

Beschreibung der Idee zum Terminal Draw App

Dokument-Daten			
Schule	CsBe Bern		
Klasse	Bis18/P		
Autor	Michal Durik		
E-Mail	Mdurik2@gmail.com		
Ausgabedatum	21.03.2020		
Version	V1.0		
Status	<input type="checkbox"/> In Arbeit	<input type="checkbox"/> In Prüfung	<input checked="" type="checkbox"/> Genehmigt, zur Nutzung
Klassifizierung	<input type="checkbox"/> Vertraulich	<input checked="" type="checkbox"/> Intern	<input type="checkbox"/> Zur Veröffentlichung

Tabelle 1 – Dokument-Daten

1. Änderungsprotokoll

Version	Datum	Name	Beschreibung
V0.1	21.03.2020	Michal Durik	Vorlage mit Inhalt erstellen

Tabelle 2 - Änderungsprotokoll

2. Inhaltsverzeichnis

BESCHREIBUNG DER IDEE ZUM	1
LOCATION-SHARING-APP IM REACT JS.....	ERROR! BOOKMARK NOT DEFINED.
1. ÄNDERUNGSPROTOKOLL	2
2. INHALTSVERZEICHNIS.....	2
3. EINLEITUNG	3
4. FUNKTIONELLE ANFORDERUNGEN	3
5. TECHNOLOGIEN.....	4
6. MOCK-UP	4
7. CLEAN CODE.....	6
8. PROJECT STRUKTUR	7
8.1 BESCHREIBUNG	7
9. ABBILDUNGSVERZEICHNIS.....	8
10. TABellenverzeichnis	8

3. Einleitung

Definieren Sie, was sauberer Code ist, und beschreiben Sie seine Vor- und Nachteile. Definieren Sie dann den Fortschritt der Anwendungserstellung und erstellen Sie schließlich die Anwendung.

4. Funktionelle Anforderungen

*** The Problem ***

GOAL: CLEAN CODE !!

Description

You're given the task of writing a simple console version of a drawing program.
At this time, the functionality of the program is quite limited but this might change in the future.

In a nutshell, the program should work as follows:

1. Create a new canvas
2. Start drawing on the canvas by issuing various commands
3. Quit

Command	Description
C w h	Should create a new canvas of width w and height h.
L x1 y1 x2 y2	Should create a new line from (x1,y1) to (x2,y2). Currently only horizontal or vertical lines are supported. Horizontal and vertical lines will be drawn using the 'x' character.
R x1 y1 x2 y2	Should create a new rectangle, whose upper left corner is (x1,y1) and lower right corner is (x2,y2). Horizontal and vertical lines will be drawn using the 'x' character.
B x y c	Should fill the entire area connected to (x,y) with "colour" c. The behaviour of this is the same as that of the "bucket fill" tool in paint programs.
Q	Should quit the program.

Sample I/O

Below is a sample run of the program. User input is prefixed with enter command:

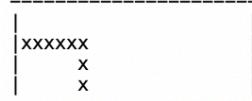
enter command: C 20 4



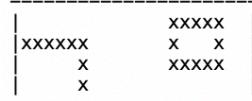
enter command: L 1 2 6 2



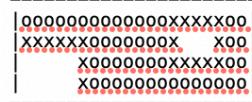
enter command: L 6 3 6 4



enter command: R 14 1 18 3



enter command: B 10 3 o



enter command: Q

5. Technologien

Für mein Projekt habe ich mich für Python 3.6.5 mit virtueller Umgebung entschieden, was in der Python-Welt ein Standard ist, da es sich nur um eine Konsolenanwendung handelt.

6. Mock-Up

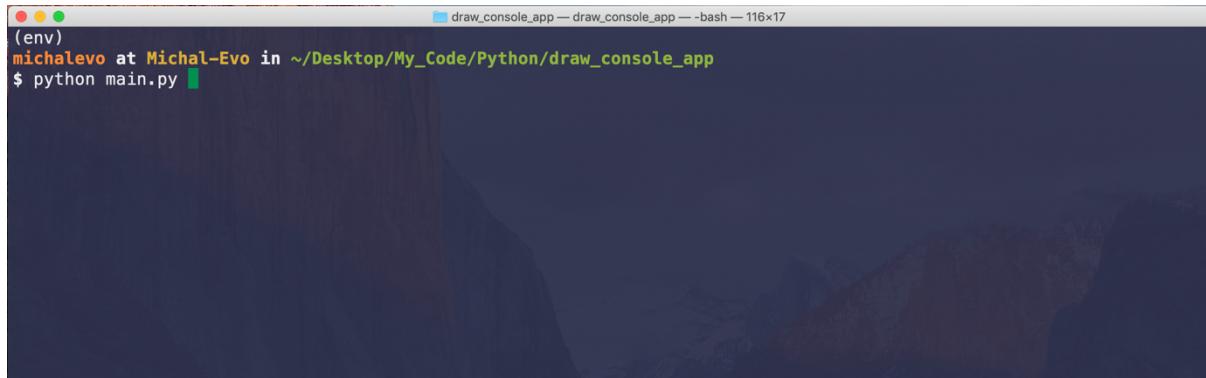


Abbildung 1 - Start app



Abbildung 2 - Menu app



Abbildung 3 - Create canvas

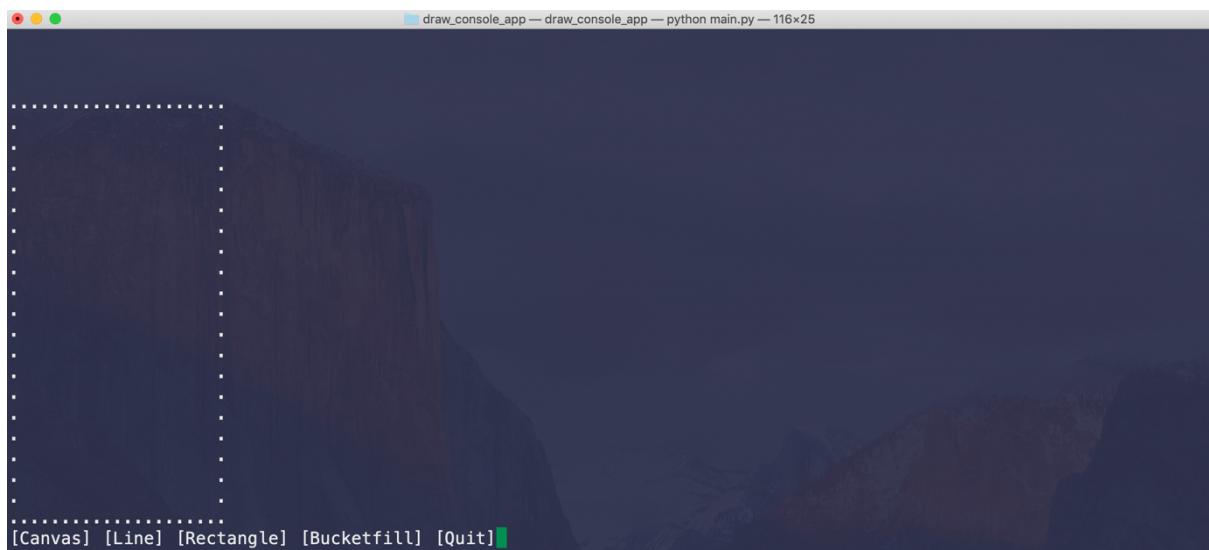


Abbildung 4 – Canvas

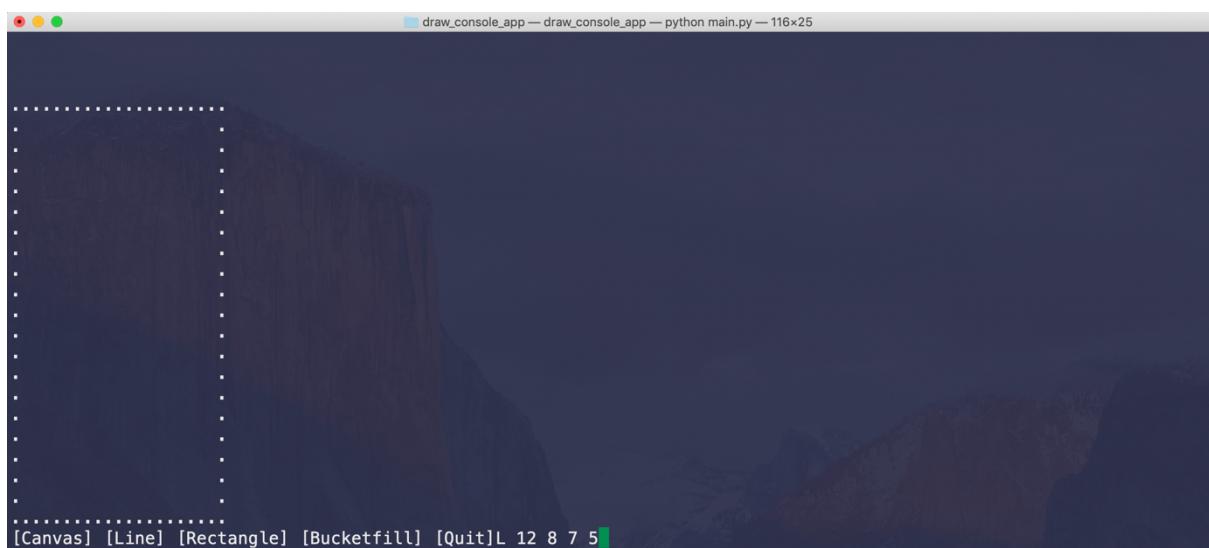


Abbildung 5 - Create Line

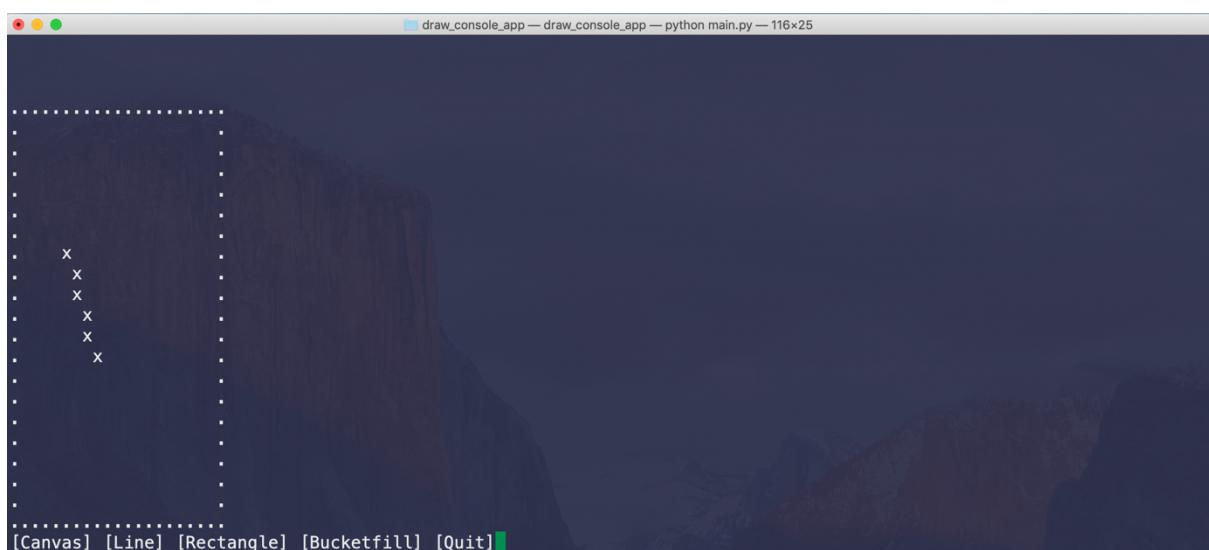


Abbildung 6 - Created Line

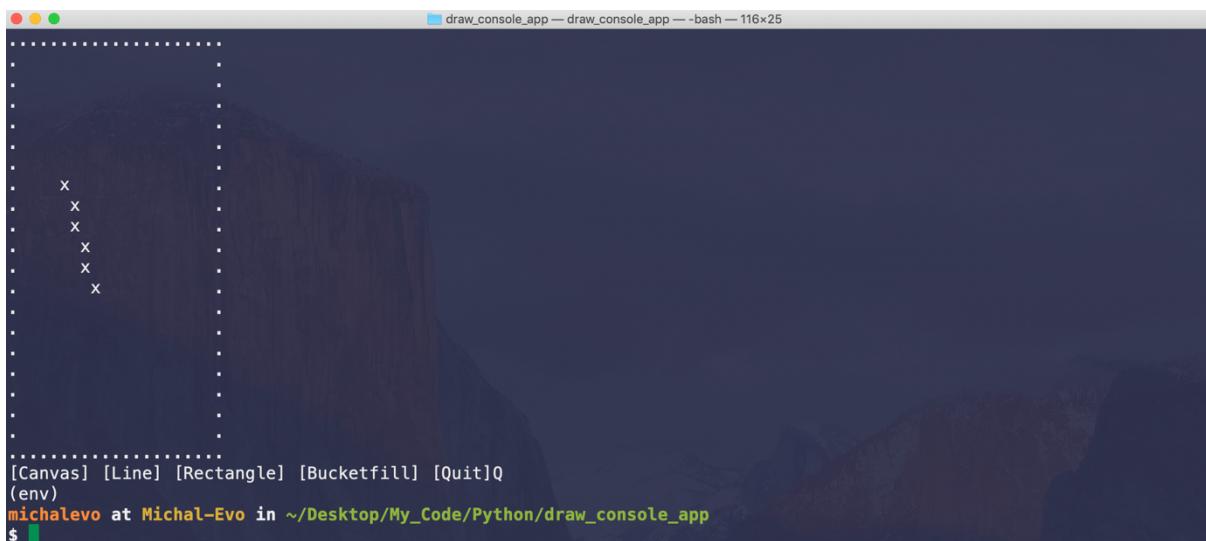


Abbildung 7 - Quit app

7. Clean Code

Clean Code ist ein Begriff aus der Softwaretechnik, der seinen Ursprung im gleichnamigen Buch von Robert Cecil Martin hat. Als „sauber“ bezeichnen Softwareentwickler in erster Linie Quellcode, aber auch Dokumente, Konzepte, Regeln und Verfahren, die intuitiv verständlich sind. Als intuitiv verständlich gilt alles, was mit wenig Aufwand und in kurzer Zeit richtig verstanden werden kann. Vorteile von Clean Code sind stabilere und effizient wartbarere Programme, d. h. kürzere Entwicklungszeiten bei Funktionserweiterung und Fehlerbehebungen. Die Bedeutung wächst mit der Beobachtung, dass im Schnitt 80 % der Lebensdauer einer Software auf den Wartungszeitraum entfällt.

Schwierigkeiten beim Entwickeln von Clean Code liegen:

- häufig in zunächst unklaren oder sich widersprechenden Anforderungen,
- zum Teil begründet im Fehlen von Erfahrung im Entwickeln von Clean Code,
- im Mangel an Disziplin beim Programmieren und
- im Aufwand nachträglicher Quellcode-Bereinigungen (dem sog. Refactoring).

Die Notwendigkeit, Code noch nach der Entwicklung von „unsauberer“ Stellen zu reinigen, wird häufig nicht gesehen oder vom Management nicht bewilligt, sobald das Programm seine vorgesehene Funktion ausübt. Ein direktes Schreiben von „sauberem“ Code ist nahezu unmöglich, kann jedoch durch den bewussten Umgang mit den Prinzipien und Praktiken von Clean Code verbessert werden.

Darstellungen wie die in “Clean Code” von Robert C. Martin oder “The Art of Readable Code” von Dustin Boswell haben ihren Wert – doch letztlich greifen Sie unserer Ansicht nach zu kurz. Sie reduzieren ein tiefgreifendes Problem der Softwareentwicklung auf vor allem ein Artefakt: die textuelle Form einer Lösung, Code.

Clean Code Development ist notwendigerweise agile Softwareentwicklung. Agilität reduziert auf das Wesentliche ist:

- inkrementell
- lernend
- reaktiv

In diesem Rahmen soll aber nicht nur funktionaler und effizienter Code entstehen, sondern eben auch wandelbarer und korrekter. Clean Code Development ist für uns ohne eine Systematik, wie er inkrementell und eng am Kunden hergestellt werden kann, nicht denkbar.

8. Project Struktur

```
> __pycache__
> .vscode
> env
↳ .gitignore
🐍 helpers.py
🐍 main.py
🐍 methods.py
```

Abbildung 8 - Project Struktur

8.1 Beschreibung

Zuerst habe ich die ganze Aufgabe dreimal hintereinander gelesen. Dann habe ich die Programmiersprache definiert, in meinem Fall Python, weil ich es am besten kenne.

Ich habe ein Python-Projekt erstellt, sodass ich zuerst eine virtuelle Python-Umgebung erstellt habe, die die Grundlage für jedes Python-Projekt bildet. Dann habe ich main.py erstellt, wo ich eine Zeichenfläche basierend auf meinem alten Code erstellt habe. Ich musste es ein wenig anpassen. Alles, was ich geschrieben habe, habe ich prozedural als eine Funktion in einer Datei.

Nach dem Erstellen der Zeichenfläche habe ich ein Menü für die Verwendung des Benutzers erstellt. Außerdem habe ich einen Handler für Eingaben erstellt, die der Benutzer eingibt. Nur Canvas-Eingaben funktionieren und ich habe keine Zeit, die anderen zu definieren.

Dann habe ich die Erstellung einer Leinwand hinzugefügt, die auf der benutzerdefinierten Höhe und Breite sowie dem Rand mit Punkten basiert.

Ich habe auch eine benutzerdefinierte Linie basierend auf 2D-Koordinaten gemäß Mathematik x_1 , y_1 , x_2 , y_2 erstellt. Einfache Benutzereingaben funktionierten wie sie sollten, aber als zwei Einheiten zusammen erscheinen sollten, fielen sie auseinander. Deshalb musste ich lange im Internet Googeln, um eine Lösung und einen Code zu finden, die ich für meine Bedürfnisse modifizierte. Ich habe auch eine Exit-Anwendung erstellt.

Als ich diese Basis bereits hatte, teilte ich die gesamte Methode in mehrere auf, um sie transparenter zu machen und natürlich Redundanz zu vermeiden und die Grundlagen von sauberem Code einzuhalten.

9. Abbildungsverzeichnis

Abbildung 1 - Start app	4
Abbildung 2 - Menu app.....	4
Abbildung 3 - Create canvas.....	4
Abbildung 4 – Canvas	5
Abbildung 5 - Create Line	5
Abbildung 6 - Created Line	5
Abbildung 7 - Quit app	6
Abbildung 8 - Project Struktur.....	7

10. Tabellenverzeichnis

Tabelle 1 – Dokument-Daten	1
Tabelle 2 - Änderungsprotokoll.....	2