

<b>ID</b>	18
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	getDistance(City city, City neighbour): double Rückgabe der Distanz zweier Städte zueinander
<b>Vorbedingung</b>	City(1), City(2) City(2) hat zu City(1) eine Distanz von 5
<b>Eingaben</b>	City(1), City(2)
<b>Erwartetes Ergebnis</b>	5
<b>Ergebnis</b>	5 <b>Test erfolgreich</b>

<b>ID</b>	19
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	getNeighboursSize(): int Rückgabe der Größe der Distanzmatrix
<b>Vorbedingung</b>	City(1), City(2) City(2) hat zu City(1) eine Distanz von 5
<b>Eingaben</b>	-
<b>Erwartetes Ergebnis</b>	Distanzmatrix hat eine Größe von 2
<b>Ergebnis</b>	Distanzmatrix hat eine Größe von 2 <b>Test erfolgreich</b>

<b>ID</b>	20
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	getSpecifiedNeighbours(City currentCity): double[] Rückgabe der Distanzen zu allen erreichbaren Städten ausgehend von der übergebenen Stadt
<b>Vorbedingung</b>	City(1), City(2), City(3) Städte haben untereinander eine Distanz von je 1
<b>Eingaben</b>	City(1)
<b>Erwartetes Ergebnis</b>	{1,1}
<b>Ergebnis</b>	{1,1} <b>Test erfolgreich</b>

<b>ID</b>	21
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	getNeighbours(): double[][] Rückgabe der kompletten Distanzmatrix
<b>Vorbedingung</b>	City(1), City(2), City(3) Städte haben untereinander eine Distanz von je 1
<b>Eingaben</b>	-
<b>Erwartetes Ergebnis</b>	Distanzmatrix hat eine Länge von 3
<b>Ergebnis</b>	Distanzmatrix hat eine Länge von 3 <b>Test erfolgreich</b>

<b>ID</b>	22
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	addNeighbours(City a, City b, double distance): int Abspeichern der übergebenen Distanz zwischen a und b in die Pheromonmatrix
<b>Vorbedingung</b>	-
<b>Eingaben</b>	City(1), City(2), 1
<b>Erwartetes Ergebnis</b>	Checksumme ist 1 Distanzmatrix hat eine Länge von 2 Distanz zwischen City(1) und City(2) beträgt 1
<b>Ergebnis</b>	Checksumme ist 1 Distanzmatrix hat eine Länge von 2 Distanz zwischen City(1) und City(2) beträgt 1 <b>Test erfolgreich</b>

<b>ID</b>	23
<b>Verfahren</b>	Parametrisierter JUnit-Test
<b>Klasse</b>	Landscape
<b>Methoden</b>	addNeighbours(City a, City b, double distance): int Abspeichern der übergebenen Distanz zwischen a und b in die Pheromonmatrix
<b>Vorbedingung</b>	City(1), City(2) City(1) hat eine Distanz von 1 zu City(2)
<b>Eingaben</b>	City(1), City(2), 1
<b>Erwartetes Ergebnis</b>	Checksumme ist -1 Distanzmatrix hat eine Länge von 2
<b>Ergebnis</b>	Checksumme ist -1 Distanzmatrix hat eine Länge von 2 <b>Test erfolgreich</b>