

Worksheet: Functional Programming and Recursion

Functional Programming in Java

Write a function *evens*, that uses purely functional approach (no mutation, probably want to use recursion), that takes a *GList<Integer>* and outputs a list that contains only and all the even numbers from the original list, in the same order. In which class(es) or interface(s) will you put this function?

“Functional” solution:

```
public interface GList<T> {
    static GList<Integer> evens(GList<Integer> input){
        if (input.isEmpty()) return input;
        return input.head() % 2 == 0
            ? evens(input.tail()).prepend(input.head())
            : evens(input.tail());
    }
}
```



“Object-Oriented” Solution?

```
public interface GList<T> {
    class Cons<T> implements GList<T> {
        GList<Integer> evens() {
            return head() % 2 == 0
                ? evens(tail()).prepend(head())
                : evens(tail());
        }
    }
    class Empty<T> implements GList<T> {
        GList<Integer> evens() {
            return empty();
        }
    }
}
```

