RPN Evaluator Ideas

# The Big If-statement

|  |  |
| --- | --- |
| Pros | Very Easy |
| Cons | Lots of code (# operand types \* # operation types)  Potentially very slow – lots of branches to test through. O(n+m)  Maintenance: everything is mixed together. |

# The Operand Polymorph

|  |  |
| --- | --- |
| Pros | Medium difficulty  Correct operand is quickly chosen.  Operation code is separated into the operand implementations that can do it. |
| Cons | Operation code is divided between several operands.  A long if statement is required to choose the operation to perform.  Slow to choose the correct operation. O(n)  Mixed operand? Must up-cast to common types first. |

# The Operation Polymorph

|  |  |
| --- | --- |
| Pros | Medium difficulty  Correct operation is quickly chosen.  Each operation only implements the operands that it works on. |
| Cons | Operand code is divided between many operation classes.  An if-statement is required to choose the operand to work on.  Slow to choose the correct operand. O(m) |

# The Double Dispatch Polymorph

|  |  |
| --- | --- |
| Pros | Correct operation is quickly chosen.  Correct operand is quickly chosen.  Every method is small and simple. |
| Cons | Implementation is spread over a large number of classes.  High difficulty.  Each operand requires a method for each operation (or vice versa). |

# Lookup Table

|  |  |
| --- | --- |
| Pros | Correct operation/operand combination is instantly chosen.  Handles mixed operands easily (nothing special).  Non-OOPs |
| Cons | High difficulty  Complex setup.  Requires a separate lookup table and table management.  Instant access requires complex initialization. |