**Вопросы и ответы по поводу написания курсовой работы**

**1**. **Был следующий вопрос.**

*У меня возник вопрос по поводу курсовой: немного не понятно как ее писать. Например, должна ли вся таблица храниться в одном файле, если да, то как должен осуществляться ввод (по строкам целиком или по ячейкам, как было в 5)? Если ввод по строкам, то нужно ли использовать разделители? Использование структуры запрещено в принципе или только с массивом? Не могли бы вы это уточнить?*

Ответ.

Просто одну структуру использовать можно. Вводить надо, конечно, по отдельным полям. Так и пользователю удобно, и программисту легче защиты сделать. Нельзя хранить в оперативной памяти весь список, то есть массив структур. Весь список должен целиком храниться в файле.

   Например, организуете ввод одной строки по отдельным полям (по ячейкам). Ввели ее — и записали в файл. Потом, используя тот же код и те же переменные, ввели вторую строку, и записали ее в файл. И т. д. Процесс аналогичен созданию файла с числами, который я Вам высылала в личный кабинет. В том примере массива чисел в оперативной памяти нет, все числа хранятся в файле. Вот и у Вас в курсовой работе не должно быть массива структур.

**2. Вопрос.**

*Вы написали, что нужна защита, когда мы даём имя файлу. Можно ли создавать файлы как в 6 и 7 лр и самим давать им имя?*

*Или нужно у пользователя просить ввести имя и делать проверку?*

Ответ.

 В курсовой работе непосредственно в коде указать имя файла с базой — можно, работу я приму. Но на практике это, конечно, не самое лучшее решение. Ниже — информация на эту тему.

   Лучше всего, когда программа запрашивает у пользователя вариант работы: пользователь будет работать со стандартной базой данных (со стандартным именем) или со своей? Если пользователь планирует работать со своей базой, он должен ввести имя, а программа затем должна провести соответствующие проверки. Если пользователь планирует работать со стандартной базой данных, то программа обычно читает имя из специального файла конфигурации. Конечно, в этом случае тоже идут проверки, например, существует ли этот файл конфигурации. Просто указать имя файла в коде программы в виде строковой константы, конечно, можно. Этот способ — самый простой для реализации, но самый неудачный с точки зрения организации работы приложения.

**3.  Замечания по организации взаимодействия программы с пользователем.**

    В любой ситуации ввода пользователя должна быть предусмотрена возможность либо отказаться от ввода, либо при любой введенной информации программа продолжает работать, а иначе мы «зависаем» в определенных ситуациях. Например, были следующие случаи работы.

    Режим «Удаление строк». Надо предусматривать выход из режима, если пользователь раздумал удалять строку или строк нет. Например, была 1 строка, пользователь ее удалил и случайно нажала «продолжение удаления». Программа зацикливается, так как просит ввести номер , а его уже нет

   Или др. ситуация. Начало работы. Пользователь говорит, что будет работать со своим имеющимся файлом. Программа просит ввести имя, а его нет. И что делать пользователю? Ему из этой ситуации не выйти, только крестик нажать.

**4. Я пишу с вопросами насчет курсовой от потока**

*Мы должны сдать вам комплект только с курсовой или еще нужно сдавать все отчеты по лр?*

Ответ.

 Нет, сдавать отчеты по лабам не надо. Что должно быть в папке «Курсовая» = см. 2 раздел в метод. указаниях. Должно быть: 1) файл с отчетом, 2) файл с экзешником и всем, что к нему причитается для нормальной работы 3) отдельный файл с шаблоном отзыва 4) отдельный файл с заявлением. Всю папку назовите типа «Курс\_Иванов\_ПИ91».

**5. Я пишу с вопросами насчет курсовой от потока**

*Вопрос: что писать в отмеченных местах?*

*3.2.5.2 Раздел «Проектирование»*

*Алгоритм решения на данном этапе описывается в общем виде, без излишней детализации. Объясняется общий подход к реализации задачи и логика работы программы (обычно в виде укрупненных структурных схем или словесно, возможно словесное описание алгоритма в совокупности со структурными схемами).*

**Ответ.** В 2 разделе описываете подход к решению. Здесь еще нет имен функций, переменных – это этап проектирования решения задачи.

Можно написать примерно так.

Исходя из поставленной задачи, реализуемая программа должна работать в диалоговом режиме. Общая логика работы соответствует организации и построению диалогового меню. Предлагается следующий вид основного меню:

. . . . .

При выборе элемента меню «….» = следующий уровень меню имеет вид:

…. . . . .

Можно писать все словами, можно изобразить это схематично Схема будет напоминать укрупненную структурную схему. Много писать не надо: 1-2 страницы на весь 2-ой раздел.

*3.2.5.3 Раздел «Реализация»*

*3.2.5.3.2 Подраздел «Структура программы»*

*Приводится структура программы в виде схемы иерархии модулей. Эта схема показывает, какие модули вызывают друг друга. Схема иерархии состоит из прямоугольников и соединительных линий. Отдельный прямоугольник – это отдельный модуль, в прямоугольнике записывается название модуля. Линии показывают подчиненность модулей (а не порядок выполнения, как в структурной схеме). Каждый модуль активизируется вышестоящим модулем и, закончив работу, возвращает управление вызвавшему модулю.*

*В отчете желательно использовать средства UML (диаграммы компонентов и диаграммы последовательности, диаграммы вариантов использования и др.).*

**Ответ.** В 3 разделе описываете, как Вы уже написали программу. Обычно этот раздел студентам писать легко, они пишут, как и что у них сделано.

Вопрос о модульной структуре программы. Модуль – это понятие, которое в программировании трактуется очень широко в зависимости от контекста. В Паскале есть такая языковая конструкция. Там это что-то вроде библиотеки функций определенного назначения, и этот модуль ( библиотеку) можно подключить примерно как в Си мы подключаем библиотеки по include. Вообще, модуль – это каким-то образом оформленная часть программы, реализующая цельную логическую функцию. В простейшем случае, а это как раз наш случай, - модуль – это отдельная функция. Поэтому модульную структуру изображают примерно так (некогда рисовать, обведите имя каждой функции в прямоугольник и соедините стрелочками «кто кого вызывает»).

main

spisk vivod zapros

vvod korret udalen

Далее, в следующем подразделе, когда описывается состав программы, для каждой из этих функций указывают прототип и назначение. Описывать мелкие служебные функции типа «определение четности числа» не надо. Здесь описываются значимые функции.

Иногда при изображении модульной структуры дают названия не функций, а смысловые названия логических модулей по-русски. Например, «работа со списком», «Вывод». «Запрос».

В любом случае, модуль – это не строго определенное понятие в программировании, здесь есть простор для выражения своей мысли, когда надо некую программу разбить на функциональные составляющие.

Что касается UML – ведь написано «желательно», значит – «необязательно». Может, кто-то в щколе изучал, может, на информатике давали, может, кто-то очень любопытный, и сам все посмотрел. Вообще, UML – это «универсальный язык моделирования». Это среда, которая позволяет графически (схема, диаграмма и т.д.) изображать самые различные действия. Очень широко используется в программировании для презентации алгоритмов, классов, и др. Вас обяжут с этим разобраться самостоятельно. Сейчас – по желанию и если время есть.

**6. Что писать в «Обзор предметной области»?**

**Ответ.** Обзор предметной области.

Ваша задача к какой области знаний относится? Работа со структурированными данными. Вот и пишите об этом. Как это важно, какие типы данных используются (массивы, структуры, классы, файлы), какие специальные технологии (электронные таблицы, базы данных). Для работы с базами данных существуют специализированные языки – какие. Но можно реализовать данную задачу и на просто языке высокого уровня, чему и посвящена эта работа.

Вот примерно в таком духе.