

4. При промежуточном и итоговом тестировании тесты закрытого типа необходимо дополнить другими видами тестов, так как тесты закрытого типа не позволяют оценить более высокие уровни усвоения знаний, а следовательно, оценить знания студентов объективно.

### **Библиографический список**

1. *Анастаси А., Урбина С.* Психологическое тестирование. СПб., 2002.
2. *Дрейпер Н., Смит Г.* Прикладной регрессионный анализ: В 2 кн. / Пер. с англ. Ю. П. Адлера, В. Г. Горского. 2-е изд., перераб. и доп. М., 1986.
3. Педагогика: педагогические теории, системы, технологии: Учеб. для студентов высш. и сред. пед. учеб. заведений / *С. А. Смирнов, И. Б. Котова, Е. Н. Шиянов и др.*; Под ред. С. А. Смирнова. 4-е изд., испр. М., 2003.
4. *Реан А. А., Бордовская Н. В., Розум С. И.* Психология и педагогика. СПб., 2002.
5. *Смирнов С. Д.* Педагогика и психология высшего образования: от деятельности к личности: Учеб. пособие для слушателей фак. и ин-тов повышения квалификации преп. вузов и асп. М., 1995.
6. *Смирнов С. Д.* Педагогика и психология высшего образования: от деятельности к личности: Учеб. пособие для студентов высш. учеб. заведений. М., 2001.
7. *Хан.* Статистические модели в инженерных задачах. М., 1969.

**Т. П. Телепова**

## **МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ ПЕДАГОГИЧЕСКОЙ СИСТЕМЫ**

В наше время стремительного совершенствования увеличение объема знаний влечет за собой усложнение процесса их передачи, который должен стать более эффективным и продуктивным. Использование только привычных, классических методов обучения уже не приводит к повышению качества обучения. Одним из методов интенсификации учебного процесса является применение компьютерных информационных технологий,

для которых характерны новая организация форм уроков и изменение роли и функций преподавателя.

Создаваемые в настоящее время программы в основном направлены на моделирование одной из функций педагогической системы, тогда как сам процесс обучения остается без тщательного анализа и возможного проектирования. Поэтому проблема разработки эффективных методов моделирования, позволяющих в наглядной форме анализировать изучаемую предметную область, разрабатывать приложения в соответствии с информационными потребностями пользователей в педагогических системах, остается актуальной и сегодня.

Методы моделирования широко применяются при проектировании программного обеспечения информационных экономических систем (ИЭС). CASE-средства (*CASE – Computer Aided Software Engineering*) используются в компьютерных технологиях создания и сопровождения программного обеспечения ИЭС. Большинство существующих CASE-средств основано на методах структурного или объектно-ориентированного анализа и проектирования, позволяющих использовать спецификации в виде диаграмм или текстов для описания внешних требований к проектируемой системе и архитектуре программного обеспечения. Сущность объектно-ориентированного подхода при проектировании программного обеспечения заключается в объектной декомпозиции системы, при этом статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы – в терминах обмена сообщениями между объектами.

Любую педагогическую систему, так же как и экономическую, можно назвать информационной, поскольку она обладает информацией. Поэтому к ней применимы методы моделирования, используемые в ИЭС. Но отличия в представлении информации вносят в процесс моделирования педагогической системы свои особенности.

Моделирование педагогической системы целесообразно начать с составления диаграммы вариантов ее использования, т. е. с последовательности действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (пользователем). В простейшем случае вариант использования – это функция, которую реализует проектируемая система. Рассмотрим методику проведения практических занятий по дисциплине «Арифметические и логические основы вычислительной техники». В ее основе лежит разработанная система учебных за-

дач, направленных на усвоение алгоритмов операций, происходящих в операционных устройствах вычислительной техники.

Основная задача методики – повышение эффективности проведения практических занятий за счет роста активности как студентов, так и преподавателя. Метод предполагает тесное творческое взаимодействие студентов с преподавателем и между собой. Преподаватель – главный консультант; он отвечает на вопросы, возникающие у студентов при решении задач, осуществляет контроль за ходом выполнения задания. Итоговый контроль повышает заинтересованность студентов в конечном результате своего труда, а применение принципа индивидуального выполнения задания способствует развитию личных качеств каждого учащегося.

Начальная диаграмма вариантов использования моделируемой системы выглядит следующим образом (рис. 1).

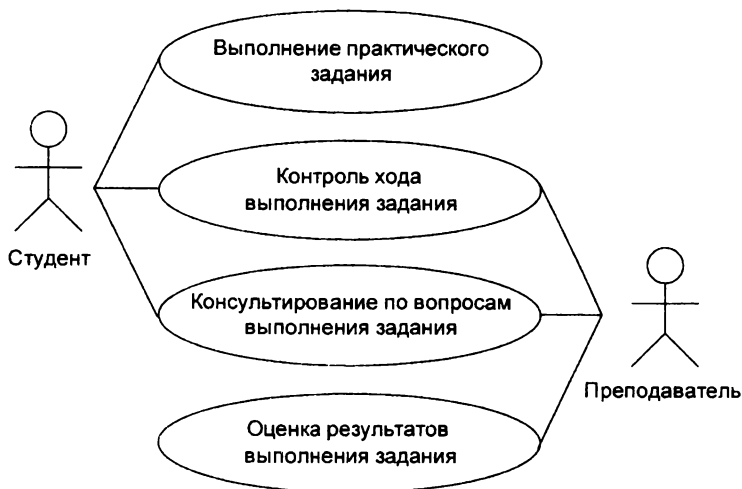


Рис. 1. Начальная диаграмма вариантов использования моделируемой системы

Все варианты использования (обозначены овалами) так или иначе связаны с внешними требованиями к функциональности системы. Если студенту требуется контроль хода выполнения задания, то это требование должно быть удовлетворено. Функция контроля выполнения задания и функция итогового контроля, по сути, являются составляющими одной

большой функции контроля, но выделены в разные варианты использования, так как реализуются различными методами.

Можно подойти к описанию вариантов использования с разной степенью детализации. Например, вариант использования «выполнение практического задания» для темы одного из практических занятий по дисциплине «Арифметические операции суммирования» можно разбить на два варианта: «сложить два числа, представленных в формате байта с фиксированной точкой» и «сложить два числа, представленных в формате двух байт с плавающей точкой». Каждый из вариантов использования сопоставляется с решаемой задачей данного занятия.

*Диаграмма классов* является центральным звеном объектно-ориентированных методов. Она определяет объекты проектируемой системы и связи, которые существуют между ними.

Объект определяется как осязаемая реальность – предмет или явление, имеющее четко определяемое поведение. Объект обладает состоянием, поведением и индивидуальностью. Структура и поведение схожих объектов определяют общий для них класс. Термины «экземпляр класса» и «объект» являются эквивалентными. Состояние объекта характеризуется перечнем всех возможных (статических) свойств данного объекта и текущими значениями (динамическими) каждого из этих свойств. Поведение характеризует воздействие объекта на другие объекты, и наоборот.

Определение классов объектов – одна из самых сложных задач объектно-ориентированного проектирования. Каждый класс должен иметь уникальное имя, отражающее характер абстракции, представляемой данным классом. Если классу трудно придумать краткое и содержательное имя, то это является признаком неудачного выделения класса. В ИЭС информация представлена в виде определенных данных (цена, количество, строка заказа, номер счета и др.) и методов их обработки, например, обработка заказов клиентов. Поэтому в ИЭС класс, как правило, определяется из уже существующих исходных данных.

В педагогической системе информация может быть представлена в виде конспекта лекций, методических указаний, заданий. Такую информацию трудно рассматривать в качестве данных при проектировании программного обеспечения. К тому же в информационной педагогической системе, как правило, не существует однозначно определенных методов обработки происходящих в этой системе событий. Например, контроль

правильности выполнения практического задания преподаватель может проводить по-разному. У одного студента педагог тщательно проверит ход выполнения задания и укажет ошибки, а у другого проверит конечный результат и попросит найти ошибки самому.

В этом случае при определении классов надо отталкиваться от функции, которые должна реализовывать проектируемая система, и на основе этого отбирать данные из общего потока имеющейся информации, необходимые для реализации этих функций.

В рассматриваемом примере основная функция системы – выполнение практического задания. Класс, с помощью которого будет реализовываться эта функция, назовем «Задание». Для обеспечения выполнения консультирующей функции создадим класс «Справка». В него войдет информация, используемая при решении задач. Для реализации функции контроля нам потребуется класс «Контроль», содержащий данные, необходимые для осуществления контроля за ходом выполнения задания.

Функцию оценки выполнения практического задания оставим за преподавателем. Но так как предыдущие функции студент будет реализовывать самостоятельно в качестве пользователя разрабатываемого программного обеспечения, преподавателю понадобятся данные класса «Статистика», на основе которых можно будет оценить результаты выполнения задания каждым студентом (рис. 2).

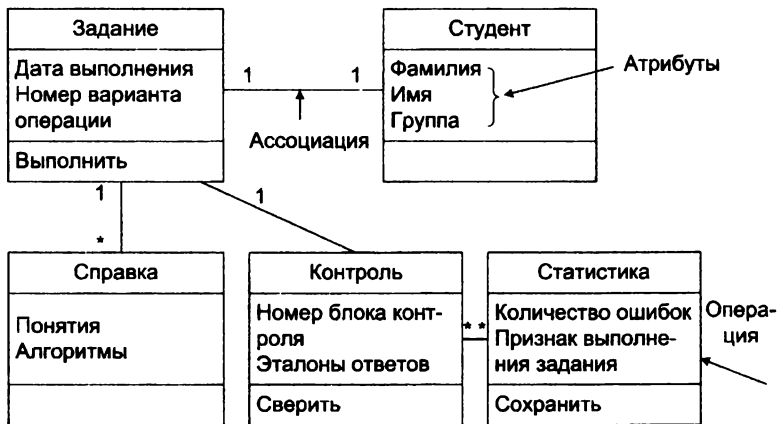


Рис. 2. Диаграмма классов исследуемой предметной области

Ассоциации представляют собой связи между отдельными понятиями предметной области (классами). Ассоциация выполняет две роли, каждая из которых показывает направление ассоциации. Таким образом, ассоциация между классами «Студент» и «Задание» происходит в двух направлениях: от «Студента» к «Заданию» и от «Задания» к «Студенту».

Символ «\*» означает, что с одним заданием может быть связано несколько понятий, тогда как каждый студент может выполнять только свой вариант задания (цифра 1).

Атрибуты описывают структуру класса, а операции – процессы, реализуемые классом.

Следует отметить, что данная диаграмма представляет собой обобщенную модель классов моделируемой системы, так как требует дальнейшей объектной декомпозиции. Так, класс «Задание» можно разделить на классы «Блок решения задачи» и «Операции, выполняемые при решении». Таким образом, класс «Задание» будет являться пакетом, включающим в себя взаимозависимые классы, полученные путем декомпозиции класса более высокого уровня (рис. 3).

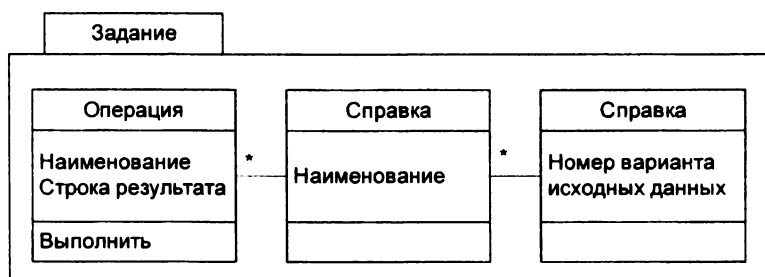


Рис. 3. Диаграмма пакета

*Диаграммы состояний* определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий.

На рис. 4 представлена диаграмма состояний, отображающая поведение объекта «Задача», в рассматриваемой системе выполнения практического задания по теме «Арифметические операции суммирования».

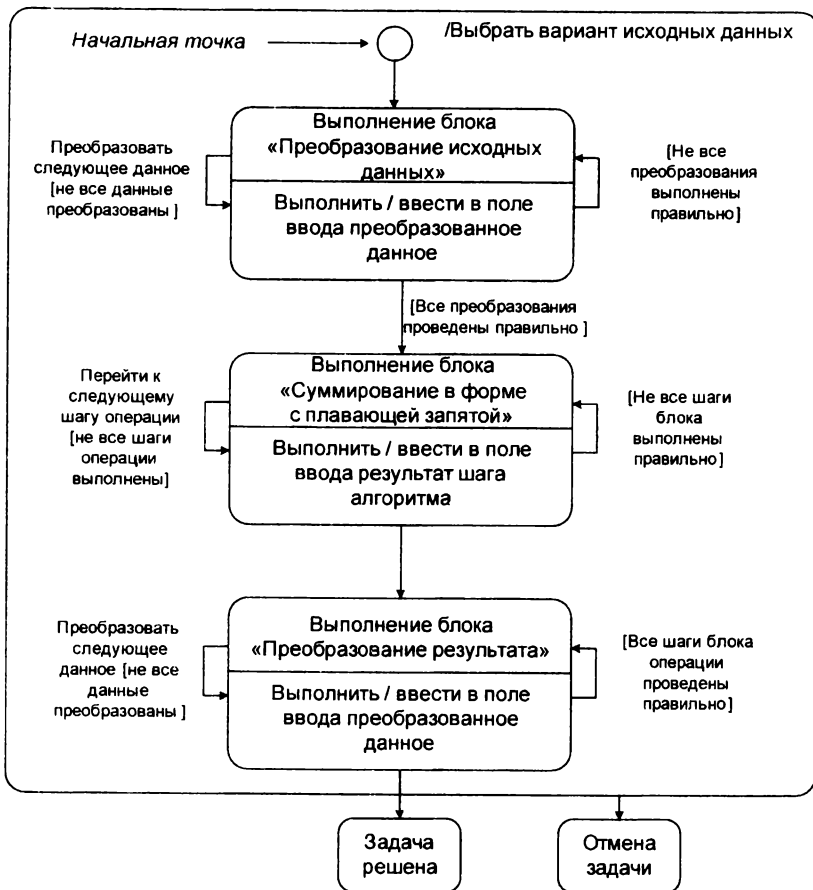


Рис. 4. Диаграмма состояний объекта «Задача»

Рассматриваемый алгоритм имеет линейную структуру, шаги выполняются в строго определенной последовательности. Как показывает практика, не следует осуществлять контроль после выполнения каждого шага задачи, так как в этом случае каждый раз прерывается логическая цепочка алгоритма, что снижает уровень усвоения изучаемого алгоритма. Алгоритм задачи лучше разделить на логически завершенные блоки. Каждый блок задачи – это последовательность действий, которые завершаются по-

лучением промежуточного результата. Например, чтобы преобразовать исходные десятичные числа в двоичные, представленные в форме с плавающей запятой (блок преобразования исходных данных), необходимо получить их двоичные эквиваленты, преобразовать в форму с плавающей запятой и записать полученные числа в 16-разрядную сетку.

На диаграмме показаны различные состояния, в которых может находиться объект «Задача». Процесс начинается с начальной точки, затем следует первый переход в состояние «Преобразование исходных данных». С этим состоянием связана деятельность «Выполнить /ввести в поле ввода преобразованное данное». Из данного состояния возможны три перехода:

1) если проведены не все преобразования данного блока, то происходит возврат в состояние «Преобразование исходных данных» и осуществляется следующее преобразование;

2) если хотя бы одно из преобразований выполнено неправильно опять происходит возвращение в исходное состояние и исправляются неправильно выполненные шаги;

3) если все преобразования проведены правильно, осуществляется переход к следующему состоянию.

Состояние «Суммирование в форме с плавающей запятой» также имеет три условных перехода, и переход в следующее состояние возможен только в том случае, если все действия данного блока выполнены правильно.

Состояние «Отмена задачи» позволяет в любой момент прекратить выполнение задачи.

Существуют также состояния, связанные с выполнением контроля каждого блока задачи (рис. 5). Переход из состояния «Проверка преобразованного данного» в состояние «Формирование статистики и счетчика ошибок» происходит, если состояние «Контроль» обнаружило, что введенное преобразованное данное не совпадает с эталонным, который содержится в массиве правильных результатов выполняемых преобразований.

Консультирующие функции моделируемой системы были отданы классу «Справка». При этом подразумевалось, что основная информация, необходимая для выполнения практического задания, содержится в конспектах лекций и методических указаниях. Поэтому структура класса представляет собой перечень основных понятий, используемых при решении



данной задачи и их краткое пояснение. Кроме этого, в класс «Справка» вошли алгоритмы задач, выполняемых на практическом занятии. Получить информацию из класса можно в любой момент при решении задачи.

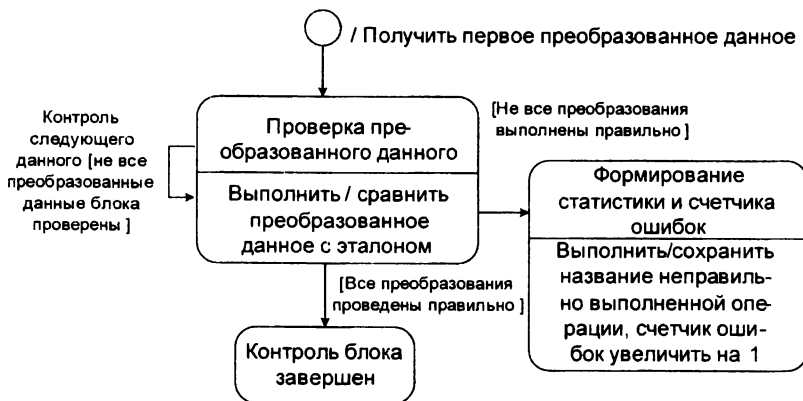


Рис. 5. Диаграмма состояний объекта «Контроль»

Диаграмма деятельности устанавливает порядок выполнения действий при решении любой задачи практического задания (рис. 6).

Если при описании поведения системы имеются параллельные процессы, то их необходимо синхронизировать. Диаграмма указывает на то, что эти виды деятельности могут выполняться параллельно или чередоваться. Так, при выполнении блока «Преобразование исходных данных» можно сначала просмотреть справочную информацию в блоке «Справка», а затем сделать преобразования или наоборот.

Диаграмма деятельности описывает поведение системы в рамках трех вариантов использования: выполнение практического задания, контроль хода выполнения задания и консультирования по необходимым вопросам.

Четвертый вариант использования «Оценка результатов выполнения задания» (см. рис. 1) реализуется самим преподавателем на основе формируемого в процессе контроля блока «Статистика», в который записывается информация по ходу решения задачи (название операций, в которых были сделаны ошибки и количество ошибок).



Рис. 6. Диаграмма деятельности

Построение диаграмм может происходить с различной степенью декомпозиции. Так, например, следовало бы подробнее остановиться на структуре каждого блока решаемой задачи (порядок выполняемых действий) и контроле выполняемых действий блока, поскольку возможны варианты контроля как всех выполненных действий, так и определенных или даже конечной операции блока.