****Закрепите полученные знания, выполнив практическую работу. Оба задания обязательны для выполнения и сдачи на проверку.****

Выполнить работу будет гораздо проще после самостоятельной практики. Задания для неё мы разместили под некоторыми видео этого модуля.

#### Задание 1

****Цель****

* Опробовать принципы работы статических переменных и методов.
* Научиться самостоятельно создавать статические переменные и реализовывать в классах статические методы.

****Что нужно сделать****

Для выполнения этого задания используйте проект из практической работы 4.10, в GitLab — директория ObjectsAndClasses\_Methods.

Внесите следующие изменения в класс Basket:

1. Добавьте две статические переменные для хранения общей стоимости и общего количества всех товаров во всех корзинах.
2. Реализуйте статические методы, которые будут увеличивать значения этих переменных при добавлении в корзину новых товаров.
3. Реализуйте статический метод расчёта средней цены товара во всех корзинах. Он должен рассчитывать и возвращать отношение общей стоимости всех корзин к общему количеству всех товаров.
4. Реализуйте статический метод расчёта средней стоимости корзины (отношение общей стоимости всех корзин к количеству корзин).

Названия и типы переменных, а также названия и методов придумайте самостоятельно.

****Критерии оценки работы****

Принято:

* Выполнены все указанные в задании требования: созданы обе переменные и реализованы все методы.
* Имена переменных соответствуют хранимым в них данным.
* Имена методов соответствуют выполняемым действиям.
* Все методы работают без ошибок, код компилируется.

На доработку: задание не выполнено, выполнено неточно либо частично.

****Как отправить работу на проверку****

Выполните также второе задание (смотрите ниже) и отправьте куратору результаты сразу по обоим заданиям.

#### Задание 2

****Цель****

Закрепить на практике принципы реализации и использования статических методов и переменных, а также перечисляемых типов enum.

****Что нужно сделать****

1. Задание выполняйте в директории “ObjectsAndClasses\_StaticFinal” репозитория “java\_basics”.
2. Создайте новый проект и класс Computer со свойствами-комплектующими. Класс Computer должен иметь методы, позволяющие менять его комплектующие с помощью сеттеров: процессор, оперативную память, накопитель, экран и клавиатуру.
3. Для каждого такого свойства создайте классы с перечисленными полями (имена классов и полей, а также типы полей придумайте самостоятельно) для хранения следующих сущностей:
   * Процессор:
     + частота;
     + количество ядер;
     + производитель;
     + вес.
   * Оперативная память:
     + тип;
     + объём;
     + вес.
   * Накопитель информации:
     + тип — HDD, SSD;
     + объём памяти;
     + вес.
   * Экран:
     + диагональ;
     + тип (IPS, TN, VA);
     + вес.
   * Клавиатура:
     + тип;
     + наличие подсветки;
     + вес.
4. Все эти классы должны быть иммутабельными (все поля должны быть final). Имена классов и полей, а также типы полей придумайте самостоятельно. ****Сам класс Computer иммутабельным быть не должен!****
5. В классе Computer также создайте текстовые поля vendor (производитель) и name (название), задаваемые при создании компьютера (в конструкторе). Они также должны быть final.
6. Реализуйте в классе Computer следующие методы:
   * Метод расчёта общей массы компьютера, возвращающий суммарный вес всех его комплектующих.
   * Геттеры и сеттеры для всех комплектующих компьютера.
   * Метод toString, возвращающий в удобочитаемом формате всю информацию о компьютере и его комплектующих.
7. Напишите в методе main код, который будет создавать несколько компьютеров и выводить в консоль всю информацию о них.

****Критерии оценки работы****

Принято:

* Выполнены все указанные в задании требования.
* Имена переменных соответствуют хранимым в них данным.
* Имена созданных и реализованных методов соответствуют выполняемым ими действиям, а имена геттеров и сеттеров — правилам именования геттеров и сеттеров.
* После запуска кода выводится информация обо всех созданных компьютерах, соответствующая информации, заданной у них при создании.

На доработку: задание не выполнено полностью или выполнено частично.