Рефлексия / Reflection API

---

1)
Создайте в классе Main, статический типизированный метод для вывода любого количества объектов одного класса в консоль в виде таблицы.
У таблицы должны быть заголовки столбцов – имена нестатических полей класса, полученные с помощью рефлексии.
Строка таблицы – значения всех нестатических полей класса, полученные с помощью рефлексии.
Сигнатура метода:

```
public static <M> void table(M... os) {}
```

Пример вызова метода:

```
Main.<User>table(u1, u2, u3, u4);
```

Явная типизация дженерик типа метода при вызове, контролирует то, что метод будет принимать объекты одинакового типа, указанного в дженерик типе.

Для проверки метода, создайте класс User с полями:

```
private int id;
private String name;
private int age;
private boolean activated;
private String login;
private String pass;
```

Добавьте статическое поле:

```
private static int idCounter;
```

Добавьте конструктор со всеми полями кроме id.
Значение id должно автоматически инициализироваться из статического поля idCounter.

Создайте класс Student с полями:

```
private String surname;
private String name;
private int age;
private String group;
```

Проверьте метод table() с объектами класса Student.

---

2)

Создайте класс ArrRef для рефлексионной работы с массивами.
Добавьте в этот класс метод для вывода любого массива в консоль с помощью рефлексии. Пример сигнатуры метода:

```
public static void arrOut(Object arr) {}
```

Этот метод должен проверять, чтоб принятый объект был массивом.
Для этого название типа данных должно содержать "**[**"
Если это массив, его элементы нужно вывести в консоль в линию через табуляцию. Для вывода элементов массивов в консоль используйте класс Array.

Добавьте метод для заполнения массива случайными значениями с помощью рефлексии. Пример сигнатуры метода:

```
public static void arrFill(Object arr) throws Exception {}
```

Этот метод должен проверять, чтоб принятый объект был массивом.
Если это массив, нужно перебрать его с помощью класса Array, и заполнить случайными значениями того же типа данных, как и у элементов массива.

Для заполнения массива случайными значениями нужно добавить в класс ArrRef, метод, который с помощью рефлексии будет получать, и возвращать метод (объект типа Method), из класса Random для генерации случайного значения соответствующего типа.
Пример сигнатуры:

```
public static Method getMethodFromRandom(Object arr) throws
Exception {}
```

Для этого нужно получить имя типа данных элемента массива и имя метода для генерации случайного значения такого-же типа из класса Random.
Например, можно обратится к 0 элементу массива, если длинна массива > 0, получить имя класса элемента массива.
Например, название типа данных:

```
java.lang.Integer
```

Строку нужно разбить и получить слово **integer** в нижнем регистре.
Далее перебрать методы класса Random, найти метод, который содержит в названии слово **next** и название типа данных, который генерируется этим методом.
Например, из названия метода nextInt(), нужно получить слово **int** в нижнем регистре.
Далее, проверить содержится ли в слове **integer** слово **int**, если да, нужно вернуть из метода getMethodFromRandom() соответствующий метод, и использовать его при заполнения массива случайными значениями. Если подходящий метод не был найден, нужно вернуть null.
