

Obsługiwane komendy

RoboCompass ma 10 poleceń rysowania i 12 komend pomocniczych. Komendy rysowania mogą być przypisane do oznaczeń i można odwołać się do nich w innych komendach. W efekcie, polecenia mogą być łączone w twórczy sposób w celu tworzenia wszelkiego rodzaju konstrukcji geometrycznych.

Komendy rysowania

- `point(x,y)` Przykład:- `point(3,4)` lub `A=point(3,4)` gdzie A jest oznaczeniem punktu.
- `line(x1,y1,x2,y2)` Przykład:- `line(0,0,-2,5)` lub `C=line(A,B)` gdzie A,B oznaczają punkty. Ewentualnie może być podana długość jako `line(A,B,3)`, gdzie 3 to długość odcinka na półprostej AB za punktem B.
- `arc(originPoint,radius,angleFrom,upto)` Przykład:- `arc(point(2,2),3,40,20)` Użyj punktu (2,2) jako środka okręgu o promieniu 3, zaczynając łuk w punkcie, w którym promień tworzy kąt o mierze 40° z poziomem i doda następne 20° w kierunku przeciwnym do ruchu wskazówek zegara (przy ujemnej - zgodnie). Aby skopiować promień, daj pierwszy parametr jako linię (odcinek) lub dwa punkty, jak pokazano tutaj `arc(point1,point2,originPoint,angleFrom,upto)`
- `perp(line,passThroughPoint,length=10)` Przykład:- `A=perp(line(1,2,3,4),point(1,2))` rysuje linię prostopadłą do danej linii(1,2,3,4) przechodzącą przez punkt (1,2). Ostatni opcjonalny parametr tej komendy to długość linii prostopadłej, domyślnie 10.
- `parallel(line,passThroughPoint,length=10)` Przykład:- `parallel(line(1,2,3,4),point(1,2))`
- `angle(point1,point2,degrees)` Przykład:- `angle(A,B,45,1)` Dwa punkty A i B wyznaczają linię podstawową. Ostatni współczynnik to opcjonalny parametr, który steruje położeniem kątomierza (wierzchołek kąta). Wartość 0 umieszcza go w A, 1 w B i 0.5 w środku AB.
- `polygon(comma separated points)` Przykład:- `polygon(A,B,C)` rysuje trójkąt o wierzchołkach A, B, C
- `findangle(2 lines or a polygon)` Przykład:- `findangle(A,B)` określa, kąt pomiędzy dwoma liniami A, B lub w wielokącie `findangle(C)` (C jest oznaczeniem wielokąta), określa kąty wewnętrzne wielokąta.
- `fill(A,B,fillType=0,output=1)` wypełnia wnętrza figur A,B. Wypełniane kształty mogą być wycinkiem okręgu, wielokątem itp. Komenda może zawierać dowolną ich liczbę. Domyślny typ wypełnienia `fillType=0` suma obszarów, 1 część wspólna i 2 to różnica.
`fill(A)`,
`fill(arc(0,0,3,0,360),D,polygon(2,3,4,6,1,0))`
- `trace(comma separated points)` kreśli krzywą przechodzącą przez podane punkty.
`trace(point(0,0),point(1,sin(30)),point(1,sin(60)),point(1,sin(120)),B)`

Jako zasada ogólna, w przypadku gdy jakikolwiek punkt występuje w wyrażeniu, możemy użyć `point(2,3)` lub użyć jego oznaczenia lub użyć polecenia pomocniczego, które obliczy i zaznaczy punkt. Podobnie wszędzie tam, gdzie oczekuje się linii, możemy dać oznaczenie tej linii lub oznaczenie dwóch punktów.

Przykładowo `arc(A,B,point(2,1),50,40)` wykorzystuje odległość od A do B jako promień i rysuje łuk o środku (2,1). Uwaga: zamiast punktu (2,1), możemy użyć "C", jeśli mamy punkt oznaczony literą C.

Komendy pomocnicze (przydatne do obliczeń)

- `dist(point1, point2)` oblicza odległość między dwoma punktami. Możemy także dać `dist(C)` jeśli C oznacza linię (tj. odcinek).
- `X(point)` daje współrzędną x punktu `A= X(point(2,1))`
- `Y(point)` daje współrzędną y punktu `B= Y(point(1,2))`
- `pos(polygon or line or arc,index)` zwraca punkt o danym indeksie. Jeśli wielokąt A ma 4 wierzchołki, `pos(A,3)` zwraca trzeci punkt, podobnie jeśli oznaczenie linii jest B to indeks 2 zwraca punkt końcowy. Przykład:- `pos(B,2)`
- `intersect(object1,object2,index=1)` daje punkt przecięcia dowolnych dwóch obiektów (z wyjątkiem punktów). Przykład `G=point(intersect(D,E))` domyślnie daje pierwszy punktu przecięcia. Dla drugiego i trzeciego punktu przecięcia, wpisz odpowiednio 2 lub 3. Na przykład `intersect(D,E,2)` daje drugi punkt przecięcia.

- `reflect(object,line)` daje obraz obiektu w symetrii osiowej. Przykład:- `reflect(D,A)` gdzie D jest oznaczeniem przekształcanego obiektu, a A jest osią symetrii.
- `rotate(object,angle,withrespectTo=point(0,0))` daje obraz obiektu w obrocie o podany kąt. Domyślnie obrót jest względem punktu (0,0), który może być ewentualnie zastąpiony dowolnym innym przez podanie jego współrzędnych.
- `translate(object,x,y,withrespectTo=point(0,0))` przesuwa obiekt o wektor o współrzędnych x,y (domyślnie) lub o wektor o końcu w (x,y) i początku w podanym punkcie.
`translate(arc(2,3,3,0,180),2,3,point(2,1))` lub `translate(G,2,4)`
- `dilate(object,scaleFactor,withRespectTo=point(0,0))` obraz danego obiektu w jednokładności o środku w podanym punkcie (domyślnie (0,0)) i podanej skali. `dilate(point(3,1),2)` .
- `project(point1,line)` rzutuje prostopadłe wprowadzony punkt na prostą i daje obraz tego rzutu. Przykład:-`project(A,line(0,0,1,0))` daje obraz rzutu prostokątnego punktu A na daną prostą.
- `interpolate(point1,point2,ratio)` Przykład:- `interpolate(A,B,0.5)` Stosunek 0,5 daje punkt środkowy AB
- `hide(comma separated labels)` Przykład:- `hide(A,B,C,D)` ukrywa obiekty A,B,C,D. Aby ponownie pokazać, użyj `show(A,B,C,D)`
- `group(comma separated objects)`Przekształcenia mogą dotyczyć wielu obiektów jednocześnie jeżeli je pogrupujemy. Na przykład, aby obrócić razem łuk 'A' i wielokąt 'B', użyj `C=group(A,B)` i następnie `D=rotate(C,120)` gdzie C jest oznaczeniem grupy. Grupy mogą być również wkładane jedna w drugą.
- Różne kształty mogą być tworzone przez zastosowanie operacji logicznych do prostych figur. Na przykład `and(A,B,C)`daje część wspólną A,B,C gdzie A,B,C to łuki lub wielokąty. Analogicznie `or(A,E,polygon(0,0,2,3,4,1))` i `diff(A,B)` dają, odpowiednio, sumę i różnicę danych obszarów. Obszary bez części wspólnej nie są obsługiwane (wynik operacji xor) ale ten sam rezultat może być uzyskany poprzez łączenie wielokątów za pomocą komendy `group`.
- Wszystkie elementarne funkcje matematyczne, takie jak `sin,cos,tan,asin,acos,atan,log,sqrt,max,min` są obsługiwane. Uwaga: Miary kątów w funkcjach trygonometrycznych wyrażane są w stopniach.