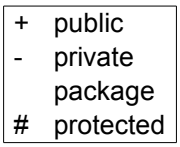
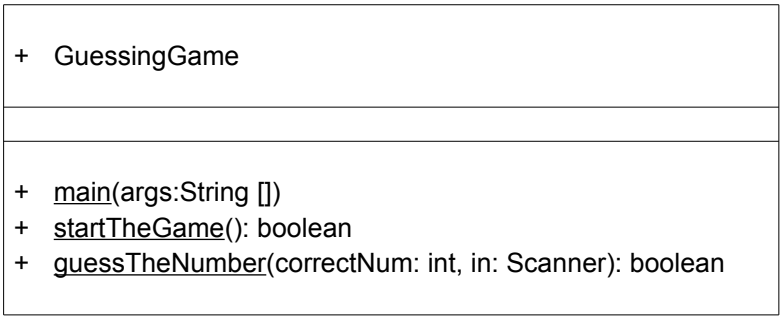
UML Class Diagram

Joseph Maples CS101
Design for a number guessing game



legend



Data Table

Joseph Maples CS101
Design for a number guessing game

Data Table for main(args)

Variable or Constant	Type	Purpose
args	String []	parameter, unused
playAgain	boolean	Condition for playing multiple games

Data Table for startTheGame()

Variable or Constant	Type	Purpose
in	Scanner	Read user input
upperBound	int	Highest number to guess
numGuesses	int	Number of guesses the user used
rand	Random	Generate random numbers
correctNumber	int	The number the user needs to guess
guessAgain	boolean	Condition for guessing multiple times

Data Table for guessTheNumber(num, in)

Variable or Constant	Type	Purpose
correctNum	int	The number of guesses to validate
in	Scanner	Read user input

Joseph Maples CS101
Design for a number guessing game

Algorithm for main(args)

```
main(args)
    playAgain ← false
    Do
      playAgain ← startTheGame()
    While playAgain is true
```

Algorithm for startTheGame()

```
startTheGame()
    Instantiate in with System.in argument
    Ask for the highest number
    upperBound ← in.nextInt()
    If upperBound is less than ten
      upperBound ← 10
      Print that the upperBound must be greater than 10.
    Instantiate rand
    correctNumber ← rand.nextInt(upperBound) + 1
    guessAgain ← false
    numGuesses ← 0
    Do
      guessAgain ← guessTheNumber(correctNumber, in)
      Increment numGuesses
    While guessAgain is true
    Print how many guesses it took and ask if they want to play again
    playAgainStr ← in.next().toLowerCase()
    if playAgainStr contains “y”
      return true
    else
      return false
```

Algorithm for guessTheNumber(correctNum, in)

```
guessTheNumber(correctNum, in)
    Ask the user for a guess
    if in.nextInt() equals correctNum
      Print correct
      return false
    else if in.nextInt() is greater than correctNum
      Print too high, ask if they want to guess again
    else
      Print too low, ask if they want to guess again
    guessAgainStr ← in.next().toLowerCase()
    if guessAgainStr contains “y”
      return true
    else
```

return false