

## ADDS Prac2 Design

### Computer:

#### Member variables:

- comp\_move is a string variable that stores the move played by the computer.

#### Methods:

- getCompMove() function assigns the computer's move to the comp\_move variable.

Computer
- comp_move: string
+ string getCompMove()

### Human:

#### Member variables:

- str\_input stores the input provided by the user.
- input\_size stores the number of games to be played.
- input\_digits stores the digit extracted from the str\_input.
- Input\_space\_alphas is a string that consists of the input, but with the digit removed from it.
- input\_alphas is the final string that only consists of the human player moves, with a space separating each move.

#### Methods:

- getSize() function gets the number of games to be played. The member function isdigit() is used to extract the digits from the input string and save them into input\_digits. This is then converted to integer type variable input\_size using the atoi() function.
- getMove() function assigns the player's moves to the input\_alphas variable. The member functions isalpha() and isspace() are used to extract the alphabets and spaces from the input string and add them to a new string called input\_space\_alphas. The second for loop removes the space at the front of this string and saves it into a new string called input\_alphas. This string is passed on to the playGame() function in the Referee class.

Human
- str_input: string
- input_size: integer
- input_digits: string
- input_space_alphas: string
- input_alphas: string
- size, new_size, len, i: integers
+ string getSize()
+ string getMove()

### **Referee:**

#### **Member variables:**

- length consists of the length of the input\_alphas string.
- c\_move stores the computer's moves.
- m\_move stores the human player's moves.

#### **Methods:**

- playGame() function compares the computer and human moves i.e. c\_move and m\_move and returns the result. ASCII characters are used to compare the input string to the respective letters, namely R, P and S, and also to replace them with W(win), L(lose) or T(tie) to produce the output string.

Referee
- length: integer - c_move: string - m_move: string
+ string playGame(string c_move, string m_move, int len)

### **Testing:**

The program is to be tested using the test cases provided in the practical. These cover all the different types of tests that can be done to ensure that the program works smoothly. Apart from these, some other test cases can be tried with different string lengths and different combinations of moves. E.g.:

- **All ties:**  
Input: 4 R R R R  
Expected output: (either W, L or T - 4 times separated by space)
- **A string with only one input:**  
Input: 1 P  
Expected output: (either W, L or T - 1 time)
- **A long string input (2-digit):**  
Input: 10 R P S S R P S R P P  
Expected output: (either W, L or T – 10 times separated by space)
- **A random combination of moves:**  
Input: 0 S P R  
Expected output: <blank>