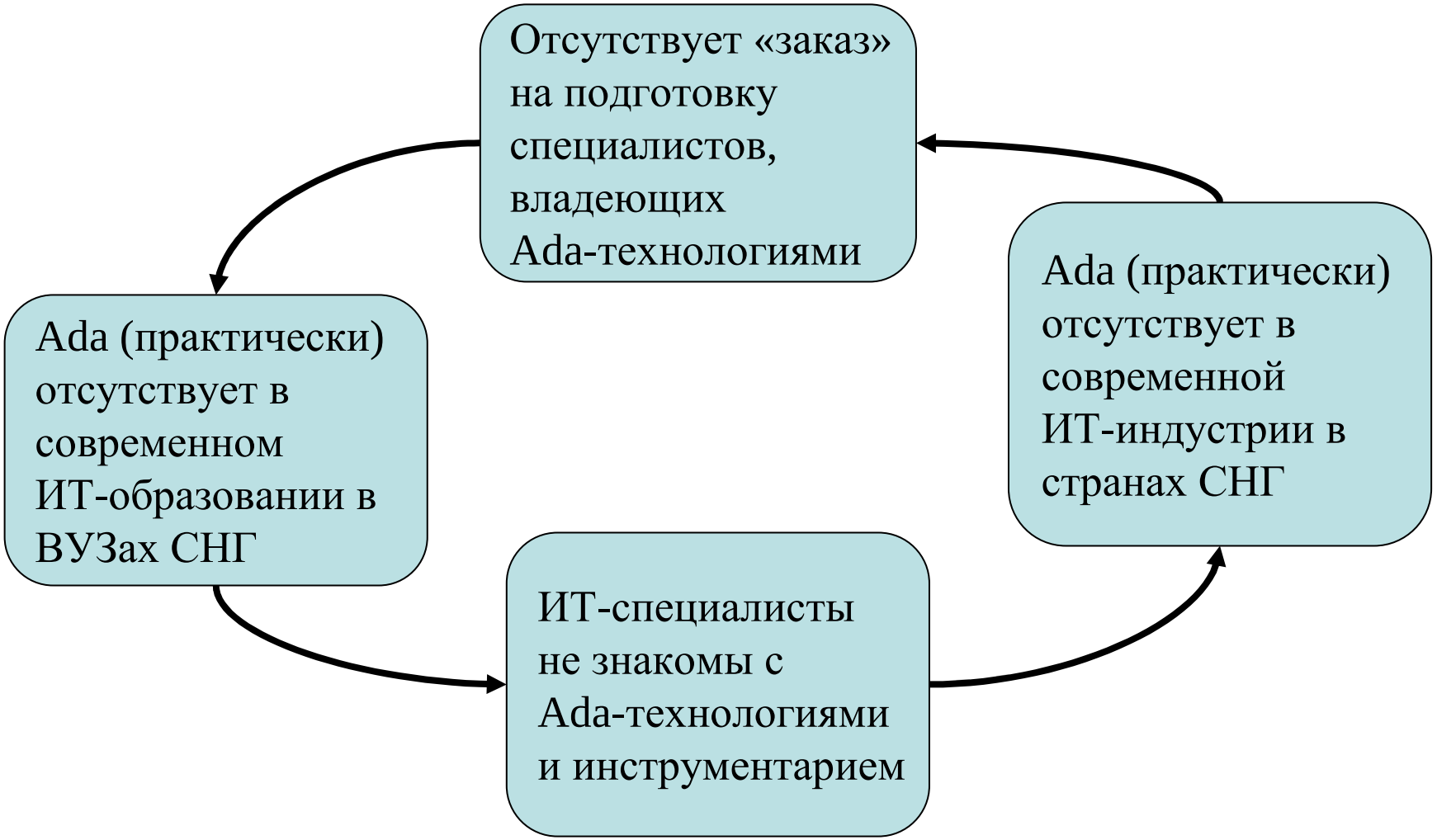
Некоторые индустриальные приложения написанные на Ada за рубежом

- Критичные по отношению к бизнес-процессам
  - Canal+ Technologies: Плата-за-просмотр. Управление доступом
  - BNP: Язык принятия решений в торговле
  - Philips: Линия по производству полупроводников
  - Хельсинский радиотелескоп

- Критичные по отношению к решаемой задаче
  - Astree: Трансевропейская система передачи сигналов на железной дороге
  - Weirton Steel – управление сталелитейным производством
  - Электронные деньги Mondex
  - Сканирующий электронный микроскоп

- Критичные по отношению к безопасности
  - Аэробус Airbus A340
  - Аэробус Boeing 777
  - Российский самолет-амфибия Бе-200

Еще одна попытка разорвать порочный круг…

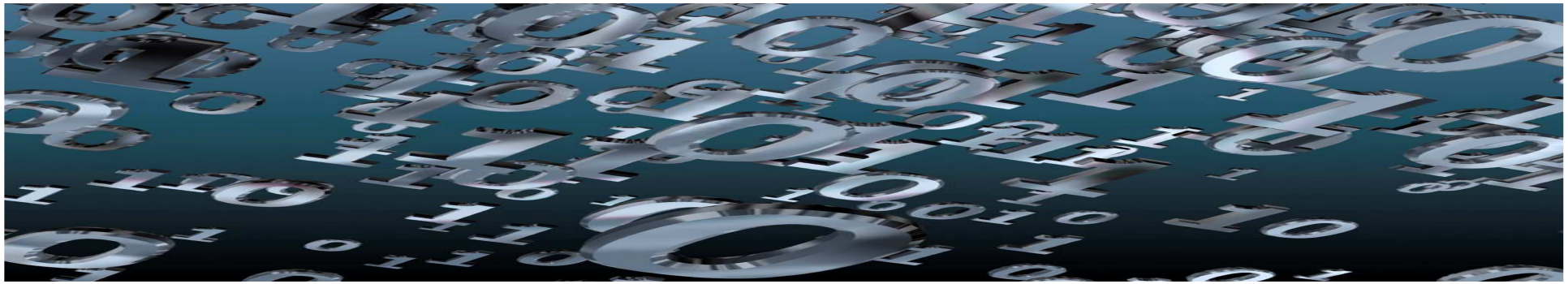


И это в ситуации, когда Ada используется

- В таких областях, как:
  - организация воздушного движения
  - авионика (военная и гражданская)
  - средства связи и телекоммуникаций
  - энергетика
  - банковские системы
  - медицинские системы
  - военные (наземного, морского и авиакосмического базирования)
  - космические технологии
  - телевидение
  - транспорт
  - электроника
  - …
- Такими компаниями, как:
  - Alenia
  - Alstom Transport
  - Ansaldo STS
  - BAE Systems
  - Boeing
  - EADS
  - European Space Agency
  - Eurocontrol
  - IPESOFT
  - JEOL
  - Lockheed Martin
  - MBDA
  - Philips Semiconductor
  - Raytheon
  - Rockwell Collins
  - SAAB
  - General Electric
  - Thales
  - Thales Alenia Space
  - ...

Почему Ada оказывается эффективным решением для образования?

- По тем же самым причинам, по которым Ada – эффективное решение для индустрии:
  - язык построен систематическим образом с опором на целостную философию, ключевым моментом которой является надежность программных услуг;
  - предоставляя больше возможностей, язык оказывается проще в изучении и использовании, обладая к тому же ясным и легко читаемым синтаксисом;
- Возможность (бесплатного) использования в учебном процессе индустриальной Ada-технологии (GNAT Academic Program);
- Простота адаптации учебных программ и курсов, основанных на Паскале.



СПАСИБО ЗА ВНИМАНИЕ

Media Scan by
Наука, разработка, внедрение