**Use case:** Expressions

**Primary actor:** computer

**Goal in context:** Constants, identifies and arithmetic operators are used to complete a single integer value.

**Preconditions:** Code has been successfully parsed.

**Post Condition:** Expression method has returned a single integer value.

**Trigger:** When certain REVERED words (LET, IF, while, for) or a SPECIAL character is encounter by the computer.

**Scenario:**

|  |  |
| --- | --- |
|  | 1. When a special character is found the following token in the lineup until a terminator token (then, end of line, comma) are used as parameters for the expression method. |
|  | 1. Leading token is compared against a switch statement of special characters. |
|  | 1. Second Token is computed by Solution integer value according to the special character. (Example Token is 1, special character is +, solution integer is 5, new solution integer is 1 + 5 = 6.) |
|  | 1. If Token is an identifier when obtain value from Symbol Table. |
|  | 1. Remaining tokens and Solution integer is passed to the expression method recursively. |
|  | 1. This recursion continues as long as tokens remain in the container of tokens. |
|  | 1. The solution integer is returned when all tokens are used. |

**Exceptions:**

1.

2. Token doesn’t match any items in the switch statement Synax error message is displayed ”Expression is invalid Line #X.”

3. If SPECIAL symbol is divide (/) and the token is zero. Computer returns a run time error “Divide by zero Line#”.

**Priority:**

**When available:**

**Frequency of use:**

**Channel to actor:**

**Secondary actors:**

**Channels to secondary actors:**

**Open issues:**

1.