

Общие требования по стилю программирования на первый семестр (смесь)

- Функции, разделенные по смыслу, должны храниться в разных файлах и подсоединяться к программе через файл проекта.
- Прототипы функций должны быть собраны в один или несколько файлов с расширением .h и подключаться через #include
- Все константы в программе должны быть переменными с модификатором const.
- Внешними переменными (теми, которые используются внутри функций как внешние) могут быть только константы. Вся остальная информация передается через параметры.
- Динамическое распределение памяти используется только там, где оно необходимо. Обязательным является освобождение памяти.
- Отведение и освобождение памяти, открытие и закрытие файлов должно происходить на одном и том же структурном уровне.
- Необходимы проверки на успешность открытия файла, отведения памяти и пр.
- Возможны два способа обработки обнаружения ошибки внутри функции: если ошибка критическая, то выход из программы; но лучше, если функция возвращает к.-л. специальное значение, которое обрабатывается в вызывающей функции. (Пока речь не идет о выбрасывании исключений – это будет позже.)
- Удобным (консольным) режимом задания имен файлов является следующий: имена входных файлов в программе берутся из командной строки; имена выходных файлов – или из командной строки или вводятся с клавиатуры.
- При вводе параметров должен быть контроль над допустимостью вводимых значений.
- При разработке функций нужно реализовывать несколько больше возможностей, чем требуется в данной задаче (например, если нужно нарисовать что-то в определенном месте на экране, то лучше написать функцию, которая может рисовать в любом месте). Более того, нужно предполагать, что делается общественно-полезная вещь, которую будут использовать другие люди (программисты).
- Нельзя в функциях перемешивать полезные действия с объектами и взаимодействие с внешними устройствами (например, вывод на печать), так как это системно-зависимые действия.
- При объединении некоторых действий в одну функцию нужно учитывать их объективную взаимосвязь (и не учитывать субъективную, т.е. то, что в данной программе эти действия идут друг за другом).
- Минимальные комментарии к каждой функции состоят из описания действия функции, ее параметров и пр. (написанный текст затем можно скопировать в описание программы). В описание функции (или хотя бы в файл) рекомендуется вставлять дату последнего исправления.
- Используйте для идентификаторов осмысленные обозначения, основанные на словах английского языка.
- В более сложных, чем в первом семестре, задачах при реализации действия, которое может занять продолжительное время, на экране должен каким-либо образом отображаться процесс работы (например, процент выполненной на данный момент работы), чтобы можно было оценить скорость выполнения и узнать, когда программа "виснет".
- При организации вычислений нужно обрабатывать возможные переполнения при арифметических операциях.