**Учебный центр IT Class**

**Задание**

**ООП**

Предприятие оказывает клиентам услуги по снабжению энергией. Расчёты за потреблённую энергию осуществляются ежемесячно. Клиенты предприятия подразделяются на **четыре** вида, каждый из которых имеет свой алгоритм вычисления оплаты за потреблённую энергию:

1. Обычные – стоимость потреблённой энергии прямо пропорциональна потреблённой энергии. Одна единица потреблённой энергии стоит 15 коп.
2. С лимитом – первые **250** единиц энергии клиент получает по цене, как для обычных клиентов (15 коп.), а всё, что потреблено сверх **250** единиц, оплачивается по тарифу **на ⅓** больше (20 коп.)
3. Льготные – стоимость потреблённой энергии прямо пропорциональна потреблённой энергии, но применяется льготный тариф, который составляет **⅔** от тарифа для обычных клиентов (10 коп.)
4. Для нужд отопления. Стоимость потреблённой энергии прямо пропорциональна потреблённой энергии, но используется дешёвый тариф, составляющий **1⁄15** от тарифа для обычных клиентов (1 коп.)

Разработать **консольную** программу, в которой создать массив из данных о потреблённой клиентами энергией **за** **один** месяц. В массиве должно быть данные о минимум трёх клиентах каждого вида. Объём потреблённой каждым клиентом энергии должен быть введён в программу с использованием константных значений (*программа не должна запрашивать ввод данных с консоли*).

Выполнить следующую обработку хранящихся в массиве данных и вывести результаты выполнения каждого шага на консоль:

* Отсортировать массив по количеству потреблённой клиентами энергии по убыванию.
* Отсортировать массив по величине оплаты клиентами по возрастанию.
* Упорядочить массив по типу клиентов (первыми – обычные, затем – с лимитом, затем – льготные и последними – для нужд отопления)
* Вычислить общую сумму SUM оплаты всех клиентов за потреблённую энергию.
* Вычислить общий размер льготы LG=SUM0–SUM. Где SUM0 – общая сумма оплаты всех клиентов за потребленную энергию, если бы они все были обычными (без льгот и лимитов).

Требования:

Использовать объектно-ориентированный подход для описания сущностей предметной области;

Не использовать графический интерфейс;

Предусмотреть возможности модернизации приложения, изменения тарифов и коэффициентов.