

### Assignment02-Q3:

TranslateNumber
- inp: Scanner - ph_num: String - regex: String - translated_num: String - rtn_num: String
+ main(args: String[]): void + translateNum(ph_num: String): String

#### Variables:

inp - is an instance of Scanner class that scans the user input and saves it to the ph\_num variable

ph\_num - stores the alphanumeric phone number entered by the user

regex - stores the regex format of a valid input case, used to check if user input is valid

translated\_num = stores the return value of the translateNum method (contains the numeric phone number for output)

rtn\_num - an empty string that is appended character by character as the phone number is translated

#### Methods:

main - the main method handles the user input, output and checks for valid input.

translateNum - the method that translates the alphabet characters (from its parameter ph\_num) to numbers and appends the rtn\_num string. Finally returning the rtn\_num String.

### Assignment02-Q5:

NumGuessGame
- host: Host - player: Player
+ startGame(): void + displayResult(result: String): void

#### Variables:

host - makes a Host class' instance

player - makes a Player class' instance

#### Methods:

startGame - starts the game by calling the generateNum method in Host

Host
- number: int - counter: int - guess: int
+ generateNum(): int + getNumber(): int + checkGuess(guess: int, number: int): String

**Variables:**

number - stores the generated number  
 counter - counts the number of attempts  
 guess - stores the guess of the player

**Methods:**

generateNum - randomly generates a number and stores it in the number variable  
 getNumber - gets the guess from the player by prompting the player, stores it in the guess variable.  
 checkGuess - checks if the guess variable matches, is higher or is lower than the number variable (both variables are taken as parameters) and returns a string to displayResult. It also keeps track of the number of guesses made by player. It also calls the getNumber method if a new guess is required.

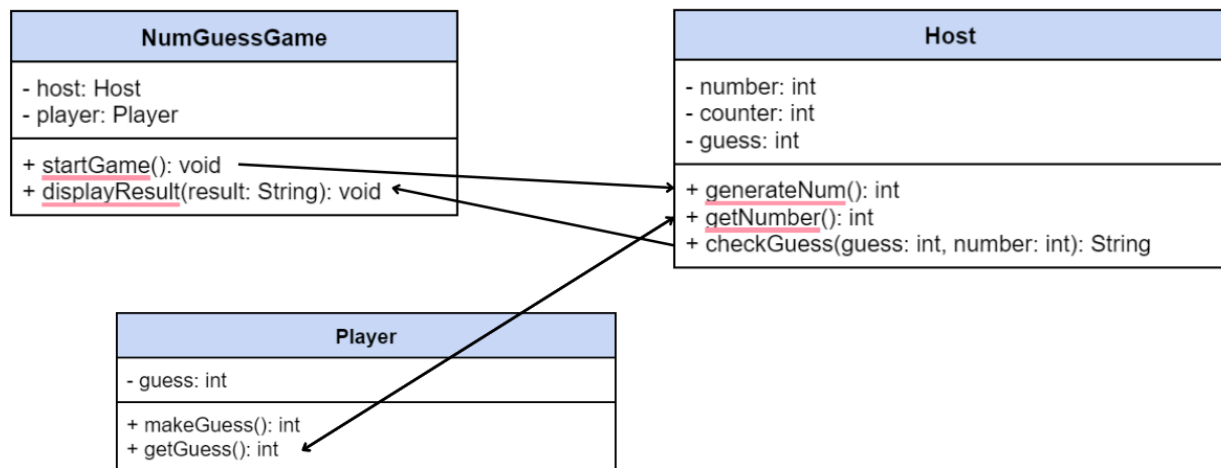
Player
- guess: int
+ makeGuess(): int + getGuess(): int

**Variables:**

guess - stores the guess of the player

**Methods:**

makeGuess - makes a random/strategic (binary search) guess for the number. Stores guessed number in the guess value.  
 getGuess - returns the guess value to the Host class (the getNumber method)



### Assignment02-Q6:

**Encapsulation Principle:** The internal state of the game (variables like number in Host, guess in Player, etc.) is encapsulated within the NumGuessingGame class. Methods provide controlled access to variables and sensitive/private information, allowing for better data protection and management.

- Encapsulation allows to protect the internal state of the object and avoid unintentional changes to data.
- It also increases flexibility to change code without worrying about the internal state or other parts of the code
- Access to other classes, variables and methods is controlled.

**Information Hiding Principle:** Details of the game's implementation, such as how the host number is generated, how guesses are checked and how player makes the guess, are hidden within the methods.

- Keeps the internal workings and data hidden from external classes, reducing dependencies.
- This allows for easier maintenance and changes in the future without affecting external components.
- Enhances security of the algorithms used and prevents misuse of data.

**Interface Principle:** The 3 separate classes allow us to implement an interface that defines the methods for starting the game, making guesses, and checking guesses all separately (ie. in different interfaces). This allows for flexibility in modifying the game or adding new types of games in the future while ensuring consistent behaviour.

- Promotes flexibility as the classes and their functions are in separate interfaces.
- The system is more modular: code can be similarly reused in other classes.
- Facilitates easier testing and maintenance by relying on abstractions of the classes rather than the whole program.