

Console Spreadsheet

Cel i opis projektu

Projekt polegał na zaimplementowaniu w pythonie konsolowego arkusza kalkulacyjnego. Program w założeniach miał umożliwiać:

- Uruchomienie/utworzenie arkusza z pliku wraz z podpoleceniami,
- Wprowadzanie do komórek wartości tekstowych i liczbowych, oraz funkcji wyliczanych na podstawie wartości z innych komórek i ich zakresów,
- Adresowanie komórek identyczne jak w excelu,
- Wyświetlanie i bieżące odświeżanie wartości komórek,
- Zapis i odczyt arkusza.

Opis podziału programu na klasy

Address

Klasa opisująca obiekt adresu, umożliwiająca:

- rozbicie adresu na literę i liczbę,
- przesunięcie adresu o wektor liczbowy w zadanym obszarze ograniczającym

RangeAddress

Klasa opisująca obiekt zakresu adresów, umożliwiająca:

- podanie wszystkich adresów wchodzących w skład zakresu,
- określenie ilości wierszy i kolumn wchodzących w skład zakresu
- robienie zakresu na litery i liczby wchodzące w skład adresów zakresu

Cell

Klasa opisująca obiekt komórki arkusza, umożliwiająca:

- podanie wartości komórki lub wpisanego weń polecenia

Spreadsheet

Klasa opisująca obiekt arkusza, umożliwiająca:

- dodanie/usunięcie komórek do arkusza
- podanie/ustawienie wartości komórki
- podanie opisu każdej komórki (adres, wartość, "wartość tekstowa")

SpreadsheetIO

Klasa opisująca obiekt menadżera do operacji IO na arkuszu, umożliwiająca:

- zapisanie arkusza do pliku
- odczytanie arkusza z pliku
- utworzenie arkusza o zadanych wymiarach i lokalizacji

SpreadsheetView

Klasa opisująca obiekt widoku arkusza, udostępniająca:

- widok arkusza w terminalu
- wprowadzanie i edytowanie zawartości komórek
- poruszanie się po arkuszu przy pomocy kursora
- skalowanie wielkości pokazywanego arkusza do wielkości konsoli

CommandInterpreter

Klasa opisująca obiekt interpretera poleceń, umożliwiającą:

- obliczenie wartości nawiasowanego wyrażenia matematycznego przy pomocy ONP
- obliczenie wartości funkcji korzystających z wartości innych komórek
- przypisanie do podanej komórki wartości wyrażenia, liczby lub tekstu
- odświeżenie wartości w arkuszu, po przez ponowne obliczenie zawartości komórek z komendami

Errors.py

Skrypt posiadający definicje klas opisujących możliwe wyjątki przy tworzeniu adresów, wprowadzaniu komend, odczycie i zapisu arkusza.

Instrukcja użytkownika

Uruchomienie programu i podpolecenia

- **python3 spreadsheet.py**

Program pokaże domyślny pusty arkusz, który przy zapisie zostanie utworzony pod nazwą "spreadsheet.csv" w miejscu uruchomienia programu.

- **python3 spreadsheet.py -l [ścieżka do pliku]**

Program załaduje arkusz dostępny pod zadaniem adresem, przy zapisie arkusza, będzie to również plik zapisu.

- **python3 spreadsheet.py -n [ścieżka do pliku] [rozmiar arkusza X:Y]**

Program utworzy i otworzy arkusz we wskazanej lokalizacji o rozmiarach podanych w formacie XxY lub X:Y, gdzie X,Y to ilość wierszy, kolumn.

Klawiszologia

Nawigacja kursora : **Strzałki**

Edycja zawartości komórki : **i**

Zapis stanu arkusza : **s**

Wyjście z programu : **q**

Wprowadzanie zawartości

Wprowadzanie tekstu i liczb

Tak jak w excelu, wystarczy wprowadzić wartość, klikając uprzednio „i”, po edycji potwierdzając enterem.

Wprowadzanie komend

Struktura komendy:

nazwa(początek zakresu komórek : koniec zakresu komórek)

Obsługiwane komendy: min, max, avg, sum

np: sum(A1:C3)

Wprowadzanie wyrażenia

Tak jak w excelu należy rozpocząć od "="

np: =sum(A1:C3)+12-A2(C1/(4+3))*

Część refleksyjna

Zakres wykonanych prac

- wyświetlanie arkusza podobnego wyglądem do excela,
- nawigacja po arkuszu, z przesuwaniem wyświetlanego zakresu arkusza,
- responsywna zmiana rozmiaru arkusza wraz z zmianą rozmiaru terminala,
- zaprojektowanie optymalnego systemu przechowywania zawartości arkusza,
- zaimplementowanie parsersa wyrażeń matematycznych z dostępem do zawartości innych komórek,
- odświeżanie widoku arkusza i wartości komórek zależnych przy zmianie wartości,
- system adresowania i konwersji na liczbowe koordynaty,
- zapis arkusza do pliku, z określeniem lokalizacji, rozmiaru i zawartości arkusza,
- odtworzenie arkusza z pliku,

Możliwe dalsze modyfikacje

- sprawdzanie rekursji w wprowadzonych poleceniach,
- możliwość ruszania kursorem w polu edycji,
- zakres komórek może być listą, a nie prostokątnym zakresem,

Głównym powodem niezaimplementowania powyższych punktów był ograniczony zasób czasowy, a także możliwe(głównie przy sprawdzaniu rekursji) obniżenie prędkości działania programu

Napotkane trudności

Największy problem sprawiło mi napisanie systemu przesuwania widoku arkusza, kiedy kursor przechodził do niewidocznych komórek, a także zmiana rozmiaru arkusza przy zmianie rozmiaru terminala.

Podsumowanie

Uważam, że mój projekt zasługuje na wysoką ocenę, ponieważ w pełni spełnia podane wymagania funkcjonalne projektu, a także posiada parę dodatkowych funkcjonalności, jak np. dość rozbudowany system wyświetlania i poruszania się po arkuszu. Ponadto starałem się minimalizować złożoność obliczeniową, np. przy ruszaniu kursora przerysowywane są tylko dwie komórki, w słowniku arkusza znajdują się tylko komórki z danymi, a przeliczane ponownie są tylko komórki zawierające wyrażenie. Sam pomysł projektu był dla mnie interesujący i przyjemnie mi się nad nim pracowało.