

Владимир Биллиг

*профессор кафедры информатики Тверского  
государственного университета*

*VLADIMIR.BILLIG@TVERSU.RU*

Преподавание Объектно-Ориентированной  
Технологии Программирования

# Основные положения

- Программистская составляющая курса информатики
- ОО проектирование иОО программирование –основной метод современной программной инженерии
- Объектный подход. Когда начинать?
- ОО подход иОО язык.

# ОО язык и среда. Три кандидата

- Язык C# и среда Visual Studio .Net
- Язык Eiffel и среда Eiffel EnViSioN 2.0
- Язык Visual Basic и среда Microsoft Office

# Кандидаты. Достоинства и недостатки

- C#. Достоинства

- Перспективный язык разработки современных систем
- Изначально объектный язык
- Ориентирован на компонентное программирование
- Открытая многоязычная среда, облегчающая интеграцию.

- C#. Недостатки

- Не поддерживает всех возможностей объектного подхода
- Нет учебников, ориентированных на начальный курс обучения

# Eiffel

- Достоинства
  - Тщательно спроектирован Бертраном Мейером. Не отягощен прошлым наследием.
  - Поддержан авторской книгой “Object-Oriented Software Construction”
  - На его основе автор читает начальный курс информатики в ЕТН, Цюрих.
  - Доступна академическая версия языка и среды EnViSioN 2.0 : [www.eiffel.com](http://www.eiffel.com)
- Недостатки
  - Менее популярен, как практический язык разработки

# Офисное программирование

См. доклад на предыдущей конференции:  
«ОФП и его роль в образовании»

<http://www.it-education.ru/archive/2003/reports/presentation/billig.ppt>

# О сути подхода. Основные понятия

- Классы. Две ипостаси – Тип и Модуль
- Встроенные классы
- Семейства классов. Отношения. Клиенты и наследники
- Повторное использование
- Динамическое связывание. Полиморфизм.

# Объектное проектирование

- Абстрактные классы
- Универсальные классы
- Ограниченная универсальность

# Проектирование по контракту

- Качество программ и спецификации
- Система как совокупность классов, связанных контрактами
- Корректные программы и триады Хоара
- Корректность класса
- **Определение: Корректность класса**

Класс  $C$  корректен по отношению к своим утверждениям, если и только если:

C1. Для любого правильного множества аргументов  $xp$  процедуры создания  $p$ :

$\{ \textit{DefaultC} \textbf{ and } \textit{prep}(xp) \} \textit{Body}_p \{ \textit{post}_p(xp) \textbf{ and } \textit{Inv} \}$

C2. Для каждой экспортируемой программы  $r$  и для любого множества правильных аргументов  $xr$ :

$\{ \textit{prer}(xr) \textbf{ and } \textit{Inv} \} \textit{Body}_r \{ \textit{postr}(xr) \textbf{ and } \textit{Inv} \}$

# Исключительные ситуации. Когда контракт нарушается

- Что такое исключительная ситуация по Мейеру?
- Как обрабатывать исключительные ситуации.
- Два исхода – повторение и возврат управления вызывающей программе

# Классы и события

- Модель событий в C#

# Детали C#

- Указатели и автоматическая сборка мусора (Garbage Collector)
- Ссылочные и значимые типы
- Boxing и Unboxing
- Иерархия классов и прародитель Object
- Встроенные классы. Пример строковых классов – класс регулярных выражений
- Структуры, интерфейсы, классы.
- Наследование.

# Эйфель. Детали

- Факторы: корректность, устойчивость, расширяемость, повторное использование, совместимость, эффективность, переносимость, простота использования, функциональность, своевременность.
- Критерии (Метод и язык): Бесшовность, Классы (как модули, как типы), Утверждения, Механизм вычислений. Скрытие информации, Обработка исключений, Статическая типизация, Наследование, Переопределение, Полиморфизм, Динамическое связывание, Отложенные классы, Универсальность, Ограниченная универсальность. Управление памятью и сборка мусора.
- Критерии (Реализация и окружение): Автоматическое обновление, Быстрое обновление, Сохраняемость, Документирование, Просмотр и поиск.
- Критерии (Библиотеки): Базисные библиотеки, Графические и пользовательские интерфейсы

# 5 критериев, 5 правил, % принципов

Метод проектирования, претендующий на звание «модульного» должен удовлетворять следующим критериям, правилам, принципам:

- Критерии: Декомпозиции, Композиции, Понимаемости, Непрерывности, Защищенности
- Правила: Прямое отображение, Минимум интерфейсов, Слабая связность, Явные интерфейсы, Скрытие информации
- Принципы: Лингвистических единиц, Самодокументирование, Единый доступ, Открыт-Закрит, Однозначный выбор.

# Итоги

Одна цитата (из предисловия к русскому изданию книги Б. Мейера):

Слишком часто, объектно-ориентированные механизмы использовались просто как некоторый способ обтесывания программы, а не как систематический подход ко всему процессу разработки программного продукта в бесшовной модели, покрывающей не только реализацию, но и этапы анализа, проектирования, сопровождения. Упор на качество компонентов должен быть особенно важен для российских читателей. Россия может поддерживать свое лидерство в глобальном оффшорном рынке только при условии представления точной и документированной гарантии качества, на самом высоком уровне сертификации для программных продуктов и компонентов, которые она производит.

# Итоги

- OO Метод следует вводить в начальном курсе информатики с тем, чтобы он стал основой во всем процессе обучения инженерии программ.
- Для программистов реальными кандидатами в поддержке OO метода могут быть языки C# и Eiffel
- Офисное программирование может быть положено в основу обучения студентов других специальностей.

# Литература

- В.А. Биллиг, М. И. Дехтярь «VBA и Office 97. Офисное программирование», Русская Редакция, 1998
- В.А. Биллиг «VBA в Office 2000. Офисное программирование», Русская Редакция, 1999
- В.А. Биллиг «Средства разработки VBA-программиста. Офисное программирование, т.1», Русская Редакция, 2001
- В.А. Биллиг «Мир Объектов Excel 2000 Средства разработки VBA-программиста», Русская Редакция, 2001

Email: [Vladimir.Billig@tversu.ru](mailto:Vladimir.Billig@tversu.ru)