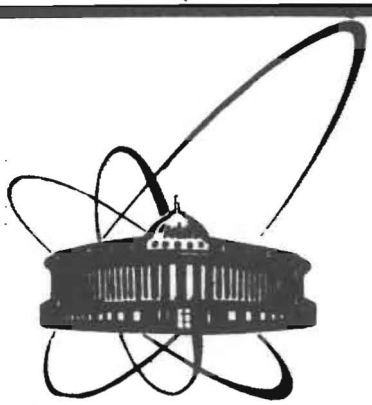


объединенный
институт
ядерных
исследований
дубна

Р11-88-721

В.Шуберт

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направлено в Оргкомитет II Международного научного семинара "Теория и применение искусственного интеллекта", НРБ, 29 мая - 2 июня 1989 г.

1988

I. Введение

Объектно-ориентированное программирование можно рассматривать как путь к владению большими программами, с учетом жизненного цикла программного обеспечения. Признаками таких программ являются, согласно Поспелову[I]:

- уникальность,
- отсутствие формализуемой цели существования программы,
- динамичность,
- неполнота описания.

В настоящее время по образцу системы программирования smalltalk [2] разработан ряд систем, использующих объектно-ориентированные диалекты таких классических языков, как Pascal, C, LISP, PROLOG [3], а также FORTRAN [4].

Цель настоящей работы состоит в исследовании объектно-ориентированных принципов программирования и их использовании для расширения фортрануориентированной технологии (Ф - технологии)[6].

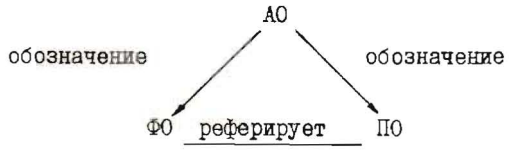
2. Основные понятия

Программа, написанная по принципам объектно-ориентированного программирования, построена из кирпичей. Эти кирпичи называют программными объектами (ПО). Программа в целом также представляет собой ПО. ПО обладают двумя важными свойствами:

-каждый ПО имеет название,

-каждому ПО соответствует некоторый физический объект (ФО), являющийся частью предметной области, на которую должен действовать результат работы программы. Предметная область в целом также представляет собой ФО.

Роль посредника между предметной областью и программой, т.е. между ФО и ПО, играет концептуальная модель, называемая абстрактным объектом (АО)[8].



АО отражает структуру и поведение, как ФО и ПО.
Модель структуры будем называть "структором" и модель поведения - "актором".

Базой для объяснения обеих моделей служит следующая простая модель жизненного цикла программного обеспечения.

В процессе проектирования получается АО как модель ФО. Реализация АО в ПО - это результат кодирования. Результат обработки ПО на ЭВМ позволяет использовать его ФО с целью изменения нескольких параметров относительно некоторой функциональной отметки. Совокупность АО и реализующих их ПО представляет собой архитектуру программного обеспечения.

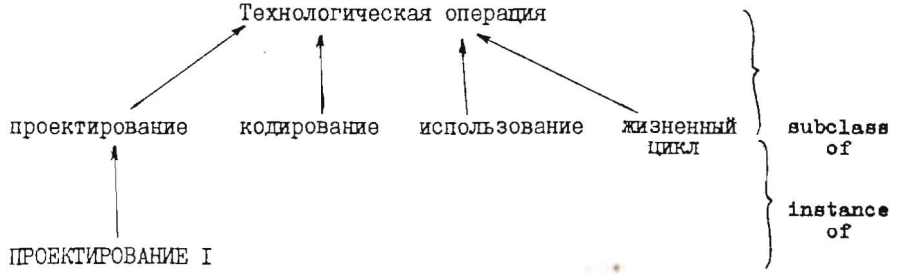
2.I. Модель структуры - "структор"

Конструкция модели статических свойств объекта (структора) начинается с определения терминов. Термин - это название объекта (индивидуальный термин) или класса объектов (обобщенный термин), например,

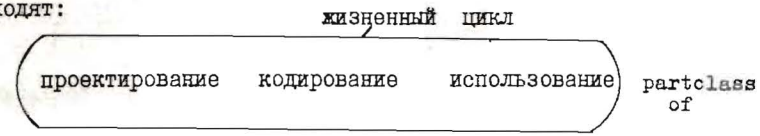
ПРОЕКТИРОВАНИЕ I - для конкретного процесса,

проектирование - для всех процессов этого вида.

Между терминами существуют отношения:



т.е. "кодирование" является одним из видов понятия "технологическая операция". "ПРОЕКТИРОВАНИЕ I" представляет собой экземпляры понятия "проектирование". Эти отношения называем вертикальными отношениями. Кроме того, существуют горизонтальные отношения. В состав этих отношений входят:

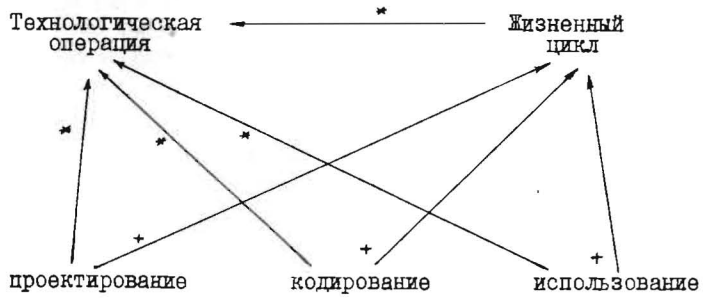


т.е. "проектирование", "кодирование" и "использование" принадлежит к понятию "жизненный цикл". В частном случае "ПРОЕКТИРОВАНИЕ I", "КОДИРОВАНИЕ 7" и "ИСПОЛЬЗОВАНИЕ 4" представляют собой части целого "ЖИЗНЕННЫЙ ЦИКЛ 3".

2.2. Модель поведения - "актор".

Понятие "актор" вводил HEWITT /9/, как модель системы экспертов. Эксперты общаются друг с другом с помощью обмена сообщениями. Цель обмена состоит в решении некоторой проблемы. Каждый актор знает какую помощь могут оказать соседние акторы при решении данной проблемы. Такое отношение называется "быть потенциальным помощником" (acquaintances).

Например:



Знает как:

a —*→ b b обеспечит соблюдение норм для a

a —+→ b b организует взаимодействие для a

Поведение экспертов описывается логико-трансформационными правилами /I/. Актор a реагирует на полученное им сообщение m передачей сообщения m' . Кроме того, сообщение m может вызвать некоторое изменение с знания актора a .

Соответственно, логико-трансформационное правило запишется в форме

$$a \; ; \; m \Longrightarrow m' \; ; \; c \; .$$

В терминологии объектно-ориентированного программирования эти правила называются "методы".

Пример логико-трансформационного правила:

проектирование; день окончания ПРОЕКТИРОВАНИЕ I ?
$\Longrightarrow$ I9ого июля I988

если знание актора "проектирование" содержит экземпляры:
ПРОЕКТИРОВАНИЕ I,
ПРОЕКТИРОВАНИЕ 2,
ПРОЕКТИРОВАНИЕ 3.

ПРОЕКТИРОВАНИЕ I окончилось I9-го июля I988 г. Это знание и помещается в ответ на запрос.

В объектно-ориентированном программировании и термин, и "актор" объединяются в понятие "объект". Если соответствующий термин представляет собой обобщенный термин, вводится, кроме того, понятие " class ".

2.3. Связь между абстрактным объектом (АО) и программным объектом (ПО)

С конструктивной точки зрения ПО реализует АО. ПО - это модуль в смысле MODULA2 /7/. Представление знания в АО является однозначным или многозначным. В общем, можно сказать, что

- однозначное знание переводится в форму представления, очень близкую к машинному языку (базовые функции),
- многозначное знание интерпретируется на уровне коммуникации человека с ЭВМ.

Из этого следует, что ПО должен содержать 3 типа модулей:

. пакет процедур,
. интерпретатор сообщения,
. дистрибутор.

3. Система программирования CLASSIC

Автором разработана система CLASSIC , которая реализует представленную архитектуру объектно-ориентированных программ. Базовым языком является СДФ /5/. CLASSIC работает в операционной системе ОС ЕС, с использованием системы TERM /I0/. Система CLASSIC позволяет построить практически любое число классов, включая определение отношений между классами и модификацию методов классов. В настоящее время система имеет 9 классов и требует приблизительно 400К байтов внутренней памяти. Система опробована на модельных задачах.

ЛИТЕРАТУРА

I. Поспелов Д.А. Ситуационное управление. М.:Наука, I986.
2. Goldberg A., Robson D. Smalltalk-80: The language and its implementation, Addison - Wesley, Reading (Massachusets), 1983.
3. Schmucker K. Object - oriented languages for the Macintosh - An overview of the languages and their capability. Byte 11(1986)8, p.177-185.
4. Bettels J., Meyers D.R. The PIONS Graphics System.CERN-Data Handling Division DD/86/6, March 1986, p.11-13.
5. Корнейчук А.А. Структурный диалект фортрана. ОИЯИ, РII-80-382, Дубна, I980.
6. Железнова К.М., Корнейчук А.А., Шарапова Э.В. Инструментальный программный комплекс, поддерживающий фортраноориентированную технологию программирования на ЭВМ ЦВК ОИЯИ. ОИЯИ, РII-85-905, Дубна, I985.
7. Wirth N. Programming in Modula-2 Springer-Verlag, Berlin Heidelberg New York Tokyo, 1985 М. : Мир , 1987.
8. Тамм Б.Г., Цуусепп М.Э., Таваст Р.Р. Анализ и моделирование производственных систем. М.: Финансы и статистика, I987.
9. Hewitt C. Viewing control structures as pattern of passing messages.AI, 8(1977) , p.323-364.
IO. Кореньков В.В. Интерактивный режим работы в диалоговой системе ТЕРМ на ЕС ЭВМ. ОИЯИ, РII-84-3I6, Дубна, I984.

Рукопись поступила в издательский отдел
30 сентября I988 года.

Шуберт В. Р11-88-721

Объектно-ориентированная архитектура программного обеспечения

Представляется модель архитектуры объектно-ориентированного программного обеспечения. Модель обобщает концепцию, реализованную в системе программирования Smalltalk. Она позволяет представить статические и динамические свойства как программы, так и предметной области, на которую должен действовать результат работы программы. Представляется система программирования, позволяющая реализовать конструкцию программного обеспечения соответственно предлагаемой модели.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна 1988

Перевод М.В.Коротковой

Schubert W. P11-88-721

Software Object-Oriented Architecture

The architecture model of the oject-oriented software is given. This model generates the conception being utilized the programming system SMALLTALK. It enables one to represent the statistic and dynamic properties of both the program and the application domain which the result of program acting must work on. The programming system CLASSIC that permits using the software type in accordance with the model suggested is presented.

The investigation has been performed at the Laboratory of Computing Techniques ans Automation, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna 1988