UFAZ / Strasbourg University

Object Oriented Programming

Year 1 – Common curriculum

Tutorial / Lab session #3: Interfaces

Exercise 1 – String manipulation & interfaces

1. Create the interface `StringFilter` that contains only one method: `String filter(String s)`. All the implementations of this method will transform the string `s` and returns the resulting string.
2. Create the following classes that implement the interface `StringFilter` :
   a. `UpperCaseStringFilter`: converts the characters of `s` into upper case
   b. `LowerCaseStringFilter`: converts the characters of `s` into lower case
   c. `PrefixStringFilter`: keep the first *n* characters of `s`. The value of *n* is given as constructor argument
   d. `SuffixStringFilter`: keep the last *n* characters of `s`. The value of *n* is given as constructor argument
3. Write the static method `String[] filter(String[] strings, StringFilter filter)` that applies the filter to the strings and returns an array containing the transformed string.
4. Create the class `CompositeStringFilter` that implements the interface `StringFilter`. This class successively applies the filters in the array `StringFilter[] filters` (given as constructor argument) to `s`.

Exercise 2 – Basic arithmetic expressions

1. Create the interface `ArithmeticExpression` and the classes `Variable`, `Sum` and `Product` in order to be able to execute the following code snippet:

```
public static void main(String args[]) {
    Variable x = new Variable("x", 2.5);
    Variable y = new Variable("y", 4);
    ArithmeticExpression exp = new Sum(x, new Product(y, new Sum(x, y)));
    System.out.println(exp.asString()); // (x+(y*(x+y))
    System.out.println(exp.asValue());  // 28.5
    x.set(5);
    System.out.println(exp.asValue()); // 41.0
}
```

2. Create the classes `Division` and `Subtraction`