Задача 1. Ветвистый if

1. Заведите две переменные типа int и занесите туда любые значения (особо продвинутые могут считать аргументы запуска типа java Task1 **42 34 .**Конвертировать строку в число можно с помощью функции Integer.valueOf )

2. Выведите на экран следующее:

- Если любое из чисел меньше нуля - надпись "Есть  отрицательное число". Само число можно не выводить

- Если второе число не ноль - надпись "Результат деления:" + результат деления первого, на второе. Дробная часть не должна быть отброшена.

- Написать в консоли «Первое число больше второго» или «Второе число больше первого»…

- Если первое число четное - надпись "число $ - четное".

Задача 2.  Массивные массивы

1. Заведите массив на 8 элементов типа int. Можете заполнить его самостоятельно или считать из аргументов запуска (в этом случае, массив может быть любой длины)

2. Посчитайте сумму всех элементов. Выведите её на экран.

3. Если хотя бы одно число в массиве больше 1000 - выведите на экран надпись "Массив - массивный!". Эта надпись должна быть выведена не больше одного раза, даже если чисел больше 1000 несколько.

Задача 3. Есть массив чисел (см. предыдущее задание)

1. Найдите в нем самое больше число.

2. Если в массиве встретилось число 100500 - игнорировать его при определении самого большого. Т.е.

в массиве [1, 2, 3, 100500, 42] - самое большое - 42.

Задача 4.

**Все объекты должны находиться в package ru.mera.<ваша фамилия>.postman;**

1.      Создать класс Person с полями name, surname

2.      Создать класс Message с полями from, to, message\_body.

3.      Создать два подкласса Message: Email, Letter.

4.      В Email должны быть поля from, to, message\_body, receive date.

5.      В Letter должны быть поля from (необязательное), to, message\_body,   fromPostalIndex, toPostalIndex

6.      Все классы выше должны быть immutable, т.е. не допускать изменения состояния после создания.

7.      Создайте класс MailBox в котором есть два массива для хранения сообщений – обычных и электронных.

8.      В классе MailBox есть 4 метода: добавить сообщение обычное, электронное, получить все обычные сообщения, получить все электронные сообщения.

9.      Массивы должны «вырастать», если емкость существующего массива заканчивается. Можете использовать System.arrayCopy для этого  
<https://www.tutorialspoint.com/java/lang/system_arraycopy.htm>

Задача 5. Приводимая неприводимость.

1. Создать класс Person с именем и фамилией. Переопределите метод toString, чтобы выводились имя и фамилия.

2. Создать класс Artist (наследник Person) с дополнительным полем style - тип "рисования"

3. Создать класс Musician с дополнительным полем Instrument - инструмент на котором играет музыкант.  
**Метод toString в классах Artist и musician не переопределяйте!**

4. Создать массив типа Person.

5. Положить туда несколько Музыкантов и Художников. (можно используя Random - класть случайных людей, выбирая имена, фамилии, стили и инструменты из заранее определенного списка)

6. Используя цикл for выведите на экран список всех людей в формате:  
"<имя> <фамилия> играет на <instrument>"  
или   
"<имя> <фамилия> рисует в стиле  <style>"

а зависимости от типа объекта (используйте instanceof, чтобы проверить)

Задача 6. Абстрактная конкретность.

Сегодня будем делать машину для приготовления коктейлей.

Рецепт - очень простой:  
public void mix() {

        взятьЧтоТоЖидкое();

        взятьОсновнойИнгредиент();

        добавитьТопинг()

        print("Смешиваю %жидкое% с %наполнитель% и добавляю сверху %топинг%")

};

Вам нужно:  
1. Сделать базовый абстрактный класс CocktailMaker с методом mix() и тремя абстрактными методами для получения жидкого ингредиента, основного и топинга.

2.Создать несколько конкретных классов, которые всегда будут возвращать одни и те же ингредиенты.

Например, BananaMilkCocktailMaker:  
public String liquid {return "Молоко"}

public String mainIngredient {return "Банан"}

public String toping{return "Мороженое"}

3. Создать массив для базового класса и положить туда несколько разных машин для приготовления коктейлей.

4. Запустить все машины по одному разу использую цикл for.