

## ПРАКТИЧЕСКАЯ РАБОТА № 2

### Структурный подход к программированию.

#### Стадия «Эскизный проект»

*Цель работы:* научиться создавать формальные модели и на их основе определять спецификации разрабатываемого программного обеспечения.

#### Теоретическая часть. Разработка спецификаций

Разработка программного обеспечения начинается с анализа требований к нему. В результате анализа получают спецификации разрабатываемого программного обеспечения, строят общую модель его взаимодействия с пользователем или другими программами и конкретизируют его основные функции.

При структурном подходе к программированию на этапе анализа и определения спецификаций разрабатывают три типа моделей: модели функций, модели данных и модели потоков данных. Поскольку разные модели описывают проектируемое программное средство с разных сторон, рекомендуется использовать сразу несколько моделей, разрабатываемых в виде диаграмм, и пояснить их текстовыми описаниями, словарями и т.п.

Структурный подход к разработке ПС предполагает использование следующих видов моделей:

- диаграмм потоков данных (DFD — Data Flow Diagrams), описывающих взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе;
- диаграмм «сущность—связь» (ERD — Entity-Relationship Diagrams), описывающих базы данных разрабатываемой системы;
- диаграмм переходов состояний (STD — State Transition Diagrams), характеризующих поведение системы во времени;
- функциональных диаграмм (методика SADT);
- спецификаций процессов;
- словаря терминов.

#### Диаграммы переходов состояний

С помощью диаграмм переходов состояний можно моделировать последующее функционирование системы на основе ее предыдущего и текущего функционирования. Моделируемая система в любой заданный момент времени находится точно в одном из конечного множества состояний. С течением времени она может изменить свое состояние, при этом переходы между состояниями должны быть точно определены.

#### Диаграммы потоков данных

Для описания потоков информации в системе применяются *диаграммы потоков данных* (DFD — Data flow diagrams). DFD позволяет описать требуемое поведение системы в виде совокупности процессов, взаимодействующих посредством связывающих их потоков данных. DFD показывает, как каждый из процессов преобразует свои входные потоки данных в выходные потоки данных и как процессы взаимодействуют между собой. Диаграммы потоков данных, используя функции, описанные на уровне функциональной модели, **позволяют детализировать описание предметной области** за счет введения накопителей, потоков данных и внешних сущностей. **Накопитель (хранилище) данных** - приспособление для хранения информации, обладающее возможностью записи и извлечения данных. Способы доступа и хранения данных в накопителях в ходе анализа не уточняются. Хранилища являются прообразами файлов или баз данных. **Поток данных** - канал передачи данных от источника к приемнику. В качестве источников и приемников данных для потоков могут выступать внешние сущности, процессы и накопители. **Внешняя сущность** - объект, являющийся поставщиком и/или получателем информации. Например, «заказчик», «банк» и т.д. Внешние сущности обозначают источники и приемники, которые не представляют для анализа интерес в данный момент и служат для ограничения моделируемой части предметной области. Отражают взаимодействие системы с внешним миром.

#### Спецификации процессов

Спецификации *процессов* обычно представляют в виде краткого текстового описания, схем алгоритмов, псевдокодов.

#### Словарь терминов

*Словарь терминов* представляет собой краткое описание основных понятий, используемых при составлении спецификаций. Он должен включать определение основных понятий предметной области, описание структур элементов данных, их типов и форматов, а также всех сокращений и условных обозначений.

На основе модели потоков данных создается **словарь данных (Data Dictionary)**, в котором хранится и анализируется состав потоков и накопителей данных, взаимосвязь отдельных элементов потоков и накопителей данных. Например, при моделировании документооборота вводятся сведения о структуре и реквизитном составе документов.

### Диаграммы «сущность—связь»

Хранимые в словаре данных описания каждого накопителя (хранилища) данных используются для перехода к построению модели данных в виде **диаграмм «сущность-связь» (ERD)**. В отличие от функциональных диаграмм (IDEF0) и диаграмм потоков данных (DFD) диаграммы «сущность-связь» (ERD) **описывают информационное пространство**, в рамках которого реализуются процессы объекта предметной области. Выявляются и определяются элементы базы данных, в которых будут храниться данные системы. Выявляются и определяются их атрибуты и отношения. Модель данных должна быть привязана к функциональной модели: элементы модели данных и их атрибуты должны соответствовать накопителям данных. Диаграмма сущность-связь — инструмент разработки моделей данных, обеспечивающий стандартный способ определения данных и отношений между ними. Она включает *сущности* и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные *сущности* и *связи* между ними, которые удовлетворяют требованиям, предъявляемым к ИС.

### Порядок выполнения работы

1. На основе технического задания из практической работы № 1 выполнить анализ функциональных и эксплуатационных требований к программному продукту.
2. Определить основные технические решения (выбор языка программирования, структура программного продукта, состав функций ПП, режимы функционирования и т.д.).
3. Разработать архитектуру проектируемого программного средства. Для проектирования архитектуры программной системы использовать диаграммы потоков данных (DFD).
4. При построении диаграммы потоков данных (DFD) учитывать следующие правила
  - Размещать на каждой диаграмме от 3 до 7 процессов.
  - Избегать несущественных на данном уровне деталей.
  - Декомпозицию потоков данных выполнять одновременно с декомпозицией процессов (т.е., параллельно!).
  - Избегать аббревиатур, имена подбирать по существу.
  - Имена процессов должны быть глаголами или глагольными существительными. Имена подсистем должны быть существительными. Имена потоков должны быть названиями документов или групп документов.
  - Не дублировать определения функционально идентичных процессов, ссылаться на имеющееся, на более высоком уровне определения.
5. Определить объекты и их атрибуты для моделирования структур данных на основе построенной модели архитектуры системы (DFD-модели).
6. Добавить словарь терминов (данных).
7. Оформить результаты проектирования в виде эскизного проекта.
8. Представить отчет по работе преподавателю для защиты.

Полезные ссылки:

[http://www.prj-exp.ru/patterns/pattern\\_draft\\_project.php](http://www.prj-exp.ru/patterns/pattern_draft_project.php)