# Специальные требования к безопасности программного обеспечения

Вяльмискин М.В
Группа: 0587

# Содержание

1. Тестирование сложности системы
2. Принципы объектно-ориентированного подхода
3. Венгерская нотация
4. Обработка исключений
5. Защита от НСД

# 1. Тестирование сложности системы

**(дополнение к показателю С1 в ГОСТ 28195-99)**

Сложность системы, для которой нужно разработать программное обеспечение определяется по пяти признакам:

- - система может быть описана в виде иерархии, т.е. может быть разделена на подсистемы, которые тоже могут быть иерархичными;
- - в системе могут быть выделены подсистемы, для которых внутриэлементные связи сильнее, чем межэлементные;
- - низший уровень одной подсистемы достаточно произволен, и может быть достаточно высоким уровнем для другой подсистемы;
- - подсистемы могут быть реализованы в различном порядке, в различных комбинациях и иметь тенденцию к развитию во времени;
- - развитие всей системы основано на развитии ее подсистем.
- К системам, удовлетворяющим вышеперечисленным признакам, структурное программирование (сверху вниз) не применимо. Такая система никогда не будет полностью работоспособна.
- Если система обладает более простой конструкцией, к ней может быть применен ГОСТ 28195-99.

## 2. Принципы объектно-ориентированного подхода (дополнение к показателям С2 и С3 в ГОСТ 28195-99)

Для анализа работоспособности сложной системы, проектирования, программирования и тестирования должен применяться объектно-ориентированный подход, состоящий из следующих основ:

- объектно-ориентированного анализа;
- объектно-ориентированного проектирования;
- объектно-ориентированного программирования.

# 3. Венгерская нотация
## (дополнение к показателю С3 в ГОСТ 28195-99)

Требования к оформлению программ должны соответствовать **венгерской нотации (Hungarian Notation),** формально обозначаемой как **добавление префикса к имени идентификатора.** Это требование предъявляется всеми ведущими мировыми фирмами.

Требования следующие:

- мнемоническое значение: идентификатор должен легко запоминаться;
- смысловое значение: роль идентификатора должна быть ясна из его названия;
- преемственность: похожие объекты должны иметь похожие идентификаторы;
- скорость решения: придумывание, ввод и редактирование идентификатора не должны занимать слишком много времени, идентификатор не должен быть слишком длинным.
- Каждая строка в программе должна иметь смысловой комментарий

# 4. Обработка исключений
## (дополнение к показателю К4 в ГОСТ 28195-99)

В программах, работающих в среде Windows (либо в собственных операционных системах, написанных на базе микропроцессора не ниже Intel386) должна присутствовать обработка исключений.

Обработка исключений предполагает наличие в программе специальных конструкций. Для языка C++ это следующие операторы:

- **try** – отмечает тот участок текста программы, в котором потенциально возможно возникновение ошибки
- **catch** – этот блок следует непосредственно за блоком try и содержит операторы обработки обнаруженной ошибки
- **throw** – этот оператор используется для передачи сообщения об ошибке в вызывающую часть программы (выбрасывает исключение); блок try далее не выполняется.

# 5. Защита от НСД
## (дополнение к показателю К5 в ГОСТ 28195-99)

- Для защиты от несанкционированного доступа в программном продукте должны быть реализованы следующие функции:
- - защита от дизассемблирования и трассировки;
- - генерация серийного номера и настройка на параметры компьютера, на котором выполняется генерация и дальнейшая эксплуатация;
- - каждая подсистема должна отлаживаться независимо от других;
- - сложная система должна разделяться и по алгоритмам и по объектам (это должно быть отражено в руководстве программиста на конкретный программный продукт);
- - в сложной системе должна присутствовать и отображаться по запросу пользователя статистика работы всех процессов системы. В частности, должны в протоколе отображаться сбои системы (счетчики сбоев за каждые 15 минут).
- - защищенный режим микропроцессора (защита от взлома должна выполняться на уровне команд микропроцессора, начиная с команд микропроцессора Intel386).
- - должна выполняться периодическая проверка производительности отдельных подсистем компьютера.

Спасибо за внимание.