**Задача 10. Массивы / списки объектов (структур или классов)**

В задаче обязательно использовать свои (описанные в вашей программе) составные типы данных (структуры или классы).

Для представления набора объектов лучше использовать списки (массивы также допустимы, но менее удобны для данной задачи).

Данная задача должна быть оформлена двумя способами:

1. в виде консольного приложения;

2) в виде оконного приложения, где двумерный массив можно задать в GridView. При этом должна быть возможность загрузить данный из файла в GridView (реализованная в виде двух функций: чтение данных из файла в двумерный массив и отображение двумерного массива в GridView), а также сохранить данные из GridView в файл (реализованная в виде двух функций: чтение данных из GridView в двумерный массив и запись двумерного массива в файл).

Функции, реализующие логику задачи и чтение / запись данных из файлов / в файлы, должны быть оформлены в виде отдельного модуля (в отдельном файле). Этот модуль без каких-либо изменений должен использоваться в двух программах: с консольным интерфесом (файлы для чтения / записи задаются в параметрах командной строки) и оконным интерфейсом.

Данные в файлах должны храниться в следующем виде: каждая отдельная строка описывает отдельный объект. Соответственно строки состоят из нескольких элементов (чисел, строк), каждый из которых описывает отдельный атрибут объекта. Например, набор линий вида ax + by + c = 0 будет храниться в текстовом файле в следующем виде:

4 5 4 10 2 12

5 34 23

и т.д. ,

где каждая строка описывает отдельную линию с параметрами a, b, c.

Заранее придумать не менее 5 различных тестов, охватывающих как типичные, так и все возможные граничные (наиболее невероятные и показательные) ситуации. (Сохранить в текстовых файлах input01.txt, input02.txt и т. д.)

Решение, естественно, должно быть оформлено в виде отдельной функции. В реализации обязательно использовать вспомогательные функции примерно так же, как в предыдущих Задачах № 8 - 9 (в этой задаче, какие именно вспомогательные функции использовать, не расписано, вы должны самостоятельно решить, какие именно будут удобны для вашего варианта

Варианты:

10.17. Прайс-лист содержит следующую информацию о планшетах: название модели, объем памяти, рейтинг модели среди пользователей (цифра от 1 до 5), стоимость. У вас есть N денег, вам необходимо купить самый хороший планшет, на который вам хватит денег. Для вас приоритетным является объем памяти, в случае одинакового объема – рейтинг модели. Если подходящих моделей будет несколько, вы выберете Samsung или Asus (присутствует в названии модели), а затем все остальные. Если по этим критериям вам подходят несколько моделей – выбирайте любой.

**Задача 13. Логические игры**

Задача обязательна для тех студентов, кто претендует на оценку «Отлично».

В задаче обязательно реализовать логику игры в виде отдельного класса Game – см. Лекцию 10, а также пример SapperGame к лекции 10.

Оформление игры может быть максимально простым (допустим, не надо плавных перемещений шариков с клетки на клетку, каких-либо визуальных эффектов, звукового сопровождения и т.п.). Элементы – просто цветные кружки / кубики / обычные цифры и т.д. в зависимости от варианта. Выделение элемента (если нужно) – изменение цвета фона ячеек игрового поля. (В любом случае все это не относится к логике игры.)

Никаких уровней и т.д.

Для отрисовки игрового поля проще использовать DataGridView, как в примере SapperGame (для каких-то игр понадобятся несколько DataGridView), но можно полностью нарисовать поле на форме самостоятельно (в обработчике события формы Paint, при этом, возможно, понадобятся дополнительные методы).

Во многих играх, приведенных в вариантах задач в виде примера, кроме центральной идеи, есть различные, так скажем, расширения-исключения. Вот этого всего реализовывать также не надо.

Кроме непосредственно реализации игры необходимо подготовить краткое описание правил игры с иллюстрациями (эти описания будут впоследствии добавлены в этот документ).

Если что-то непонятно с игрой подойдите ко мне (Соломатин Д.И.) – я поясню (например, после лекции).

Варианты:

13.17. Цепная реакция – http://igroflot.ru/flash\_games\_41.htm

Предусмотреть несколько уровней (уровни выбираются в меню). Описание уровней должно храниться в виде текстовых файлов level\_01.txt, level\_02.txt в директории levels, которая расположена в директории, в которой находится сам исполняемый файл игры.