

Задачи

1. Создать аннотацию `@IsLike`, применимую к классу во время выполнения программы. Аннотация может хранить `boolean` значение.
2. Написать метод, который рефлексивно проверит наличие аннотации `@IsLike` на любом переданном классе и выведет значение, хранящееся в аннотации, на экран.

3. Есть класс `APrinter`:

```
public class APrinter {  
    public void print(int a) {  
        System.out.println(a);  
    }  
}
```

Задача: с помощью рефлексии вызвать метод `print()` и обработать все возможные ошибки (в качестве аргумента передавать любое подходящее число). При “ловле” исключений выводить на экран краткое описание ошибки.

4. Написать метод, который с помощью рефлексии получит все интерфейсы класса, включая интерфейсы от классов-родителей и интерфейсов-родителей.

Дополнительные задачи (с собеседований)

1. Дана строка, состоящая из символов “(” и “)”
Необходимо написать метод, принимающий эту строку и выводящий результат, является ли она правильной скобочной последовательностью или нет.

Строка является правильной скобочной последовательностью, если:

- Пустая строка является правильной скобочной последовательностью.
- Пусть S — правильная скобочная последовательность, тогда (S) есть правильная скобочная последовательность.
- Пусть S_1, S_2 — правильные скобочные последовательности, тогда S_1S_2 есть правильная скобочная последовательность.

Пример:

| Входные данные | Выходные данные |
|----------------|-----------------|
|----------------|-----------------|

| | |
|----------|-------|
| ((()())) | true |
|)(| false |
| ((| false |
| ((())) | true |

2. Дана строка, состоящая из символов “(”, “)”, “{”, “}”, “[”, “]”
Необходимо написать метод, принимающий эту строку и выводящий результат, является ли она правильной скобочной последовательностью или нет.

Условия для правильной скобочной последовательности те, же что и в задаче 1, но дополнительно:

- Открывающие скобки должны быть закрыты одностипными закрывающими скобками.
- Каждой закрывающей скобке соответствует открывающая скобка того же типа.

Пример:

| Входные данные | Выходные данные |
|------------------|-----------------|
| {(){} } | true |
| { } | false |
| { } | false |
| {{(){} }}{(){} } | true |