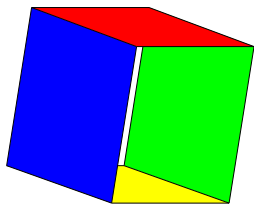


**Компонентный Паскаль/Оберон
как технологическая основа
единой системы вводных курсов информатики
и школьной ИТ-инфраструктуры**

Томск, 2007-07-02

— Благодарности.

Рабочая попытка сформулировать тезисы, обобщающие опыт общественного проекта **Информатика-21** (с 2002; см. сайт -- искать в яндексе) — попытка предложить решения проблем, поставленных в докладе Ю.Л.Костюка (или хотя бы подход с правильной стороны ["игра из общих соображений" за невозможностью просчитать варианты, см. ниже]).



Главная идея: возможность построить на основе Оберона/Компонентного Паскаля

сквозную (школа-университет)

общую (все учащиеся -- от математиков до лингвистов)

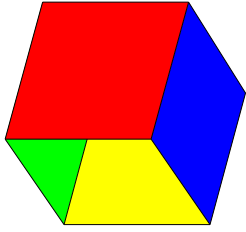
систему

преподавания основ информатики.

— За 5 лет проекта не было ни серьезных возражений по-существу, ни предложено серьезных альтернатив.

Постоянно имеется в виду образец системы математического образования.

Проект естественным образом вырос **из практических нужд**, систематически отрефлексированных с учетом достижений computer science и анализа тенденций в ИТ-индустрии.



во всех дискуссиях все время приходится ссылаться на сл.:

Два уровня ума

сложные задачи 2) абстрагирующий (организующий, "игра на основе общих соображений")

средние задачи 1) комбинаторный (нагромождающий, ориентированный на конкретные "бананы")

простые задачи 0) "запоминательный" (какую кнопку нажать? не ум, а условный рефлекс)

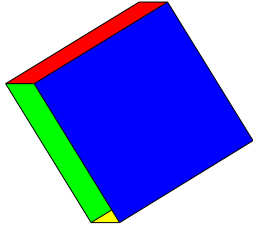
проявление в сфере ИТ:

программисты (1) <--> архитекторы (2)

в науке (Томас Кун "Структура научных революций" 1964):

"нормальная наука" (1) <--> смены парадигм (2)

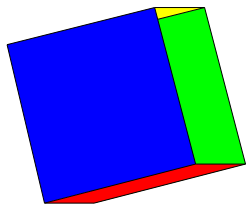
ИТ: разгул для умов комбинаторного типа, поэтому особо важен:



Принцип Калашникова:

Избыточная сложность = уязвимость.

- (почти) как закон сохранения или возрастания энтропии
- за избыточную сложность рано или поздно придется платить
- трудность в определении избыточности (мифы; непонимание задач)
- не эксплицирован, в ИТ постоянно нарушается, т.к. доминирует молодежь, которая не чувствует долгосрочных последствий, в т.ч. ПК
- **бояться избыточной сложности как огня**, для любой доп. сложности должно быть серьезное обоснование
- **вколачивать** в головы программистов, архитекторов **и менеджеров**
- грамотная политика: выдавливать избыточную сложность откуда возможно



Второй фундаментальный принцип, который постоянно нарушается:

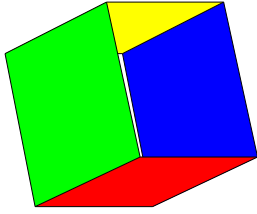
Разделяй и властвуй.

примеры:

- процедуры и модули
- статическая типизация (!)
- удаление из языка (в библиотеки) всех сложных фич (!!)
- автоматическое управление памятью как средство развязать модули, разделяющие сильно динамические структуры данных
- ...

примеры нарушения:

- курс введения в программирование на C++ (**NB**)
- курс численных методов, использующий C (**NB**)



Чему учить всех

Информатика

(фунд., почти раздел математики)

#

ИТ

(прикл.)

должны быть разведены (ср. замечания И.Л.Фукс по ЕГЭ)

"Фундаментальная информатика":

логика

структурирование информации

алгоритмика

цифровое представление аудио-, видео- данных

....

и т.п. общие, вечные вещи

языки

базовая

грамотность

информатика

точность всевозможных

описаний

математика

комбинаторное

мышление

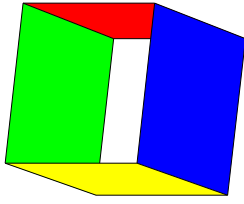
умо-структурирующие дисциплины

(за убедительной аргументацией — к И.Л.Фукс)

Фундаментальная информатика — вокруг
программирования, очищенного от всего случайного.

Вот мы и пришли к Оберону/К. Паскалю

На самом деле даже в чисто профессиональном плане имеются нетривиальные **долгосрочные** преимущества от инструментов и техник, очищенных от всего случайного.



Чему НЕ учить всех

всякого рода "мусорные" знания

знания бывают фундаментальные, прикладные, а также мусорные, коих особенно много порождается в сфере ИТ, состоящей из артефактов, нагроможденных "мастерами комбинаторного мышления с коротким горизонтом планирования"

Итак:

Оберон/Компонентный Паскаль = очищенный Паскаль

проще старого Паскаля

нет DISPOSE

нет BEGIN .. END

нет перечислимых типов

...

```
MODULE A; (* пример для Томска, 2007-07-03 *)
```

```
  IMPORT Log := StdLog, Math, In := Eps21SysIn(*Info21sysIn*);
```

```
  PROCEDURE Do*;
```

```
    VAR x: REAL;
```

```
  BEGIN (* единственный begin -- открывает тело процедуры *)
```

```
    In.Open;
```

```
    In.Real( x ); (* прочесть из входного потока *)
```

```
    WHILE x > 0 DO
```

```
      Log.Real( x ); (* печать в Log -- рабочий журнал *)
```

```
      Log.Real( Math.Sqrt( x ) );
```

```
      Log.Ln;
```

```
      x := x - 1
```

```
    END;
```

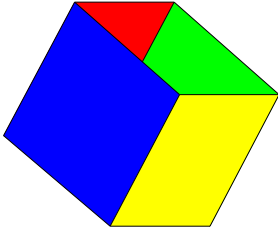
```
  END Do;
```

```
END A.
```

! A.Do 3.14159 ▾ входной поток данных для ввода

Оберон/КП = общий делитель всех совр. императивных ЯП.

Прямая дорога к Java, С# (сделаны под сильнейшим влиянием Оберона).



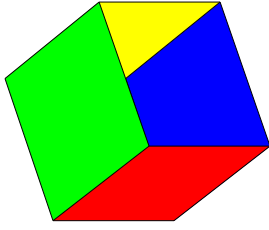
среда **Блэкбокс**:

- бесплатна и открыта
- проста и удобна (проверено уже многими! вся Витебская область переходит...)
- русифицируется вплоть до сообщений компилятора ("Важно!" -- И.Л.Фукс)
- "практически идеальна для обучения программированию" (Т.Евсикова, Институт математики и информатики, Вильнюс)
- адаптируема и настраиваема как пластилин (см. презентации)

Более чем полноценная замена Турбо Паскалю и Дельфи в школах и на младших курсах университетов.

NB А также может заменить MS Word для всех школьных нужд.

Осенью полная версия для линукса (без-GUI-ный вариант давно работает).



Система вводных курсов

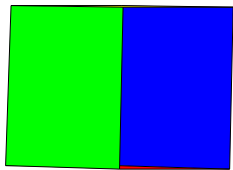
- 4-7 классы: "черепашки" и т.п.
- 8-11 классы: основы прог-я (грубо говоря, все, что нужно для четкого понимания схемы линейного поиска)
- 1-2 курсы ун-тов: основные алгоритмы, ООП, основные паттерны

На этот стержень можно навесить любые с/к в плане профподготовки (синтаксис и особенности С и т.п.).

Идеальный инструмент для физпрактикумов, курсов численных методов, мат. статистики и т.п.

Масса народа попробовала и согласилась.

Важно: сквозная система, единый язык в пособиях и т.п., как в математике, все друг друга понимают, эффективная конкуренция по качеству учебников, методик и т.п.



Еще раз спасибо организаторам!