ДИСТАНЦИОННОЕ ОБУЧЕНИЕ

Степулёнок Денис Олегович

ПРЕДМЕТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
(тезисы к видеолекции)

Большинство читателей наверняка слышали о структурном программировании и об объектно-ориентированном программировании.

Возникает вопрос, что же дальше? Как, в какую сторону будут развиваться языки программирования?

Недостижимым идеалом является слияние естественного человеческого языка и искусственных языков.

Предметно-ориентированное программирование – бурно развивающееся сейчас направление в развитии языков программирования.

Строго говоря, деление языков программирования на языки общего назначения и предметно-ориентированные весьма условно, особенно, если учесть, что любой протокол или формат файлов является языком.

В лекции рассказывается о том, что такое предметно-ориентированное программирование и в чём его отличия от других парадигм.

Предметно-ориентированный язык программирования (ПОЯ) – (англ. *domain-specific programming language, domain-specific language, DSL*) – язык программирования, специально разработанный для решения определённого круга задач, в отличие от *языков программирования общего назначения*, например C или Java, или *языков моделирования общего назначения* наподобие UML.

ПОЯ могут быть:

– графическими (процесс программирования – «рисование» схемы в специальном редакторе);

– текстовыми (программирование – составление текста на некотором формальном языке).

Кроме того, ПОЯ можно разделить на:

– статические – языки, в которых не важно, в каком порядке программист рисует элементы схемы или составляет текст программы;

– динамические – имеет значение порядок действий (например, в Geometer's Sketchpad последовательность геометрического построения задаёт алгоритм).

Пути реализации ПОЯ:

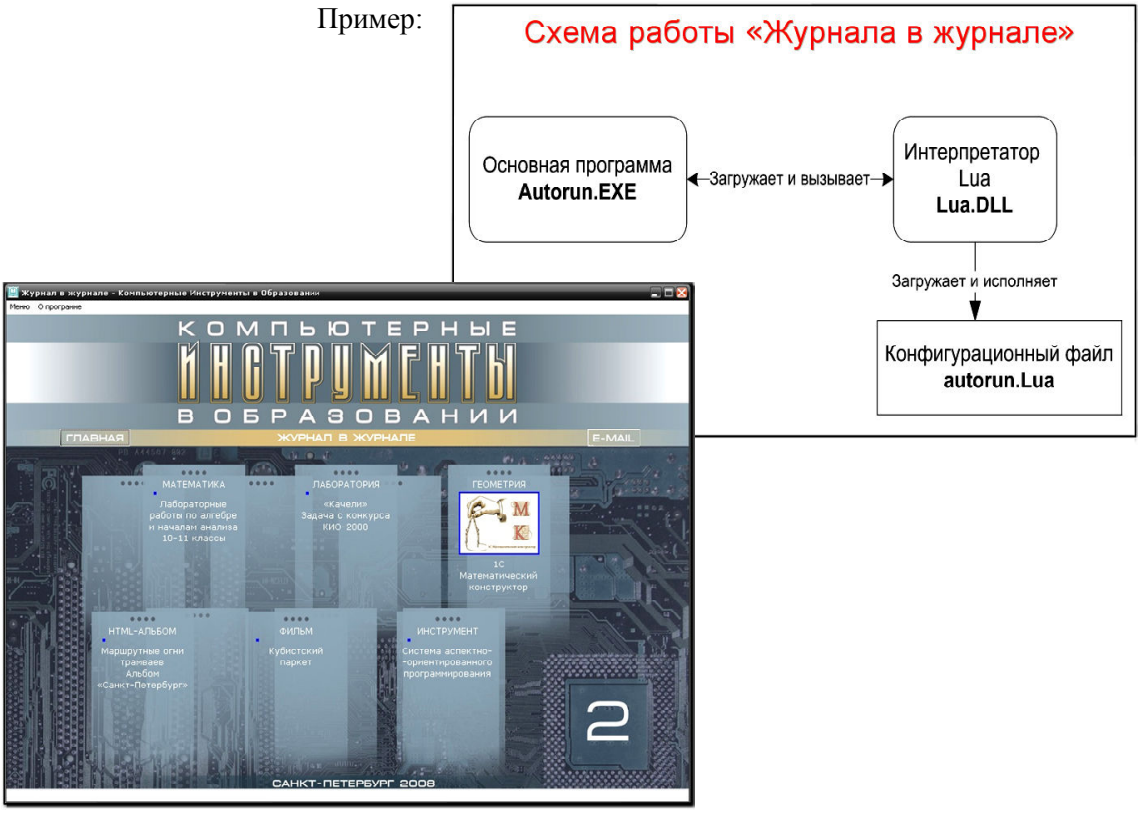
– использовать достаточно гибкий существующий универсальный язык программирования (C#, Java, Python) и добавить предметно-ориентированные возможности при помощи библиотеки компонентов (Framework'а);

– использовать существующую систему для создания предметно-ориентированных языков (Meta Programming System от JetBrains или DSL от Microsoft – позволяет рисовать графические схемы и генерировать по ним код);

– использовать среду программирования, в которой синтаксические конструкции языка можно модифицировать (настраивать процесс компиляции на определенную предметную область). Например, среда Phoenix, дополнение к компиляторам .Net**;**

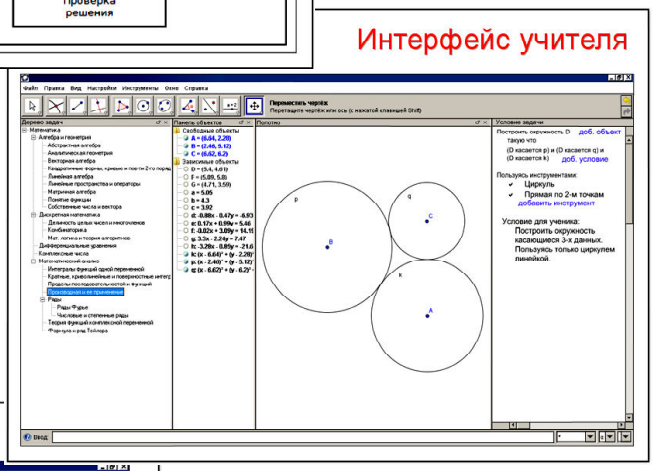
– создать собственный язык программирования с компилятором, отладчиком и т. д. Для генерации лексического анализатора можно использовать продукты **lex** и **yacc**.

Пример:

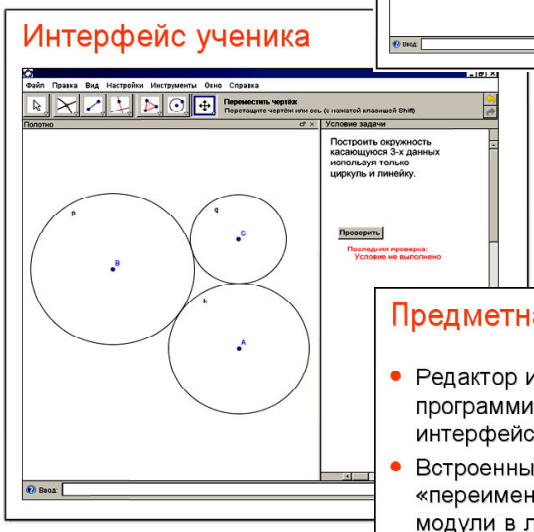


Пример:

Пример:



Пример:



Пример:

Предметная ориентированность – Delphi

- Редактор интерфейса позволяет визуально (без программирования) нарисовать большую часть интерфейса. Процесс создания нагляден.
- Встроенные средства рефакторинга позволяют «переименовать» классы, методы, компоненты, модули в любой момент когда вы обнаружите несоответствие реального использования класса, метода, модуля и представления о нём.
- Среда разработки генерирует шаблон метода при выборе события в редакторе свойств, нужно писать только само тело обработчика.

Степулёнок Денис Олегович, ассистент кафедры АСОИУ факультета Компьютерных технологий и информатики СПбГЭТУ «ЛЭТИ».