Christian Wendlandt

Professor George Georgiev

Computer Science 262

02/03/17

Nim UML Design Documentation

|  |  |
| --- | --- |
| NimGame | Represents and controls a Nim game in progress. |
| -HUMAN: int = 0  -COMPUTER: int = 0  -sticks: int  -turn: int  -winner: int  -mode: int | An integer representing the user.  An integer representing the computer.  The number of sticks in play.  An integer representing the current turn.  An integer representing the game winner.  An integer representing the current game mode. |
| +NimGame()  +getHuman(): int  +getComputer(): int  +getSticks(): int  +getTurn(): int  +getWinner(): int  +getMode(): int  +setSticks(int): void  +setTurn(int): void  +setWinner(int): void  +setMode(int): void  +playGame(int): void  +humanMove(int): boolean  +computerMove(): int  +computerMoveWin(): int  +userWon(): boolean | Constructs a game of Nim.  Returns the integer representing the user.  Returns the integer representing the computer.  Returns the number of sticks in play.  Returns the integer representing the current turn.  Returns the integer representing the game winner.  Returns the integer representing the current game mode.  Sets a new number of sticks in play.  Sets the integer representing the current turn.  Sets the integer representing the game winner.  Sets the integer representing the current game mode.  Progresses the game turn by turn.  Checks if a user move is valid.  Returns a random integer move for the computer.  Returns an integer move for the computer using the winning strategy  Checks if the user has won the game. |

|  |  |
| --- | --- |
| NimMatch | Represents and initializes a series of Nim games. |
| -humanScore: int  -computerScore: int  -gameNumber: int | An integer used to record the number of human wins.  An integer used to record the number of computer wins.  An integer used to record the current game number. |
| +NimMatch()  +getHumanScore(): int  +getComputerScore(): int  +getGameNumber(): int  +setHumanScore(int): void  +setComputerScore(int): void  +setGameNumber(int): void  +playMatch(): void  +updateScores(int): void  +reportScores(): | Constructs a match of Nim.  Returns the number of human wins.  Returns the number of computer wins.  Returns the current game number.  Sets the number of human wins.  Sets the number of computer wins.  Sets the current game number.  Creates games of Nim, keeps score, and asks the user if they want to play more games.  Increments the score for the winning player.  Prints the scores of both players. |

|  |  |
| --- | --- |
| Driver | Initializes the program and constructs the match. |
| +main(String[]: args): void | The program’s main. |