

ID	5
Verfahren	Parametrisierter JUnit-Test
Klasse	Ant
Methoden	run() Hauptalgorithmus einer Ameise: Berechnen der möglichen Wegstrecke
Vorbedingung	City(1), City(2), City(3) (Städte haben untereinander die Distanz 2) (Zwischen allen Städten liegt ein Pheromonwert von jeweils 1.5)
Eingaben	-
Erwartetes Ergebnis	Route der Ameise > 0 Aktuelle Stadt der Ameise == City(1)
Ergebnis	Route der Ameise == 4 Aktuelle Stadt der Ameise == City(1) Test erfolgreich

ID	6
Verfahren	Parametrisierter JUnit-Test
Klasse	Ant
Methoden	updatePheromones() Aktualisieren der Pheromonmatrix der Kolonie
Vorbedingung	City(1), City(2), City(3) (Städte haben untereinander die Distanz 3) Ameise hat eine Route von City(1), City(2), City(3), City(1)
Eingaben	-
Erwartetes Ergebnis	$\{\{1, 1+\frac{1}{3}, 1\}, \{1, 1, 1+\frac{1}{3}\}, \{1+\frac{1}{3}, 1, 1\}\}$
Ergebnis	$\{\{1, 1+\frac{1}{3}, 1\}, \{1, 1, 1+\frac{1}{3}\}, \{1+\frac{1}{3}, 1, 1\}\}$ Test erfolgreich

ID	7
Verfahren	Parametrisierter JUnit-Test
Klasse	Ant
Methoden	visitCity(City b) Wechsel der aktuellen Stadt sowie Aktualisierung der Route
Vorbedingung	City(1), City(2) (Städte haben untereinander die Distanz 2)
Eingaben	City(2)
Erwartetes Ergebnis	Aktuelle Stadt der Ameise == City(2)
Ergebnis	Aktuelle Stadt der Ameise == City(2) Test erfolgreich