Zadanie domowe

1 Uwagi

- Opis zadania powinien być przepisany do Jupyter Notebook z użyciem odpowiedniej składni języka znaczników Markdown, a wyrażenia matematyczne powinny być zapisane za pomocą składni LaTeXa.
- Każda funkcja powinna mieć string dokumentujący jej działanie (tzw. Docstring)! Przykłady docstringów można znaleźć w artykule na stronie DataCamp. Czytelność docstringów jest też bardzo ważna!
- Złą praktyką jest pisać baaardzo długie linie kodu. Istnieje kilka sposób na podzielenie długiej linii na kilka linii przy zachowaniu odpowiedniego działania kodu.
- W języku Python bardzo często można obejść się bez pętli. W zadaniu domowym można było na przykład wykorzystać funkcję *numpy.sum()* oraz odpowiedni wycinek zmiennej *plansza* aby obliczyć liczę zamieszkanych sąsiadów dla pewnego elementu w macierzy.
- Osiem warunków *if*, które obsługują krawędzie planszy, można zastąpić odpowiednim zakresem indeksów.
- Produktowanie list jest bardziej efektywne niż zwykłe pętle *for*!
- Ważną zasadą dla osób dopiero zaczynających przygodę z programowaniem powinno być: "Nie od razu siadamy do komputera i programujemy. Na początku warto dokładnie przemyśleć sposób implementacji zadania a później znaleźć odpowiednie metody/ narzędzia, np. w języku Python i dopiero wtedy zacząć programować."
- Programy pisane w języku Python powinny być tak krótkie jak to tylko możliwe! Python pozwala na ogromną redukcję linii kodu w stosunku do innych języków programowania. Stosy bloków *if* ... *else* ... można zastąpić znacznie szybszymi liniami kodu, które nie tylko będą krótsze ale także pozwolą komuś kto czyta Państwa kody szybko je zrozumieć.
- Nazwy zmiennych powinny odzwierciedlać znaczenie zmiennej w kodzie (oczywiście wszystko w granicach rozsądku). Czytelność kodu!
- Funkcja *nowa_plansza* powinna także obsługiwać przypadek kiedy użytkownik nie wpisze zmiennej *pola* (*pola* == None).
- Gdy używamy wyniku *sąsiedzi(plansza, i, j)* to najlepiej przypisać go do pewnej zmiennej tak aby nie wywoływać tej samej funkcji z tymi samymi parametrami kilkukrotnie.
- Obiekty typu *numpy.array()* indeksuje się podobnie jak listy. Czyli np. indeks -1 będzie pokazywał na ostatni element w obiekcie jednowymiarowym.
- Pętle *for* i *while* mają różne przeznaczenia. Pętle *for* używamy wtedy kiedy wiemy ile iteracji ma wykonać nasz program, natomiast pętle *while* wtedy kiedy liczba iteracji jest nam z góry nie znana.
- Dobra znajomość pakietu NumPy pozwoli zredukować liczbę pętli oraz zoptymalizować funkcje potrzebne w zadaniu.
- Komentarz w Pyhone piszemy za pomocą znaku # na początku linii. Komentarz pod nazwą funkcji nie pełni roli Docstringa!
- Gdy pewne części kodu powtarzają się wtedy można stworzyć dodatkową funkcję, którą można wywoływać w podobnych miejscach, zamiast powtarzać zbędny kod.
- Wykonanie instrukcji *pass* nie powoduje żadnych skutków. Instrukcja ta może być używana wtedy kiedy wymagana jest obecność jakiejś instrukcji ale nie jest potrzebne wykonane żadnego kodu.