|  |  |
| --- | --- |
| Correct Seal Colors | Northeastern University  CS 5500 – Managing Software Development  Spring, 2018 – Professors Annunziato and Weintraub |

Grading for Homework 3: Structural and Behavioral Design Patterns

|  |  |
| --- | --- |
| Student Name: | Samanjate Sood |
| Repo name: | student-137 |
| Repo commit date | Feb 18 |
| Graded by | Biswaraj |
| Total Score | 77 |

|  |
| --- |
| **Bit-Vectors (25 points)** |
| -3: The solution does not check for invalid input values (e.g. negative numbers) and throw exceptions if this is the case  -2: The solution does not use constants (final static fields) for constant values that are used more than once, e.g. ‘32’.  -3: The copy() method should have used the Iterator in Q1b and in all subsequent questions  -2: 1(c) The answer does not mention or imply that the use of the Iterator pattern avoids exposing the bit-vector’s internal state (implementation details) to the client code that uses it. |
| **Adapt a Queue (10 points)** |
| -3: Incorrect implementation of the peek() method, the head of the queue is the same element which would be the next one to be removed. So, peek() should give ***que.get(0).***As a result, tests in ***testPeekIntegers()*** are also incorrect checking for wrong values.  Refer to https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html and https://piazza.com/class/jc0uywq0rsifu?cid=280 where it was clarified. |
| **Create an Abstract Syntax Tree (35 points)** |
|  |
| **Extending the AST (30 points)** |
| -10: Incorrect implementation of visitor pattern  a. Infix, Prefix and Sequence should have called **accept()** on each of its children:  e.g. for Infix, it should have been:  @Override  **public** **void** accept(ASTVisitor visitor) {  visitor.visit(**this**);  exp1.accept(visitor);  exp2.accept(visitor);  }  b. Due to the same issue above, the client code (test cases) has multiple accepts for each Node created, which again is not the right approach to implement Visitor:  Say, the client makes 200 Nodes, does it make sense to call accept() 200 times in tests? If done correctly, in the tests, accept() will be called only **once**, on the final AST. For example, for ***testNodeCount1()*** only “**seq.accept(visitor)”** would have been enough, as the visitor would have recursively called on each element. |
| **Bonus: (+3 points) used maven-archetypequickstart and mvn test working.** |
| No bonus as project is not Maven build |

Note: Please reach out to me or drop by during my office-hours in case you have any doubts or if you believe I missed something.