UML Class Diagram

|  |  |
| --- | --- |
| + | public |
| - | private  package |
| # | protected |

legend

|  |  |
| --- | --- |
| + | TemplateUpdated |
|  |  |
| +  + | main(args:String [])  numChecker(userGuess: int, shouldBe: int): string |

Data Table

|  |  |  |
| --- | --- | --- |
| Main | | |
| Variable | Type | Purpose |
| args | String[] | Parameter, unused |
| check | String | If magicNumber = guess |
| playAgain | String | If user wants to play again |
| in | Scanner | Scans console for user input |
| random | Random | Generates random number |
| range | int | Upper range of random number |
| magicNumber | int | Computer generated random number |
| guesses | int | How many times a user can guess magicNumber |

|  |  |  |
| --- | --- | --- |
| numChecker | | |
| Variable | Type | Purpose |
| userguess | int | The number the user guessed |
| shouldBe | int | magicNumber for different method |
| result | String | Used as return variable in numCheck |

Algorithm for main(args)

main(args)

in ← Scanner Object

random ← Random Object

guesses ←0

check ← “not true”

playAgain ← “no”

guessAgain ←”no”

do

ask for maximum range for random number

read range

if range < 10

range = 10

magicNumber ← random number in from 0 to range

ask user for how many guesses

guesses ← read guesses

do

ask user for guess

read guess

check ← numChecker guess and magicNumber

if check is not true

print out check

add one to guesses

ask user to play again

read guessAgain

guessAgain to lower case

else

print congratulations

print number of guesses

while check is not true or guessAgain is yes

ask user to play again

read playAgain

playAgain to lower case

while playAgain is yes

numChecker (int userGuess, int shouldBe)

result ←no input

if userGuess equals shouldBe

result ← true

else

if userGuess is greater

result ←too high

else

result ←too low

return result