

Die Kreiszahl π bestimmen

Rupert Bogensperger

12.11.2021

Inhalt

Aufgabe 1 (15 Punkte).....	3
Bearbeitung:	3
Pseudocode	3
Aufgabe 2 (5 Punkte).....	4
Bearbeitung:	4
Aufgabe 3 (5 Punkte).....	5
Bearbeitung:	5
Aufgabe 4 (40 Punkte).....	5
Bearbeitung:	5
Aufgabe 5 (10 Punkte).....	5
Bearbeitung:	5
Aufgabe 6 (10 Punkte).....	5
Bearbeitung:	6
Aufgabe 7 (15 Punkte).....	6
Bearbeitung:	6
Verweise	6

Aufgabe 1 (15 Punkte)

Projektmanagement: Recherchiere im Internet gängige Vorgehensmodelle zum Projektmanagement einer Softwareentwicklung. Überlege dir, welches der Softwareentwicklungsmodelle zu verwenden ist

und begründe diese Wahl. Wie passen die Durchführung des Projekts und dessen Phasen in das gewählte Modell? Dein Vorgehensmodell sollte zumindest eine Designphase (= Pseudocode), eine Implementierung (= Programmierungsarbeit) und eine Testphase umfassen.

Der Pseudocode kann handschriftlich oder in Word erstellt werden!

Bearbeitung:

- **Wasserfallmodell**
Das Wasserfallmodell ist ein lineares Vorgehensmodell, insbesondere für die Softwareentwicklung, das in Phasen organisiert wird. Dabei gehen die Phasenergebnisse wie bei einem Wasserfall immer als bindende Vorgaben für die nächsttiefere Phase ein
 - **V-Modell**
Das V-Modell ist ein Vorgehensmodell in der Softwareentwicklung, bei dem der Softwareentwicklungsprozess in Phasen organisiert wird. Neben diesen Entwicklungsphasen definiert das V-Modell auch das Vorgehen zur Qualitätssicherung phasenweise. Das V-Modell nach ISTQB-Syllabus habe ich in einem anderen Artikel dargestellt.
 - **Inkrementelles Modell**
Das inkrementelle Vorgehensmodell beschreibt einen Prozess der kontinuierlichen Verbesserung, welcher häufig in kleinen oder sogar kleinsten Schritten vollzogen wird. Das große Ganze ist bereits bekannt.
 - **Spiralmodell**
Das Spiralmodell ist ein Vorgehensmodell in der Softwareentwicklung, das im Jahr 1986 von Barry W. Boehm beschrieben wurde. Das Management kann immer wieder eingreifen, da man sich spiralförmig voran entwickelt.
- Evolutionäres Modell: Im Gegensatz zum Inkrementellen Modell ist beim Evolutionären Modell das große Ganze noch nicht bekannt.
- **Agile Softwareentwicklung**
Agile Softwareentwicklung ist der Oberbegriff für den Einsatz von Agilität in der Softwareentwicklung. Je nach Kontext bezieht sich der Begriff auf Teilbereiche der Softwareentwicklung – wie im Fall von Agile Modeling – oder auf den gesamten Softwareentwicklungsprozess – exemplarisch sei Extreme Programming angeführt. Agile Softwareentwicklung versucht mit geringem bürokratischen Aufwand und wenigen Regeln auszukommen. Ein Beispiel Framework für Agile Softwareentwicklung ist Scrum.

Quelle: (Bregenzer, 2012)

Ich entwickle gerne nach dem Wasserfallmodell und bin mit dessen Umgang vertraut. Weiteres denke ich, dass es für dieses Projekt gut geeignet ist, da das Projekt nicht komplex ist.

Pseudocode

Gui

Auf der GUI¹ gibt es ein Canvas Element zur grafischen Darstellung von Ergebnissen.

Es gibt 2 Eingabe Felder, eines für eine Anzahl an Punkten und ein anderes für eine Genauigkeit der Berechnung von Pi bzw. dessen nachkommerstellen. Wenn man eine Zahl in die Felder eintippt und zum Schluss auf <Enter> drückt, werden die werte im Programm gespeichert.

Es gibt einen Knopf für Start, der den Hauptprozess der Software startet.

Es gibt einen Knopf zum Beenden des Programmes.

Es gibt einen Knopf, der dazu dient, um die Daten auszuwerten bzw. im späten Ablauf des Programmes dazu dient die Graphische Darstellung der Zufallswerte für x oder y anzuzeigen.

Es gibt ein Textfeld, welches ein berechnetes Pi Ergebnis ausgibt.

Hauptprogramm

Schaue, ob eine Anzahl oder eine Genauigkeit angegeben wurde.

Fall Anzahl:

Generiere ,x' Punkte mit einem x und y wert.

Zeichne diese Punkte in ein Canvas zusammen mit einem Grid.

Erstelle zwei Gruppen an Punkten, ob sie innerhalb des Kreises oder innerhalb des Quadrates sind.

Berechne Pi anhand des Verhältnisses der Gruppen.

Fall Genauigkeit:

Generiere einen neuen Punkt und füge ihn einer Liste an Punkten bei

Erstelle zwei Gruppen an Punkten, ob sie innerhalb des Kreises oder innerhalb des Quadrates sind.

Berechne Pi anhand des Verhältnisses der Gruppen.

Wenn die Genauigkeit mit dem errechnenden Pi zusammenpasst, dann Zeichne diese Punkte in ein Canvas zusammen mit einem Grid und Beende den Vorgang.

Sonst wiederhole den Vorgang

Datenauswertung

Mache Listen für die Bereiche 0-1 für x und y im Intervall von 0.1

Sortiere die x und y Werte der Punkte in die Listen.

Gebe jeweils die längen der Listen für x oder y graphisch aus

Aufgabe 2 (5 Punkte)

Software Engineering: Erstelle das Fenster/die GUI und erstelle einen „Beenden“-Button

Bearbeitung:



¹ Graphical User Interface

Aufgabe 3 (5 Punkte)

Software Engineering: Erstelle die 2 Eingabefelder im Fenster die ausgelesen werden können.

Bearbeitung:



Aufgabe 4 (40 Punkte)

Software Engineering: Programmiere in Python die Methode zur Berechnung von π , wie in diesen Unterlagen angegeben. Erstelle für die Ausgabe ein Feld, in dem die Iterationen und der Wert für π angegeben werden können und implementiere dies so, dass nach jeder neuen Zufallszahl die Werte aktualisiert werden.

Nutze vorerst als Zufallszahlengenerator die Erweiterung random und wähle dafür einen ausreichend guten Seed-Wert. (Erkläre warum du diesen Seed-Wert verwendest).

Implementiere diese Funktion mithilfe des „Start“-Buttons wie in der Skizzen angegeben.

Am Ende der Simulation soll das Diagramm als PNG vom Programm im Ordner des Programms als „pi_Zeitstempel.png“ gespeichert werden, wobei das Füllwort Zeitstempel mit dem jeweiligen Zeitstempel befüllt werden soll.

Bearbeitung:



Problematik:

Bei dem Versuch sich an Pi zu nähern, benötigt man gute Zufallszahlen. Wenn man nicht gut an Pi rankommt, ist es schwer eine Funktion zu implementieren, welche es erlaubt sich an die „x.“ nachkommastelle von Pi zu nähern, bzw. Pi an diese Genauigkeit zu errechnen.

Auch bei der Verwendung eines Random Seed Wertes, basierend auf der Zeitepoche in NS, war es nicht möglich Zufallszahlen zu generieren die gut genug waren.

Aufgabe 5 (10 Punkte)

Software Engineering: Werte die aus Aufgabe 4 erhaltenen Datenpunkte anhand ihrer x und y Koordinaten mithilfe eines Histogramms der Koordinaten aus und zeige diese Diagramme mittels Knopfdruck

auf den Button Daten auswerten nochmals in einem neuen Fenster.

Auch diese beiden Histogramme sollten als „Histogramm_x/y_Zeitstempel.png“ gespeichert werden.

Bearbeitung:



Aufgabe 6 (10 Punkte)

Software Engineering: Anstelle des Zufallszahlengenerators des Pakets random, schreibe eine eigene Methode die mithilfe des linearen Kongruenzgenerators die Zufallszahlen erzeugt.

Vergleiche die Ergebnisse für verschiedene Parameter des Kongruenzgenerators mit denen aus der

Python-Erweiterung random und diskutiere diese fachlich.

Bearbeitung:

Linearer Kongruenzgenerator

Formel: $x_{a+1} = ((a * x_a) + c) \text{ modulo } m$.

Generiert zufällige ganze Zahlen von 0 bis einschließlich $m-1$.

Seed, Multiplikator, Inkrement und Modul wirken sich auf die Ausgabe des LCG aus.

der Startwert $X_0 \geq 0$, der Multiplikator ' a ' ≥ 0 , das Inkrement ' c ' ≥ 0 , der Modul ' m ' $> X_0$, ' a ', ' c '

Die Anzahl der möglichen Werte ist m (einschließlich 0), daher beträgt die maximale Zykluszeit m und die maximale Anzahl von Läufen, bevor der Algorithmus wiederholt wird, beträgt ebenfalls m .

Bei $c = 0$ wird der Zufallszahlengenerator als multiplikatives Kongruenzverfahren bezeichnet.

Quelle: (ANTHONYC, 2019)

Aufgabe 7 (15 Punkte)

Projektmanagement: Dokumentiere alle Schritte genau und bringe das Protokoll in eine saubere Form.

(Genaueres zur Abgabe folgt auf der nächsten Seite)

Bearbeitung:

Verweise

ANTHONYC. (28. 10 2019). *codeing-engineer*. Abgerufen am 12. 11 2021 von codeing-engineer:
<https://coding-engineer.com/2019/10/28/random-number-generator-python/>

Bregenzer, S. (17. Jänner 2012). *milsystems*. Abgerufen am 12. 11 2021 von
<https://blog.milsystems.de/2012/01/softwareentwicklungsmodelle-nach-istqb/>