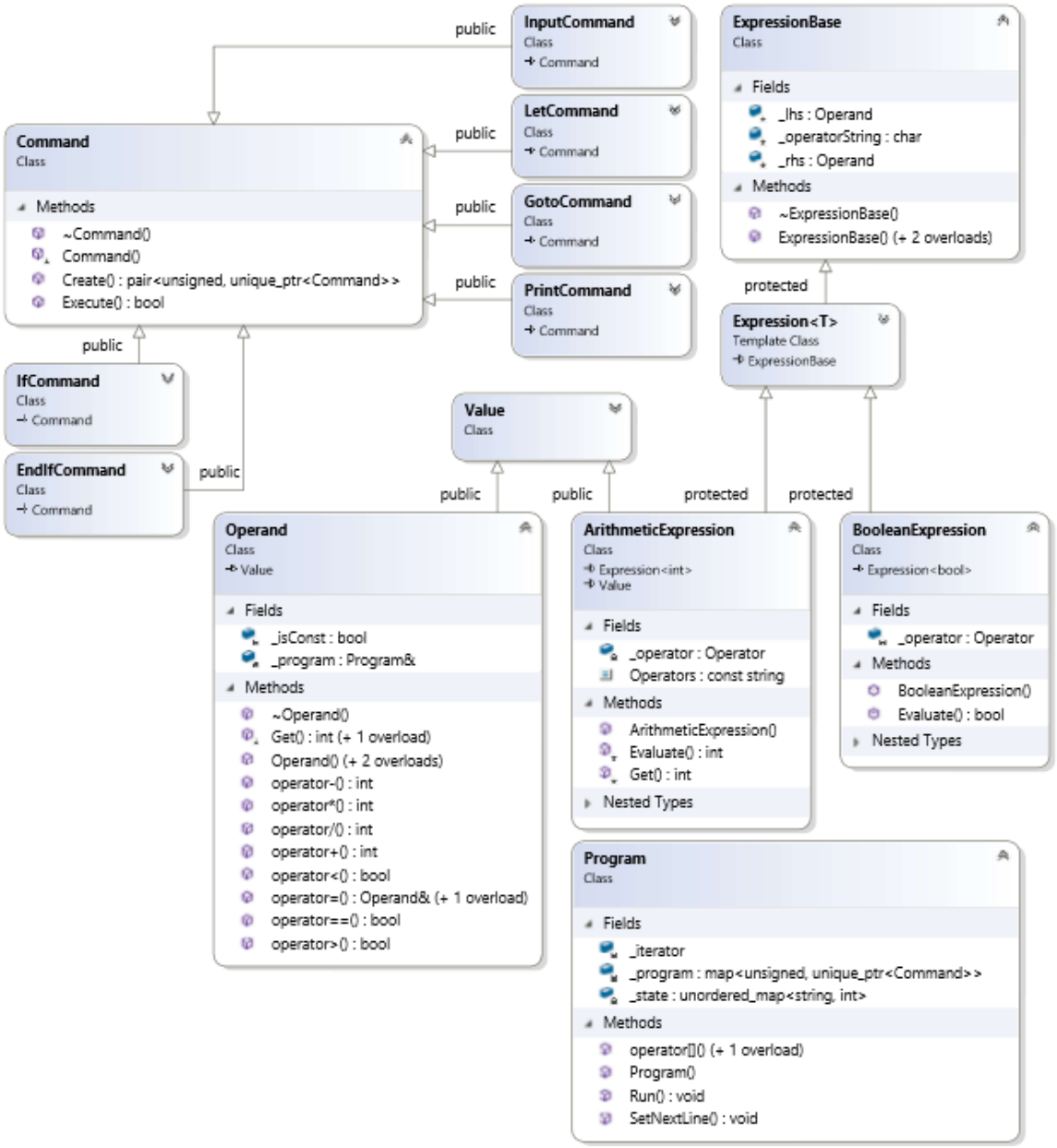
C++ Assignment – Preliminary design

Written by: Valentinas Janeiko (vj00041), Herman Yau (hy00185), Rory Jenkinson (rj00158)



Program class reads the line from a file and passes a string stream to build a specific command instance derived from Command class. Expression inside the command are constructed using ArithmeticExpression or BooleanExpression classes. Since Expressions have similar constructor logic, they are derived from ExpressionBase, which does the construction. Any variables or constant values used in commands or expressions are read and managed by Operand class. All the supported operations are implemented as overloaded operators of Operand class.

For memory management, all objects are allocated on the stack if possible, and constant references are used to pass them around the program preventing unnecessary copy operations. Since storing reference to abstract base classes is not supported, smart pointers are used to construct and do all memory management. To pass abstract objects around we use regular pointers retrieved from smart pointers. In cases we need to store one of the two different objects we use a union, and construct only one necessary object using placement new (because std::variant is only introduced in c++17).

For error handling, our design throws an exception whenever error is detected, which is then caught by the Program class. Error checks are present in all class constructors and command execution. Program class append a line number to the message and rethrows. Exceptions are caught in main() function to print the error and terminate the program.