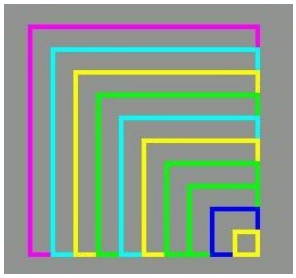
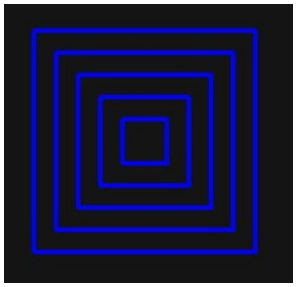
3. Arbeit DiffKurs 8 Informatik - Sporenberg Schuljahr 2010/11 Marl, 6.04.2011

1.Aufgabe: a)Schreibe ein Programm, das ein Quadrat mit der Seitenlänge 100 zeichnet. Dabei soll links gedreht werden. Der Befehl WIEDERHOLE darf hier nicht verwendet werden.

b)Verändere das Programm so, dass damit Quadrate beliebiger Länge gezeichnet werden. Hier muss der Befehl WIEDERHOLE verwendet werden.

c)Mit Hilfe eines rekursiven Programms sollen viele Quadrate, die wie in den beiden Abbildungen ineinander geschachtelt werden, gezeichnet werden. Schreibe dazu jeweils ein Programm.





d)Erläutere den Begriff der rekursiven Programmierung an dem Programm aus c).

Lösung:

a)
```
to quadrat
 cs
 fd 100 lt 90
 fd 100 lt 90
 fd 100 lt 90
 fd 100 lt 90
end
```

b)
```
to quadrat :laenge
   cs
   repeat 4 [fd :laenge lt 90]
end
```

c)
```
to VieleQuadrate :laenge
   if :laenge<0 [stop]
   repeat 4 [fd :laenge lt 90]
    VieleQuadrate :laenge-10
end
```

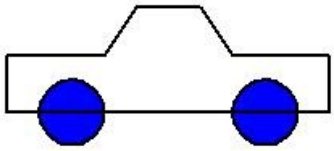
d)
```
to VieleQuadrate :laenge
 If :laenge<20 [Stop]
 repeat 4 [fd :laenge lt 90]

 PU FD 20 LT 90
FD 20
RT 90 PD

 VieleQuadrate :laenge-40
end
```

e)Rekursive Programmierung: Selbstaufruf des Programms (hier: VieleQuadrate), Veränderung der übergebenen Variablen (hier: VieleQuadrate: laenge-10) und für den Abbruch des Programms benötigt man eine Abbruchbedingung (hier: if :laenge<0 [stopp]

2.Aufgabe: a) Erläutere den Befehl **setpos** (1. Angabe der Koordinaten durch Zahlen 2. Angabe der Koordinaten durch Variable).
b) Programmiere die nebenstehende Figur. Erstelle dazu eine Skizze des Fahrzeuges und beschrifte die Punkte mit den entsprechenden Koordinaten (die linke untere Ecke hat die Koordinaten (-300/20)). Schreibe dazu das entsprechende Programm. Für die Zeichnung muss der Befehl **setpos** verwendet werden.



c) Die linke untere Ecke wird jetzt mit Hilfe zweier Variablen (XKoordinate und YKoordinate) angesprochen. Schreibe dazu das entsprechende Programm **AutoMitVariable**. Der Aufruf sollte dann wie folgt erfolgen:
AutoMitVariable -300 20

Lösung:

a) Angabe der Koordinaten durch Zahlen: setpos[40 200]

Angabe der Koordinaten durch Variable über list: setpos(list :xkoordiante :ykoordinate)

```
b)to auto
Cs setpensize[2 2] setpc 0
pu
setpos[-260 20] pd  circle 20 pu setpos[-260 20] setfc [0 0 255] fill
pu
setpos[-140 20] pd  circle 20 pu setpos[-140 20] setfc [0 0 255] fill
pu
setpos[-300 20] pd
setpos[-100 20] setpos[-100 55] setpos[-160 55]
setpos[-180 85] setpos[-220 85] setpos[-240 55]
setpos[-300 55] setpos[-300 20]

ht
end

c)  to autoVariable :xko :yko
Cs setpensize[2 2] setpc 0

pu
setpos(list :xko+40 :yko) pd  circle 20
pu setpos(list :xko+40 :yko) setfc [0 0 255] fill
pu
setpos(list :xko+160 :yko) pd  circle 20
pu setpos(list :xko+160 :yko) setfc [0 0 255] fill

pu
setpos(list :xko :yko) pd setpos(list :xko+200 :yko)
setpos(list :xko+200 :yko+35) setpos(list :xko+140 :yko+35)
setpos(list :xko+120 :yko+65) setpos(list :xko+80 :yko+65)
setpos(list :xko+60 :yko+35) setpos(list :xko :yko+35) setpos(list :xko :yko)

ht
end
```

Aufruf: autoVariable -300 20

3.Aufgabe: Auf einem waagerechten Holzbrett (s. Abbildung) soll ein Rad von rechts nach links rollen.
a) Schreibe eine Prozedur, die das Brett zeichnet (Breite des Brettes: 800 Pixel, Höhe des Brettes 20 Pixel.

b) Schreibe eine Prozedur, die das Rad (Radius des Kreises: 40 Pixel) so zeichnet wie in der Abbildung. Dabei soll der Mittelpunkt des Kreises die Koordinaten xko und yko haben. Die Prozedur hat folgende vorgegebene Struktur:
to Kreis :xko :yko
…..end

c) Schreib eine Prozedur, die den Kreis in der Farbe weiß zeichnet.
d) Schreibe jetzt eine rekursive Prozedur, die den Kreis von rechts nach links bewegt. Überleg dir, welche Koordinate verändert werden muss und begründe dies.

Die Prozedur hat folgende vorgegebene Struktur:
to Bewegung :xko :yko
…..
end

Lösung:

a)

```
to Holzbrett
setpensize[3 3]
setpc 0
pu
setpos[-400 0] pd
setpos[400 0] setpos[400 20]
setpos[-400 20] setpos[-400 0]
pu setpos[-390 10]
setfc[100 100 220] fill
ht
end
```

b)

```
to kreisBlack :xko :yko
setpc 0 setpensize[3 3]
pu setpos(list :xko :yko)
pd circle 20
pu setpos(list :xko :yko)
setfc[255 255 0]fill
ht
end
```

c)

```
to kreisWhite :xko :yko
setpc 7 setpensize[3 3]
pu setpos(list :xko :yko)
pd circle 20
pu setpos(list :xko :yko)
setfc 7 fill
```

```
ht
end
```

d)

```
to bewegungHorizontal :xko :yko

if :xko<-250[kreisBlack :xko :yko stop]

kreisBlack :xko :yko
wait 3
cs

bewegunghorizontal :xko-1 :yko
end
```

4. Aufgabe: Ein Lichtstrahl soll an einer spiegelnden Decke reflektiert werden.
a)Schreib ein Programm zur Darstellung einer Decke.
b)Schreib ein Programm, das einen Strahl an der Decke reflektiert.

Lösung:

a)

```
to Decke
setpensize[3 3]
setpc 0
pu
setpos[-400 0] pd
setpos[400 0] setpos[400 20]
setpos[-400 20] setpos[-400 0]
pu setpos[-390 10]
setfc[100 100 220] fill
ht
end
```

b)

```
to reflexion :xwert :ywert :winkely
if :ywert>200[make "winkely -:winkely]

setpos(list :xwert :ywert)

wait 10
reflexion :xwert :ywert+:winkely :winkely
end
```

MSWLogo-Befehl	Beispiel/Erläuterung
Allgemein	
;;	Einfügen eines Kommentars
~ (sog. Tilde)	Verkettung zweier Zeilen
print	print "Hallo / erscheint im Commandfeld
cleartext	löscht den Textschirm
readchar, readword, readlist	Dateneingabe (Tastatur, Datei)
keyboardon, keyboardoff	Tastaturbefehle
mouseon, mouseoff, mousepos	Mausbefehle
openread,setread,setreadpos	Dateibefehle
Turtlegrafik	
setpos	setzt die Turtle auf die angegeb. Position
setheading	setzt die Richtung der Turtle: setheading 90
setx, sety	
left, lt, right, rt	Links- bzw. Rechtsdrehung der Turtle
home	setzt die Turtle in Bildmitte ohne zu löschen
forward, fd	fd 100, 100 Pixel vorwärts
back, bk	bk 100, 100 Pixel rückwärts
heading	liefert den aktuellen Kurs, make "k1 heading
pos	liefert die aktuelle Position, show pos
xcor, ycor	liefert die x- bzw. y-Koordinate, show xcor
setpencolor, setpc	setpc[255 0 0] - Penfarbe rot
clearscreen, cs	löscht den Bildschirm
setfloodcolor mit fill	füllt ein Polygon mit einer Füllfarbe setfloodcolor[0 255 0]fill - ergibt grün
clean	löscht den Grafikbildschirm
hideturtle, hat	versteckt die Turtle
showturtle, st	zeigt die Turtle
Prozeduren und Funktionen	
to … end	Beginn … Ende einer Prozedur
make	speichert einen Wert, make "laenge 100
stop	beendet die aktuelle Prozedur
Kontrollstrukturen	
repeat	repeat 5 […..]
if	if :laenge<0 [stop]
ifelse	ifelse :laenge<0 [stop] [repeat 4[…]]
halt	stoppt das Programm
wait	wartet n/60 s, wait 120 -> 120/60 = 2 s