# когда узнал что: китайские дети делают на уроках труда айфоны, Мем твое выражение лица - Рисовач .Ру

# **Лабораторная работа № 10**

В связи с недавним выходом iPhone 12, власти Китая решили сделать «подарок» всему миру   
и наладили производство данной модели в школах силами учащихся, прямо на уроках труда.   
Для учета выпущенных телефонов, а также премирования школьников, которые собрали телефонов больше всех, вас просят написать информационную систему под названием «ЯДелатьТвойАйВон».

**Описание возможностей информационной системы «ЯДелатьТвойАйВон».**

Тестовую эксплуатацию вашей информационной системы планируется провести в небольшой сельской школе, где обучается 50 млн. учеников. На этапе проектирования вы приняли решение для описания сущности «Ученик» использовать структуру (struct), а не класс. Этот выбор был сделан вами, так как состав проектируемой сущности – поля исключительно значимого типа: порядковый номерe ученика, количество сделанных айфонов за урок, количество заработанных дополнительный обедов. То есть, в ходе анализа разрабатываемой сущности, вы увидели, что она занимает много места в памяти, и выбранный тип данных позволит вам оптимизировать процесс создания такого большого количества легковесных сущностей в программе.

**Технические требования к информационной системе «ЯДелатьТвойАйВон».**

* структура Student содержит поля исключительно значимого типа: Number; CountPhone; CountLunch и поле, которое характеризует ученика с позиции его возраста – enum Position. Позиция определяет, является ли ученик старшеклассником/младшеклассником/дошколенком.
* класс School содержит коллекцию учеников, а также методы:
  + FindMinMaxEmployee() – метод по нахождению самого производительного и самого медлительного школьника по итогом урока. Тип возвращаемого значения метода – картеж из следующего набора данных: Student, Student
  + Reward(ссылка на картеж) – метод принимает кортеж данных из FindMinMaxEmployee() и вычитает обед у ленивца, прибавляет обед торопыге.

Поиск самого производительного школьника и аутсайдера реализовать средствами сортировки коллекции с учениками (метод Sort() класса List). Для обеспечения работы метода сортировки сортируемые сущности должны имплементировать интерфейс IComparable (метод CompareTo()). Сортировку настроить по полю CountPhone, в случае одинакового количества собранных телефонов, более быстрым считать того ученика, который младше по полю Position.

Для демонстрации небольшого превосходства структур перед классами в плане скорости создания, произведите замер времени работы программ. Сначала запустите программу со структурами, а потом с классами (сделайте по 2-3 запуска).

