Prof. Dr. Eicke Godehardt
Frankfurt University of Applied Sciences

Exercise Sheet 1

October 19th: Strategy Pattern

Exercise 1

Try out the code from Head First for the Strategy Pattern[1], see package `headfirst.strategy`:

1. Install the sources into the IDE of your choice!
2. Compile the packages!
3. Run the `MiniDucksimulator` and the `MiniDucksimulator1` and analyze, debug and understand the code!
4. Write a `new HIS-DUCK` class extending the `Duck` class!
5. Write a `new Fly-The-HIS-Way` class implementing the `Flybehavior` interface!

Exercise 2

Implement flexible sorting algorithms.

1. Implement a Java class `Student` with two attributes `matriculationNumber` and `name`
2. Implement a `Sorter` interface

```
public interface Sorter {
    public List<Student> sort(List<Student> list);
}
```

3. Implement a `NumberSorter` and a `NameSorter` class that both implement the `Sorter` interface and that sort according to the students' matriculation numbers or names, resp.

Hint: You can implement your own sorting algorithm (copy and paste) or you can have a look at the `public static void sort(List list, Comparator c)` of the `Collections` class.

[1] http://www.headfirstlabs.com/books/hfdp/

Exercise 3

Look into the Unix/Linux man pages for `qsort` or `heapsort`. The signature reads

```
heapsort(void *base, size_t nel, size_t width,
    int (*compar)(const void *, const void *));
```

1. What is the role of the `compar` function?
2. How is this related to the Strategy Pattern?

Exercise 4

If you know Python, have a look at the following code snippet: If

```
def f(x):
    return x+3
```

and

```
def g(x):
    return x-5
```

then calling

```
map(f, [1,2,3])
```

yields

```
[4, 5, 6]
```

and calling

```
map(g, [1,2,3])
```

yields

```
[-4, -3, -2]
```

Again, how is this related to the Strategy Pattern?

Hints

- Consult the literature!
- You can work in pairs, if you want!
- If you want to learn a Java API, look into the java docs!
- Always use the same familiar IDE (suggestion Eclipse)!