



Risanje



Programiranje 2, Tomaž Dobravec

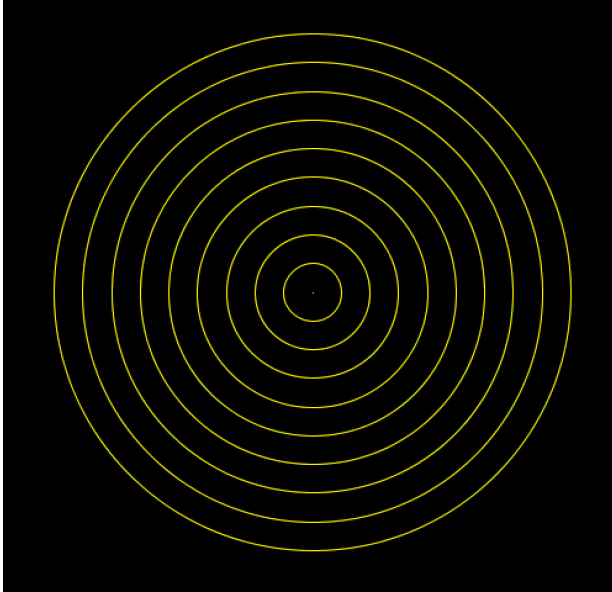


Uporaba knjižnice StdDraw

- ▶ Za (udobno) delo z grafiko uporabimo knjižnico, na primer StdDraw (by Robert Sedgewick);
- knjižnico `stdlib.jar` (<https://introcs.cs.princeton.edu/java/stdlib/stdlib.jar>) posnemi v folder `lib` in jo dodaj v projekt:
 - Netbeans: Project Properties / Libraries / Add JAR/Folder
 - IntelliJ IDEA: Module Settings / Libraries / + (New project library)
- ▶ Za “ročno” uporabo (v lupini) uporabi stikalo `-cp`:

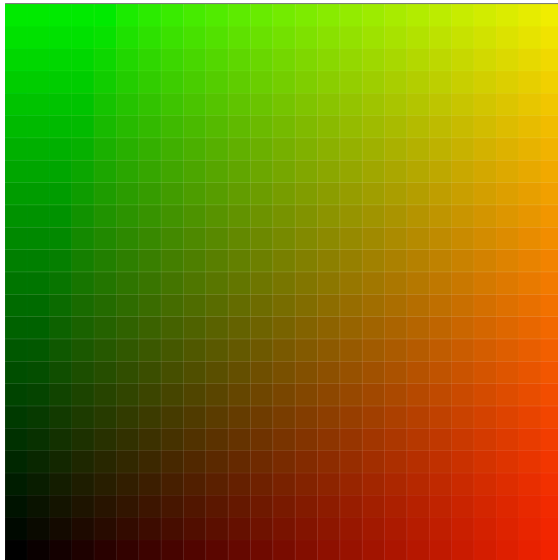
```
javac -cp .:stdlib.jar Risanje.java  
java -cp .:stdlib.jar Risanje
```

Tarča

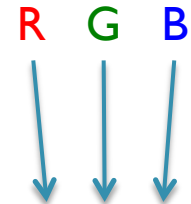


- `StdDraw.setScale(-100, 100);`
- `StdDraw.clear(Color.BLACK);`
- `StdDraw.setPenColor(Color.yellow);`
- `StdDraw.circle(...)`

Barvni kvadratki



- mreža 25 x 25;
- spodaj levo začnemo z barvo (0,0,0);
- v vodoravni smeri se povečuje vsebnost rdeče, v navpični smeri pa vsebnost zelene barve; modre ni



```
StdDraw.setScale(0,250);
```

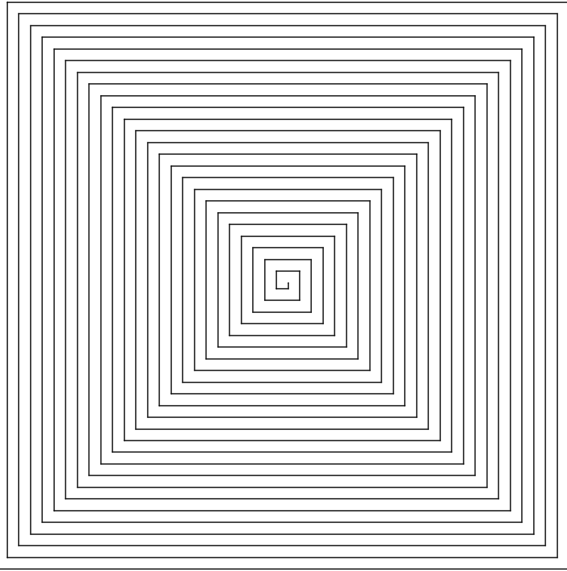
```
...
```

```
StdDraw.setPenColor(10*i, 10*j, 0);
```

```
StdDraw.filledRectangle(10*i+5, 10*j+5, 5, 5);
```



Kvadratna spirala



- Imamo 4 smeri izrisa

$\{\{1, 0\}, \{0, -1\}, \{-1, 0\}, \{0, 1\}\}$

- Sprememba smeri:

`trSmer = (trSmer + 1) % 4;`

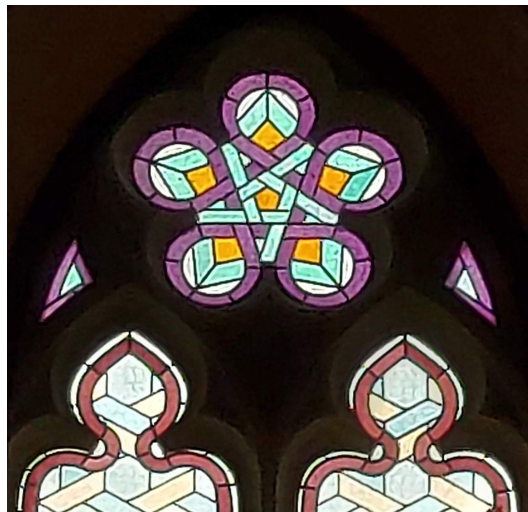
```
double nX = trX + smeri[trSmer][0] * trDolzina;  
double nY = trY + smeri[trSmer][1] * trDolzina;
```

```
StdDraw.line(trX, trY, nX, nY);
```

```
trX = nX;  
trY = nY;
```



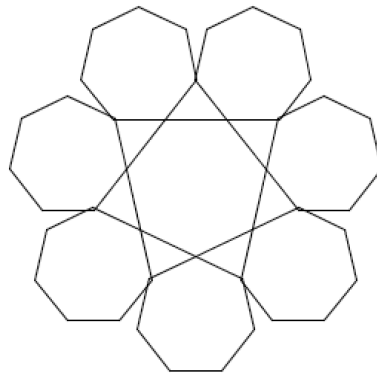
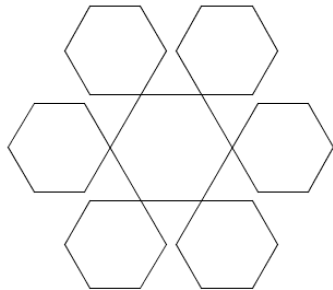
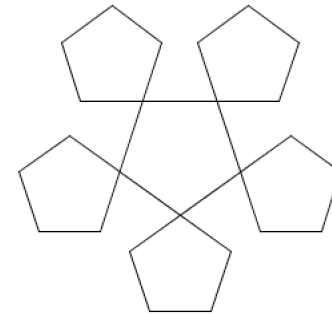
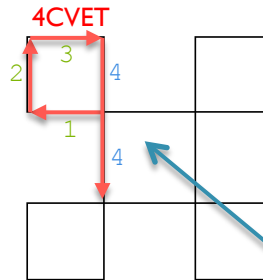
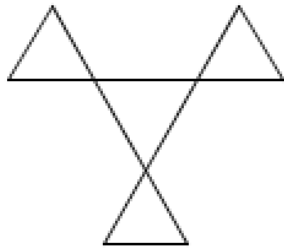
Vitraži z vzorcem



(vitraži so iz cerkve Marijinega rojstva v Cerknici)



nRože



nCVET:

$(n-1)$ -krat ponovimo:

nariši črto dolžine d ;

obrni za kot $360/n$;

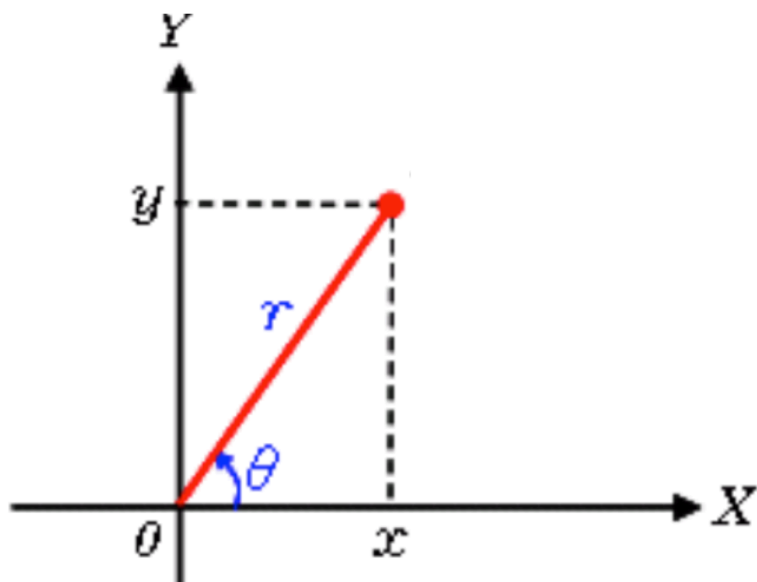
nariši črto dolžine $2 \cdot d$;

nROŽA:

n -krat narišemo nCVET



Pretvorba med polarnimi in kartezičnimi koordinatami



Zapis kota (deg / rad):

$$\text{kot}_d = 30^\circ$$

$$\text{kot}_r = 2\pi * \text{kot}_d / 360$$

Točka s polarnimi koordinatami (r, θ) ima kartezične koordinate

$$x = r * \cos(\theta)$$

$$y = r * \sin(\theta)$$

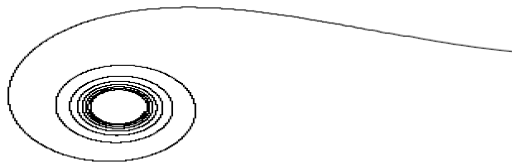
Točka s kartezičnimi koordinatami (x, y) ima polarne koordinate

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan(y/x)$$

Matematične funkcije (`Math.sin()`, `Math.cos()`, ...) kot parameter sprejmejo kot v radianih!

Spirala



Osnovna ideja:

- v zanki izrisujemo točko s polarnimi koordinatami (r, kot) ;
- v vsaki iteraciji zanke enakomerno povečamo kot in
- v vsaki iteraciji zanke spremenimo r in sicer (za različne spirale):

Lituus

```
r = 50 / Math.sqrt(kot);
```

Fermatova spirala

```
r = 10 * Math.sqrt(kot);
```

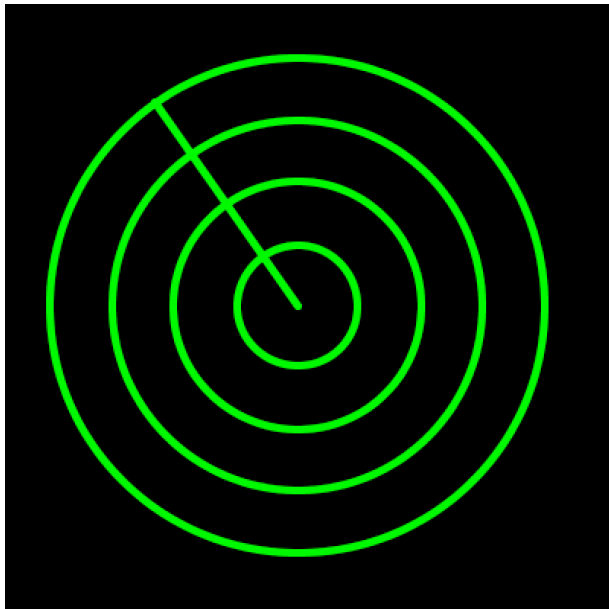
Arhimedova spirala

```
r = 1.1 * kot;
```

Hiperbolična spirala

```
r = 100/kot;
```

Radar (z vrtečim kazalcem)



- Za “animacijo” uporabimo metodo

```
StdDraw.show()
```

- Za prikaz kazalca napišemo metodo

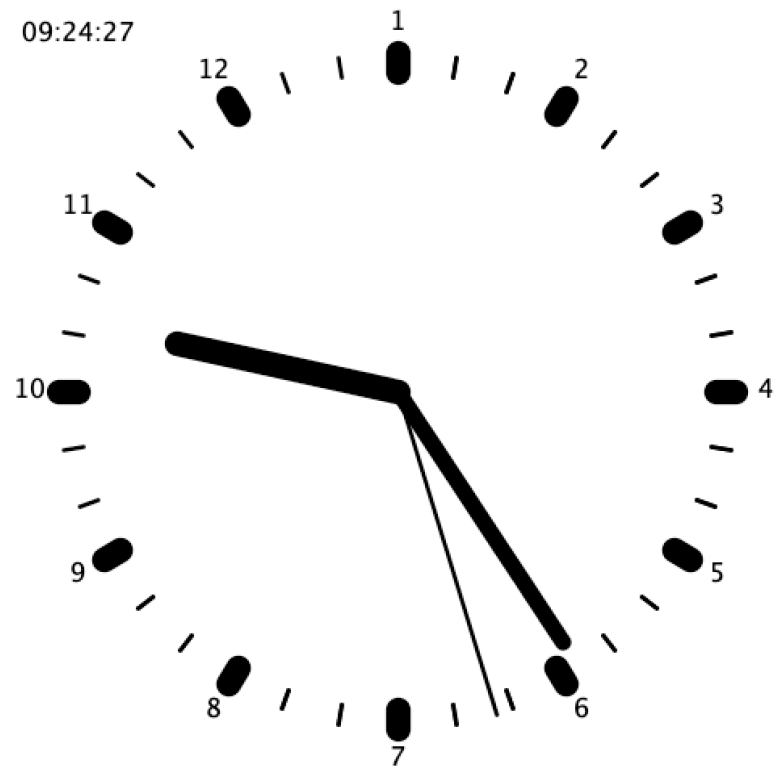
```
kazalec(int dolzina, double kot, double debelina) {...}
```

- Kazalec gre od $(0, 0)$ do (x, y) pri

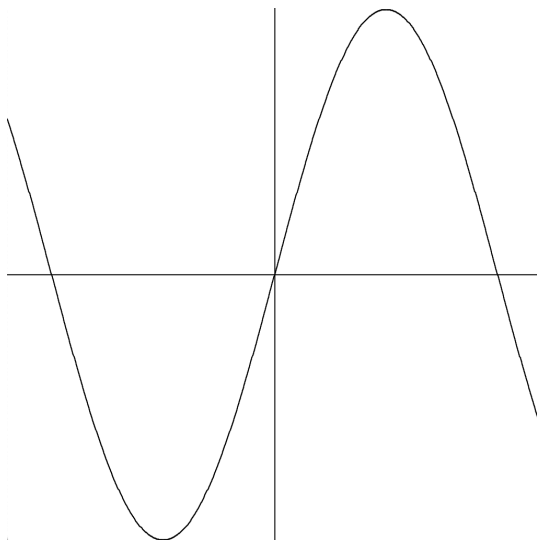
```
x = dolzina * Math.sin(2 * Math.PI * kot / 360);  
y = dolzina * Math.cos(2 * Math.PI * kot / 360);
```



Ura



Graf funkcije



- programu podamo:
 - definicijsko območje (x_1 , x_2)
 - zalogo vrednosti (y_1 , y_2)
 - graf, ki ga želimo izrisati
- rišemo na območje dimenzije $[0, W] \times [0, H]$

za vsak $i = 0, 1, 2, \dots, W - 1$:

$$x = \frac{x_2 - x_1}{W} * i + x_1$$



preslikava iz $[0, W]$ na $[x_1, x_2]$

$$y = \sin(x)$$

$$j = H * \frac{y - y_1}{y_2 - y_1}$$



preslikava iz $[y_1, y_2]$ na $[0, H]$

nariši_piko(i, j)

