|  |
| --- |
| **ConnectFourBoard** |
| + dim\_row: int  + dim\_col: int  + p1: str  + p2: str  + em: str  + turn: str  + board: list of int |
| + \_init\_(self: ConnectFourBoard, rows: int, cols: int): NoneType  + drop(self: ConnectFourBoard, col: int): NoneType  + can\_drop(self: ConnectFourBoard, col: int): Bool  + other\_player(self: ConnectFourBoard): str  + whos\_turn(self: ConnectFourBoard): str  + switch\_turn(self: ConnectFourBoard): NoneType  + check\_for\_win(self: ConnectFourBoard): Bool  + check\_win\_at\_position(self: ConnectFourBoard, row: int, col: int): Bool  + alternation(self: ConnectFourBoard, x: int, y: int, dx: int, dy: int): Bool  + valid\_move(self: ConnectFourBoard, col: int, row: int): Bool  + is\_game\_over(self: ConnectFourBoard): Bool  + get\_drop\_loc(self: ConnectFourBoard, column: int): int |

|  |
| --- |
| **Disk**  0…42 |
| + NUMOFDISKS: int  - width: int  - x: int  - y: int  - line\_thickness: int  - colour: tuple of int |
| +\_\_init\_\_(self: Disk, width: int, x: int, y: int, line\_thickness: int, colour: tuple of int): NoneType + get\_width(self: Disk): int  + set\_width(self: Disk, width: int): NoneType  + get\_x(self: Disk): int  + set\_x(self: Disk, x: int): NoneType  + get\_y(self: Disk): int  + set\_y(self: Disk, y: int): NoneType  + get\_line\_thickness(self: Disk): int  + set\_line\_thickness(self: Disk, line\_thickness: int): NoneType  + get\_colour(self: Disk): tuple of int  + set\_colour(self: Disk, colour: tuple of int): NoneType |

1