* **Базовая сложность**
  1. Создать класс Employee, который содержит информацию о Ф. И. О., отделе и зарплате сотрудника. Отделы для простоты должны быть названы от 1 до 5.
  2. Добавить статическую переменную-счетчик, которая будет отвечать за id.
  3. Добавить в класс Employee поле id, которое проставляется из счетчика, а затем счетчик увеличивает свое значение.
  4. Добавить возможность получать значения полей из Employee (геттеры) для всех полей.
  5. Добавить возможность устанавливать значения полей отдела и зарплаты (сеттеры).
  6. По умолчанию все поля должны передавать через конструктор (кроме id) и заполняться в нем (включая id, который нужно получить из счетчика).
  7. Создать внутри класса с методом main поле типа Employee[10], которое будет выполнять роль «хранилища» для записей о сотрудниках.
  8. Создать статические методы, которые будут взаимодействовать с массивом и предоставлять результат:
     1. Получить список всех сотрудников со всеми имеющимися по ним данными (вывести в консоль значения всех полей (toString)).
     2. Посчитать сумму затрат на зарплаты в месяц.
     3. Найти сотрудника с минимальной зарплатой.
     4. Найти сотрудника с максимальной зарплатой.
     5. Подсчитать среднее значение зарплат (можно использовать для этого метод из пункта b).
     6. Получить Ф. И. О. всех сотрудников (вывести в консоль).
     7. **Критерии оценки**

– Корректно создан класс Employee.

– Реализованы геттеры для всех полей класса.

– Реализованы сеттеры для всех полей класса.

– Создано поле типа Employee[10] для хранения записей о сотрудниках.

– Созданы методы, которые корректно выводят информацию:

* + - * список всех сотрудников со всеми данными,
      * поиск сотрудника с минимальной зарплатой,
      * поиск сотрудника с максимальной зарплатой,
      * подсчет среднего значения зарплат,
      * список Ф. И. О. всех сотрудников.

– Программа работает корректно при изменении любых данных о сотрудниках и выводит верный результат.

* **Повышенная сложность**

Создать дополнительные статические методы для решения следующих задач.

* 1. Проиндексировать зарплату (вызвать изменение зарплат у всех сотрудников на величину аргумента в %).
  2. Получить в качестве параметра номер отдела (1–5) и найти (всего 6 методов):
     1. Сотрудника с минимальной зарплатой.
     2. Сотрудника с максимальной зарплатой.
     3. Сумму затрат на зарплату по отделу.
     4. Среднюю зарплату по отделу (учесть, что количество людей в отделе отличается от employees.length).
     5. Проиндексировать зарплату всех сотрудников отдела на процент, который приходит в качестве параметра.
     6. Напечатать всех сотрудников отдела (все данные, кроме отдела).
  3. Получить в качестве параметра число и найти:
     1. Всех сотрудников с зарплатой меньше числа (вывести id, Ф. И. О. и зарплатой в консоль).
     2. Всех сотрудников с зарплатой больше (или равно) числа (вывести id, Ф. И. О. и зарплатой в консоль).
  4. Критерии оценки

– С помощью **шести** методов можно вывести данные:

* + 1. сотрудник с минимальной зарплатой,
    2. сотрудник с максимальной зарплатой,
    3. сумма затрат на зарплату по отделу,
    4. средняя зарплата по отделу,
    5. индексация зарплаты на %,
    6. печать всех данных всех сотрудников одного отдела, кроме номера отдела.

– Реализован параметр поиска всех сотрудников с зарплатой больше определенного числа, которое можно указать.

– Реализован параметр поиска всех сотрудников с зарплатой меньше определенного числа, которое можно указать.

– Программа работает корректно при изменении любых данных о сотрудниках и выводит верный результат.

* **Очень сложно**

Привести структуру проекта к ООП.

* 1. Создать класс EmployeeBook.
  2. Перенести хранилище сотрудников в него (массив), закрыть к нему доступ извне (сделать приватным).
  3. Все статические методы по работе с массивом перенести в этот класс и сделать нестатическими.
  4. Добавить несколько новых методов:
     1. Добавить нового сотрудника (создаем объект, заполняем поля, кладем в массив). Нужно найти свободную ячейку в массиве и добавить нового сотрудника в нее. Искать нужно всегда с начала, так как возможно добавление в ячейку удаленных ранее сотрудников.
     2. Удалить сотрудника (находим сотрудника по Ф. И. О. и/или id, обнуляем его ячейку в массиве).
  5. Изменить сотрудника (получить сотрудника по Ф. И. О., модернизировать его запись):
     1. Изменить зарплату.
     2. Изменить отдел. Придумать архитектуру. Сделать или два метода, или один, но продумать его.
  6. Получить Ф. И. О. всех сотрудников по отделам (напечатать список отделов и их сотрудников).
  7. Критерии оценки

– Создан приватный класс EmployeeBook.

– Данные о сотрудниках перенесены в приватный класс EmployeeBook.

– Реализованы метод добавления нового сотрудника.

– Реализован метод удаления сотрудника.

– Реализован метод изменения зарплаты сотрудника.

– Реализован метод изменения отдела.