

Game

- Attribute:
 - players: List<Player>
 - board: Board
- Klasse verwaltet alle Spieler und das Spielbrett

Player

- Attribute:
 - name: String
 - robot: Robot
- Methoden:
 - programRobot(List<ProgramCard> programCards)
- Klasse repräsentiert einen Spieler mit Namen und zugehörigen Roboter

Board

- Attribute:
 - checkpoints: List<Checkpoint>
 - obstacles: List<Obstacle>
 - factoryLayout: Array[][] (repräsentiert die Fabrik-Layout-Informationen)
 - als HeshMap?
- Klasse enthält Informationen über Checkpoints, Hindernisse und Layout der Fabrik

Checkpoint

- Attribute:
 - position: Position
- Klasse repräsentiert Kontrollpunkt auf dem Spielbrett

Position

- Attribute:
 - x: int
 - y: int
- Position auf Spielbrett

Obstacle

- Attribute:
 - type: ObstacleType
 - position: Position
- Klasse repräsentiert Hindernisse auf Spielbrett

Enum: ObstacleType

- Werte: WALL, PIT, CONVEYOR_BELT, usw.

Robot

- Attribute:
 - name: String
 - position: Position
 - health: int
 - programmedActions: List<Action>
- Klasse repräsentiert Roboter mit Name, Position, Gesundheitszustand und programmierten Aktionen

Action

- Attribute:
 - type: ActionType
 - parameters: Map<String, Object> (z. B. für die Bewegung: Richtung und Anzahl der Schritte)
- Aktionen die ein Robo ausführen kann

Enum: ActionType

- Werte: MOVE, ROTATE_LEFT, ROTATE_RIGHT, USE_LASER, usw.

Vererbungen und Assoziationen klar nach Klasse:

Beziehung zwischen Game und Player ist eine Assoziation mit Multiplizität, was bedeutet, dass ein Spiel mehrere Spieler haben kann.

Zwischen Player und Robot Aggregation, mehrere Roboter.

Board Checkpoint und Obstacle sind Aggregationen da Checkpoints und Hindernisse Teil des Bretts sind.

Position wird von mehreren Klassen aggregiert, gemeinsame Komponente.

Robot Klasse erbt von Position. x und y ints

Action Klasse enthält Assoziation zu ActionType.