

Poruszanie się robota po magazynie

1. Klasa: *MoveRobotsInStorehouse*

Zmienne prywatne:

obiekty 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, paletteId) 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 *idMiejscaWMagazynie* miejsca na składowanie jest ono wolne, w przypadku stanowiska jest miejsce w buforze)
 - ✓ 1 – miejsce *idMiejscaWMagazynie* 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*

definiuje poruszanie się robota po wybranej ścieżce za pomocą funkcji z klasy *Robot* i interakcję z obiektami w magazynie (paletami).

Zmienne prywatne:

obiekt klasy *MoveRobot*

obiekt klasy *Storehouse*

Metody publiczne:

- *goOnPath(pathId)* wjedź na ścieżkę *pathId*
- *goPathToEnd(pathId)* jedź ścieżką do końca ścieżki
- *goOutOfPath(paletteId)* zjedź ze ścieżki na wysokości palety
- *joinPalette(paletteId)* przyczep paletę do robota
- *unjoinPalette(paletteId)* odłącz paletę od robota
- *setSpeed(velocity)*

3. Klasa *MoveRobot*

implementuje metody najniższego poziomu

mechanizm zliczania czasu pracy (timer aktywujący się, gdy robot się porusza)

Metody publiczne:

- *goStraight(dystans)*
jazda na wprost na określoną odległość *dystans*
- *rotate(stopnie)*
rotacja w miejscu z określonym kątem *stopnie*
- *setSpeed(predkosc)*
ustawia prędkość poruszania się robota *predkosc*
- *getPosition()*
zwraca aktualną pozycję robota

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

- *tablica places*
przechowuje pozycje miejsc w magazynie dostępnych pod *idMiejscaWMagazynie*
- *tablice obiektów klasy Palette*
przechowuje aktualną pozycję palet w magazynie
- *tablica stations*
przechowuje pozycje stanowisk w magazynie dostępnych pod *stationName*
- *tablica dockStations*
przechowuje pozycje stacji dokujących w magazynie dostępnych pod *stationName*
- *tablice ścieżek Paths*
- *tablice obiektów klasy StationBuffer*

Metody publiczne:

- *reset()*
przywraca pozycje początkową wszystkich obiektów w magazynie (robotów, palet)
- *getPalettePosition(paletteId)*
zwraca aktualną pozycję palety klasa *Point*
- *getStationPosition(stationName)*
zwraca pozycję stanowiska klasa *Point*
- *getDockStationPosition(dockStationName)*
zwraca pozycję stacji dokującej klasa *Point*
- *getPathBeginPosition(pathId)*
zwraca pozycję początkową ścieżki klasa *Point*
- *getPathEndPosition(pathId)*
zwraca pozycję końcową ścieżki klasa *Point*

4. Enum *Robots*

zawiera nazwy robotów

5. Enum *Stations*

zawiera nazwy stanowisk obsługujących

6. Enum *Dockstations*

zawiera nazwy stacji dokujących/ ładujących

6. Enum *PaletteAction*

onRobot, inStation, inBuffer, isReady

7. Struktura *Palette*

zawiera

położenie początkowe

położenie aktualne

działanie wykonywane na palecie, jest na: stanowisku magazynie, robocie, w buforze, w magazynie

8. Struktura *Path*

zawiera

pozycja początku ścieżki

pozycja końca ścieżki

9. Struktura *Station*

zawiera

czasObsługi

10. Struktura *StationBuffer*

zawiera

position1 pozycja bliżej od stanowiska

position2 pozycja dalej od stanowiska

~~stationName id stanowiska do którego należy~~

~~paletteId1 id palet, które się w nim znajdują~~

~~paletteId2 id palet, które się w nim znajdują~~

~~robotName1 id robotów, które się w nim znajdują~~

~~robotName2 id robotów, które się w nim znajdują~~

actualCount aktualna ilość palet w buforze