Тема 2.2 ***Методы и принципы программирования***

1. Методы программирования

При разработке программ используют ***3 метода*** их построения:

1) *Структурное программирование* – описывается каждое действие алгоритма для достижения конечного результата, используя процедурный стиль программирования и последовательную декомпозицию алгоритма решения задачи сверху вниз;
2) *Модульное программирование* – проектирование новой программной системы на базе разработанных и отлаженных ранее модулей. Этот метод многократного использования разработанного программного обеспечения.
3) *Объектно-ориентированное программирование* – описывает программные системы в виде взаимодействия объектов.

Объект – это совокупность данных и действий над ними. Каждый объект обладает свойствами. *Свойства* – это характеристики состояния объекта. Взаимодействие с объектом происходит через интерфейс, использующий *визуальные компоненты*.

Таким образом, структурное программирование легло в основу всех методов программирования.

В современных условиях широко развивается объектно-ориентированное программирование, которое имеет следующие достоинства: упрощение процесса проектирования программных систем, легкость их сопровождения и модификации, минимизирование времени разработки за счет многократного использования готовых модулей.

Этот метод используется в современных *средах разработки приложений*. При формировании общего вида главного окна при выполнении приложения и способов управления работой приложения для каждого компонента определяют внешний вид, размеры, способ и место размещения в области окна приложения. Компоненты разбиты на две группы:

1) *Визуальные компоненты* (элементы управления – кнопки, списки, переключатели, надписи) имеющие фиксированное местоположение и размеры. Они подразделяются на «оконные» (становятся активными для взаимодействия с пользователем) и «неоконные» (графические).
2) *Невизуальные компоненты* не имеют фиксированного местоположения и размеров (диалоговые окна открытия и сохранения файлов, таблицы баз данных).

При разработке процедур обработки событий необходимо запрограммировать реакцию на все возможные изменения состояния объектов.

2. Виды программного обеспечения. Общие принципы разработки ПО

Программное обеспечение (ПО) – совокупность программ, которые могут выполняться вычислительной системой.

Программное обеспечение различается по назначению, выполняемым функциям, формам реализации.

Программное обеспечение разделяется на ***3 вида***:

1) *Системное* (*базовое* – операционные системы; *сервисное* – программы диагностики работоспособности компьютера, программы обслуживания дисков, программы архивирования данных, антивирусные программы);
2) *Прикладное* (текстовые редакторы, электронные таблицы, системы баз данных, графические редакторы, программы проектирования, обучающие программы и др.);
3) *Инструментальное* (интегрированные среды разработки приложений, системы программирования, средства создания информационных систем).

При разработке ПО вне зависимости от его вида используют ***общие принципы***:

- *частотный принцип* – выделение действий и данных по частоте использования;
- *принцип модульности* – обособление составных частей ПО в отдельные модули по функциональному признаку;

- *принцип функциональной избирательности* – при проектировании ПО, объем которого превышает объем оперативной памяти, часть ПО, которая постоянно используется, хранится в оперативной памяти, другая часть – на внешних запоминающих устройствах;
- *принцип генерируемости* – осуществление настройки на конкретную конфигурацию технических средств, круг решаемых проблем, условия работы пользователя;
- *принцип функциональной избыточности* – возможность проведения одной и той же работы (функции) различными средствами;
- *принцип «по умолчанию»* - хранение в системе базовых описаний структур, модулей, конфигураций оборудования и данных, определяющих условия работы с ПО.

При создании и развитии ПО применяют также общесистемные принципы:

- *принцип включения* – требования к созданию и функционированию ПО определяются включающей его в себя системы;
- *принцип системного единства* – обеспечение целостности между подсистемами;
- *принцип развития* – возможность наращивания и совершенствования компонентов ПО и связей между ними;
- *принцип комплексности* – обеспечение связности обработки информации;
- *принцип информационного единства* – использование единых терминов, символов, условных обозначений и способов представления;
- *принцип совместимости* – обеспечение совместного функционирования всех подсистем ПО;
- *принцип инвариантности* – подсистемы и компоненты ПО универсальны по отношению к обрабатываемой информации.

3. *Жизненный цикл программного обеспечения*

Решение задачи на ЭВМ проходит по этапам и предусматривает возможность возврата на предыдущие этапы после анализа полученных результатов.

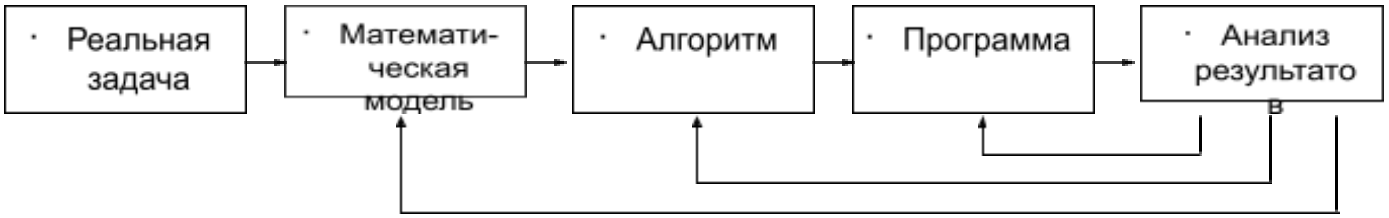

Рисунок 1. Технологическая цепочка решения задач на ЭВМ

Жизненный цикл ПО – это процесс от начала создания ПО до полного изъятия из эксплуатации.

Структура жизненного цикла содержит процессы, действия и задачи, которые должны быть выполнены во время создания ПО.

Жизненный цикл ПО состоит из следующих ***стадий***:

1) *Определение требований и спецификаций* – устанавливаются общие требования по надежности, технологичности, универсальности, эффективности, информационной согласованности, разрабатываются входные и промежуточные языки, формы выходной информации.
2) *Проектирование ПО* – формируется структура ПО, разрабатываются алгоритмы, устанавливается состав модулей и интерфейсы.
3) *Программирование* – проектные решения реализуются в виде программ.
4) *Отладка ПО* – проверка выполнения всех требований, всех структурных элементов системы, работоспособности ПО, выявление и исправление ошибок.

5) *Сопровождение ПО* – процесс координации всех элементов системы в соответствии с требованиями пользователя, внесения всех необходимых ему исправлений и изменений, дальнейшее совершенствование системы во время эксплуатации.

В процессе разработке ПО приходится осуществлять несколько итераций (последовательных приближений) к приемлемому варианту системы.