

Poruszanie się robota po magazynie

1. Klasa: *MoveRobotsInStorehouse*

Klasa najwyższego poziomu. Pozwala wykonywać zadania wykonywane na magazynie przez roboty.

Zmienne prywatne:

- tablica obiektów klasy *RobotInStorehouse*

Metody publiczne:

- *setSpeed(robotName, velocity)*
ustawia prędkość [m/s] *velocity* poruszania się robota *robotName* (Enum *Robots*)
- *getPalette(robotName, paletteId) int*
robot określony przez *robotName* jedzie do wybranej palety *paletteId* i pobiera ją.
zwracane wartości:
 - ✓ 0 – prawidłowe zakończenie (sprawdza czy paleta ma zajętość równą false i jest w obszarze na składowanie towaru)
 - ✓ 1 – paleta pobrana/załadowana przez/na innego robota (niewykonanie funkcji)
- *putPalette(robotName, placeId) int*
robot określony przez *robotName* jedzie z załadowaną paletą do wybranego miejsca w magazynie *paletteId* (od 1 do 64)
zwracane wartości:
 - ✓ 0 – prawidłowe zakończenie (paleta została wcześniej pobrana i jest załadowana na robota i może być przeniesiona na miejsce *placeId* miejsca na składowanie jest ono wolne, w przypadku stanowiska jest miejsce w buforze)
 - ✓ 1 – miejsce *placeId* jest już zajęte (niewykonanie funkcji)
 - ✓ 2 – brak palety na robocie (niewykonanie funkcji)
- *goToStation(robotName, stationName) int*
robot określony przez *robotName* jedzie z załadowaną paletą do wybranego stanowiska *stationName* (od A do B) (Enum *Stations*)
zwracane wartości:
 - ✓ 0 – prawidłowe zakończenie (paleta została wcześniej pobrana i jest załadowana na robota i może być obsłużona na stanowisku *stationName* - jest miejsce w buforze)
 - ✓ 1 – stanowisko *stationName* nie ma miejsca w buforze) (niewykonanie funkcji)
 - ✓ 2 – brak palety na robocie (niewykonanie funkcji)
- *goToDockStation(robotName, dockStationName) int*
robot *robotName* wraca do wybranej stacji *dockStationName* (Enum *DockStations*)
zwracane wartości:
 - ✓ 0 – prawidłowe zakończenie
 - ✓ 1 – paleta jest załadowana przez robota (niewykonanie funkcji)

2. Klasa *RobotInStorehouse*

Klasa niższego poziomu. Definiuje poruszanie się robota po wybranej ścieżce i drodze za pomocą funkcji z klasy *MoveRobot* i interakcję z obiektami w magazynie (np. paletami – przyłączenie/odłączenie).

Zmienne prywatne:

obiekt klasy *MoveRobot*

obiekt klasy *Storehouse*

Metody publiczne:

- *goOnPath(pathId)* wjedź na początek ścieżki *pathId* od bieżącej pozycji robota
- *goPathToEnd(pathId)* jedź ścieżką do końca ścieżki *pathId*
- *goOutOfPath(placeId)* zjedź ze ścieżki na wysokości miejsca w magazynie na palety
- *joinPallette(paletteId)* przyczep paletę *paletteId* do robota
- *unjoinPallette(paletteId)* odłącz paletę *paletteId* od robota
- *setSpeed(velocity)* ustaw prędkość *velocity* poruszania się robota

3. Klasa: *MoveRobot*

implementuje metody najniższego poziomu służące do poruszania się robota bez mechanizmu zliczania czasu pracy robota (timer aktywujący się, gdy robot się porusza)

Metody publiczne:

- *goDestinationPoint(destinationPoint)*
przemieszczenie robota z bieżącej pozycji do pozycji przeznaczenia *destinationPoint* wyrażonej w współrzędnych xy.
- *rotate(stopnie)*
rotacja w miejscu z określonym kątem *stopnie*
- *setSpeed(predkosc)*
ustawia prędkość poruszania się robota *predkosc*
- *updatePosition(position)*
aktualizuje pozycję robota pobraną z symulatora vrep
- *getPosition()*
zwraca aktualną pozycję robota
- *getName()* string
zwraca nazwę robota

4. Klasa *Storehouse***Zmienne prywatne:**

- tablica obiektów klasy *Place*
przechowuje pozycje miejsc w magazynie dostępnych pod *placeId* (index tablicy)
- tablice obiektów klasy *Palette*
przechowuje aktualną pozycję palet w magazynie dostępnych pod *paletteId* (index tablicy)
- tablica obiektów klasy *Station*
przechowuje pozycje stanowisk w magazynie dostępnych pod *stationName* (index tablicy)
- tablica *dockStations*
przechowuje pozycje stacji dokujących w magazynie dostępnych pod *stationName* (index tablicy)
- tablice ścieżek *Paths*
przechowuje wszystkie dostępne ścieżki w magazynie dostępnych pod *pathId* (index tablicy)

Metody publiczne:

- *reset()*
przywraca pozycje początkową wszystkich obiektów w magazynie (robotów, palet)

- *getPalletePosition(paletteId) Point*
zwraca aktualną pozycję palety
 - *getStationEmptyPosition(stationName) Point*
zwraca wolną pozycję stanowiska/bufora, w której nie znajduje się robot i jest najbliższa stanowisku (zaimplementowane w klasie *Station*).
 - *getPathBeginPosition(pathId) Point*
zwraca pozycję początkową ścieżki względem pozycji robota
 - *getPathEndPosition(pathId) Point*
zwraca pozycję końcową ścieżki względem pozycji robota
4. Enum *Robots*
zawiera nazwy robotów (Robot1-Robot6)
 5. Enum *Stations*
zawiera nazwy stanowisk obsługujących (A-D)
 6. Enum *Dockstations*
zawiera nazwy stacji dokujących ładujących roboty (A-F)
 6. Enum *PaletteAction*
zawiera nazwy akcji, które wykonywane są na magazynie za pomocą robota (ON_ROBOT, IN_STATION, IN_BUFFER, IS_READY)
 7. Enum *StorehouseOperations*
zawiera dopuszczalne operacje na magazynie (TAKE_PRODUCT, PUT_PRODUCT, CONTROL_PALLETE)
 8. Struktura *Palette*
zawiera:
położenie początkowe
położenie aktualne
nazwę w symulatorze vrep
 9. Struktura *Path*
zawiera
pozycja początku ścieżki
pozycja końca ścieżki
 10. Struktura *PathWay*
definiuje drogę jaką porusza się robot
zawiera listę struktur *Path*
 11. Struktura *Station*
zawiera:
obiekt klasy *StationBuffer*
pozycję stanowiska *position*
id ścieżki *pathId* prowadzącej do stanowiska (od poziomu 1 bufora)
 12. Struktura *StationBuffer*
zawiera
positionLevel1 pozycja bliżej od stanowiska
positionLevel2 pozycja dalej od stanowiska
pathId1 id ścieżki prowadzącej do pozycji poziom 1 *positionLevel1*
pathId2 id ścieżki prowadzącej do pozycji poziom 2 *positionLevel2*

actualCount aktualna ilość palet w buforze