Einführung in die mathematische Software
Übung 4

TECHNISCHE UNIVERSITÄT DARMSTADT

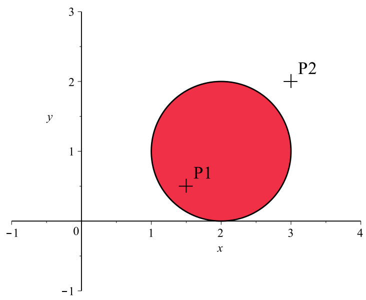
PD Dr. Ulf Lorenz
Dipl.-Math. Thomas Opfer

Wintersemester 2012/2013
Woche: 03.12.2012 - 07.12.2012

Aufgabe 1 Primzahlen ☆☆★

Generieren Sie mit Hilfe der `nextprime`-Methode eine 5-, eine 10- und eine 50-stellige Primzahl.

Aufgabe 2 Kreisscheibe ☆★★



Sie wollen für beliebig viele Punkte überprüfen können, ob sie in der (abgeschlossenen) Kreisscheibe mit Radius 1 um den Punkt $(2;1)$ liegen. Um diese Überprüfung nicht mehrfach implementieren zu müssen, soll eine Prozedur `PunktImKreis(x,y)` geschrieben werden, die überprüft, ob ein gegebener Punkt $(x;y)$ in dieser Kreisscheibe liegt. Die Prozedur soll `true` zurückgeben, falls der Punkt in der Kreisseibe liegt und `false`, falls nicht. Testen Sie ihre Prozedur u.a. mit den Punkten $P1 = \left(\frac{3}{2}; \frac{1}{2}\right)$ und $P2 = (3;2)$.

Hinweis: Punkte $(x;y)$, die in der Kreisscheibe liegen, erfüllen die Ungleichung $(x-2)^2 + (y-1)^2 \leq 1^2$.

Aufgabe 3 Prozeduren ☆★★

Schreiben Sie eine Prozedur `p(n)`, die "Zahl ist einstellig" zurückgibt, falls $0 \leq n \leq 9$, "Zahl ist zweistellig", falls $10 \leq n \leq 99$, "Zahl ist dreistellig", falls $100 \leq n \leq 999$, und ansonsten "Zahl hat mehr als drei Stellen". Beachten Sie, dass es hier um den Rückgabewert der Prozedur geht, so dass Sie die Prozedur im Prinzip wie eine mathematische Funktion verwenden können. Schauen sie sich ggf. das Schlüsselwort `return` noch einmal genauer an.

Im folgenden Beispiel soll also a den Wert "Zahl ist zweistellig" haben und es soll nach der ersten Zeile nichts ausgegeben werden:

```
a:=p(42):
print(a):
```

```
"Zahl ist zweistellig"
```

Aufgabe 4

Lösen Sie die Aufgaben von Übung 3, die Sie noch nicht gelöst haben.

> Ein Schüler findet zwanzig Euro und gibt sie seinem Mathelehrer zurück. Er sagt: „Gibt's dafür nicht 10% Finderlohn?“ Darauf der Lehrer: „Sei nicht so gierig, hier hast fünf Euro und nun Ruhe!“

Aufgabe 5 Die Mandelbrot-Menge **Intensivaufgabe**

Die Mandelbrot-Menge ist diejenige Teilmenge der komplexen Zahlen $c \in \mathbb{C}$, für die die Folge $z_0, z_1, z_2, \ldots$ mit $z_0 = 0$ und $z_{n+1} = z_n^2 + c$ beschränkt bleibt. Diese Menge soll im Folgenden visualisiert werden.

a) Erzeugen Sie mit dem `Create`-Befehl aus dem Paket `ImageTools` ein Bild img mit der Höhe 201, der Breite 301 und der Hintergrundfarbe Weiß.

b) Der Bildpunkt (x, y) soll die komplexe Zahl $\left(\frac{1}{100} \cdot x - \frac{201}{100}\right) + \left(\frac{-1}{100} \cdot y + \frac{101}{100}\right) \cdot I$ darstellen. (Dies dient lediglich der Skalierung, damit die Ecken des Bildes ganze Zahlen repräsentieren und die Koordinatenachsen jeweils durch eine Bildpunktreihe repräsentiert werden.) Schreiben Sie eine Funktion t, die diese Transformation durchführt und verifizieren Sie, dass $t(1,1) = -2 + I$, $t(301, 201) = 1 - I$ und $t(201, 101) = 0$.

c) Schreiben Sie eine Prozedur m in Abhängigkeit einer komplexen Zahl c, die überprüft, ob die oben genannte Folge beschränkt bleibt. Initialisieren Sie z_0 mit 0.0, um auf die numerische Berechnung umzuschalten. Führen Sie 50 Iterationen durch. Wenn der Betrag eines Folgengliedes größer als 50 wird, geben Sie 1 zurück, ansonsten 0.

d) Färben Sie alle Bildpunkte (x, y) nach der Vorschrift $img[y, x] := m(t(x, y))$.

(*Anmerkung:* 1 steht für weiß, 0 für schwarz. Bei Bildern wird die y-Koordinate zuerst angegeben.)

e) Schauen Sie sich das Bild img mittels `View` an.