

Jakub Lemański projekt: Arkusz kalkulacyjny

Celem projektu jest wykonanie arkusza kalkulacyjnego którego będzie można wyświetlić w konsoli tekstowej. Mój pomysł na rozwiązanie tego zadania to stworzenie klasy „Spreadsheet” która będzie odpowiedzialna za stworzenie oraz wypisanie arkusza do konsoli. Program można uruchomić przy pomocy dwóch modułów: Main.py i oraz Main2.py. Różnicą stanowią pod polecenia „(subcommands)” przyjmujące dodatkowe argumenty wywołania, znajdujące się w module Main2.py.

Opis klasy „Spreadsheet”:

Konstruktor:

Konstruktor klasy przyjmuje cztery parametry. Pierwszym „number_of_columns” jest to liczba kolumn które zostaną wyświetlone w konsoli tekstowej(jako domyślną wartość została uznana liczba 15 ponieważ jest to za razem najbardziej optymalna jak i wystarczająca liczba). Drugim parametrem jest „number_of_rows” ten parametr odpowiada za liczbę wierszy arkusza(tak samo jak w przypadku kolumn domyślną wartością jest liczba 15). Trzecim parametrem jest „constants_dict” jest to słownik w którym przechowywane są stałe liczbowe i tekstowe. Czwartym a zarazem ostatnim parametrem jest „function_dict” jest to słownik odpowiedzialny za przechowywanie funkcji przypisanej do danej komórki oraz komórek na podstawie której ta funkcja jest przypisana. W tym słowniku do komórki przypisana jest lista której pierwsza pozycja jest strigniem oznaczającym jaka funkcja jest przypisana do tej komórki a na drugiej pozycji jest lista zawierająca komórki na podstawie której funkcja jest wyliczana.

Metody:

1. __str__ Ta metoda jest odpowiedzialna za utworzenie arkusza. Na początku przy pomocy funkcji „generate_dict” z modułu „table_dict_generator.py” jest generowany słownik który umożliwia nazwanie poszczególnych kolumn, dzięki tej funkcji możemy generować takie nazwy jak aa, ab itp.. Następnie przechodzimy do generowania arkusza, jeżeli dana komórka jest w słowniku stałych wtedy jej wartość zostaje wpisana w odpowiadającej jej komórce.

2.New_value Dzięki tej metodzie możemy przypisać danej komórce stałą tekstową lub liczbową. Musimy podać tej metodzie dwa parametry. Pierwszym jest komórka (np.”A3”) drugim jest wartość którą chcemy przypisać

3.Następne siedem metod są to metody które umożliwiają nam przypisanie do komórki, podanej jako parametr, funkcji odpowiednio add – dodawania, subtract – odejmowania , min- najmniejsza wartość, max- największą wartość, average- średnia, division- dzielenia , multiplication - mnożenia. Po użyciu jednej z funkcji w słowniku funkcji do odpowiedniej komórki zostaje zapisana dana funkcja.

4.Kolejną metodą jest „assing_cells_to_function” jak wspomniałem wyżej w słowniku „function_dict” mamy zapisaną nie tylko funkcje komórki ale też komórki na podstawie której ta funkcja ma być wyliczana. Ta metoda pozwala nam przypisać listę komórek podrzędnych do komórki funkcji.

5.Metoda „deactivate_function” jest przeciwieństwem metod w podpunkcie 3 i umożliwia ona dezaktywowanie funkcji przypisanej do komórki.

6.Dwie kolejne metody odpowiadają za zapisanie oraz wczytanie pliku w formacie json. Zdecydowałem się na ten format ponieważ idealnie pasuje do struktury która chcemy przechować

7. Siedem kolejnych metod jest odpowiedzialne za działania zawarte w 3 podpunkcie. Pięć z nich za parametry przyjmuje „cell_of_value” czyli komórkę której ma zostać przypisana wartość oraz „cells_list” listę komórek na podstawie których ma zostać wykonana funkcja. Dwie pozostałe metody są inne ponieważ są to operacje odejmowania oraz dzielenia czyli działania do których w przeciwieństwie do pięciu poprzednich nie możemy podać większej ilości liczb niż 2. Dlatego też metoda „subtract” przyjmuje 3 parametry, kolejno: komórkę do której przypisujemy wartość, odjemna i odjemnik. W metodzie „division” podobnie jak w odejmowaniu musimy podać trzy parametry kolejno: komórkę do której przypisujemy wartość, dzielna, dzielnik

8. Kolejną metodą jest „cell_calculate” , jest to niezwykle ważna metoda ponieważ to ona odpowiada za wywołanie metody z podpunktu 3 w zależności od słownika funkcyjnego. Do tego umożliwia nam przeliczanie wartości zależnych przy jakiegokolwiek zmianie w naszym arkuszu.

9. Przedostatnia metoda „new_dependent_cell” jest dosyć podobna do metody z 4 podpunktu jednak tam mieliśmy możliwość przypisania tylko listy zawierające komórki a tutaj możemy przypisać pojedynczą komórkę.

10. Ostatnią metodą „remove_dependent_cell” umożliwia ona usunięcie komórki przypisanej do jakiejś komórki funkcyjnej

Instrukcja obsługi programu:

Main2.py

Do dyspozycji mamy 13 argumentów z których 6 przyjmuje dodatkowe argumenty wywołania. Aby poprawnie uruchomić program należy wprowadzić jeden z dwóch argumentów: „-file” oraz podać nazwę pliku aby wyświetlić arkusz zapisany w pliku lub „-empty_spreadsheet” który przyjmuje wartość prawdy lub fałszu aby wyświetlić i działać na pustym arkuszu. Nie należy podawać obu argumentów na raz ponieważ zostanie wyświetlony pusty arkusz. Po wprowadzeniu jednej z dwóch powyższych komend możemy zacząć działać na naszym arkuszu przy pomocy pozostałych 11 argumentów.

-„size” Do tego musimy podać dwa argumenty kolejno rows i columns które pozwalają ustawić ilość wierszy i kolumn.

-„-dont_save” W przypadku wczytania z arkusza z pliku wszystkie zmiany są domyślnie zapisywane aby temu zapobiec, trzeba użyć argumentu „dont_save”

-„-save” W przypadku wybrania opcji pustego arkusza aby go zapisać należy użyć „-save” oraz podać nazwę jak chcemy zapisać plik.

-„new_value” Ten argument pozwala nam przypisać stałą wartość liczbową lub tekstową do danej komórki. Aby poprawnie podać argumenty należy najpierw podać komórkę a następnie wartość.

-„function_set” Ten argument pozwala nam przypisać komórce jedną z 7 funkcji. Aby poprawnie skorzystać z tej funkcji po wpisaniu „function_set” należy wybrać jedną z 7 funkcji i podać nazwę komórki

-„assign_cell” Dzięki temu możemy przypisać komórki do komórki funkcyjnej na podstawie których funkcja będzie wyliczana. Argumentem obowiązkowym jest komórka funkcyjna do której chcemy przypisać komórki. Możemy podać jedną komórkę używając single_cell, lub kilka komórek oddzielonych przecinkiem za pomocą cells_list.

-„remove_cell” Jest przeciwieństwem powyższej funkcji. Umożliwia nam usunięcie komórki z listy komórki funkcyjnej dzięki czemu funkcja nie będzie wyliczana na podstawie usuniętej komórki . Należy podać komórkę funkcyjną oraz komórkę którą chcemy wyrzucić.

-„show_assigned_cell” Dzięki temu możemy zobaczyć w konsoli jakie komórki są przypisane do podanej przez nas komórki funkcyjnej.

-„-deactivate_function” Umożliwia nam dezaktywowanie funkcyjnej komórki podanej przez nas.

-„-remove_value” Pozwala usunąć przypisaną stałą wartość z komórki podanej przez nas.

Uwagi:

Należy pamiętać aby komórki nazywać dużą literą np. A3

Argumentów przyjmujących dodatkowe argumenty wywołania należy używać pojedynczo.