W zadaniu 3 mieliśmy stworzyć własną symulację z wykorzystaniem kolejki. Jako symulację wybrałyśmy sytuację polegającą na określeniu prędkości wpuszczania ludzi do auli na koncert   
w zależności od tego czy ludzie byli wpuszczani pojedynczo, czy bilety można było zakupić grupowo.

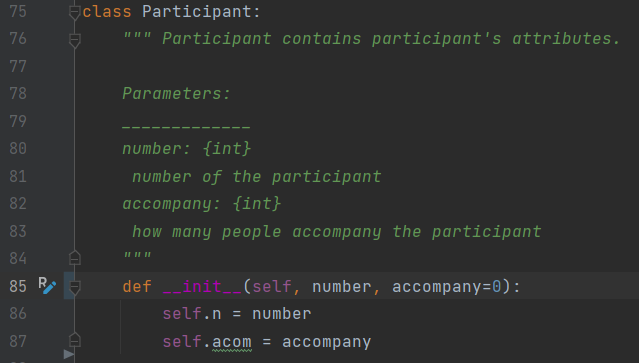
Uczestnicy koncertu otrzymują różne atrybuty.

Symulacje przeprowadzamy w różnych wariantach:

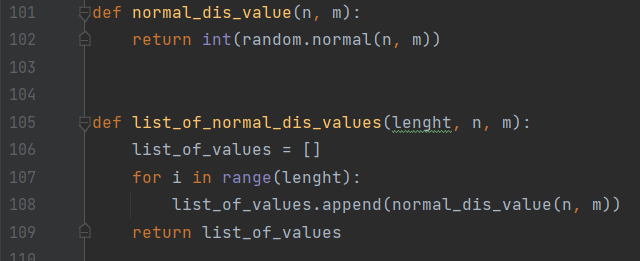
-gdy można zakupić bilety tylko pojedynczo;

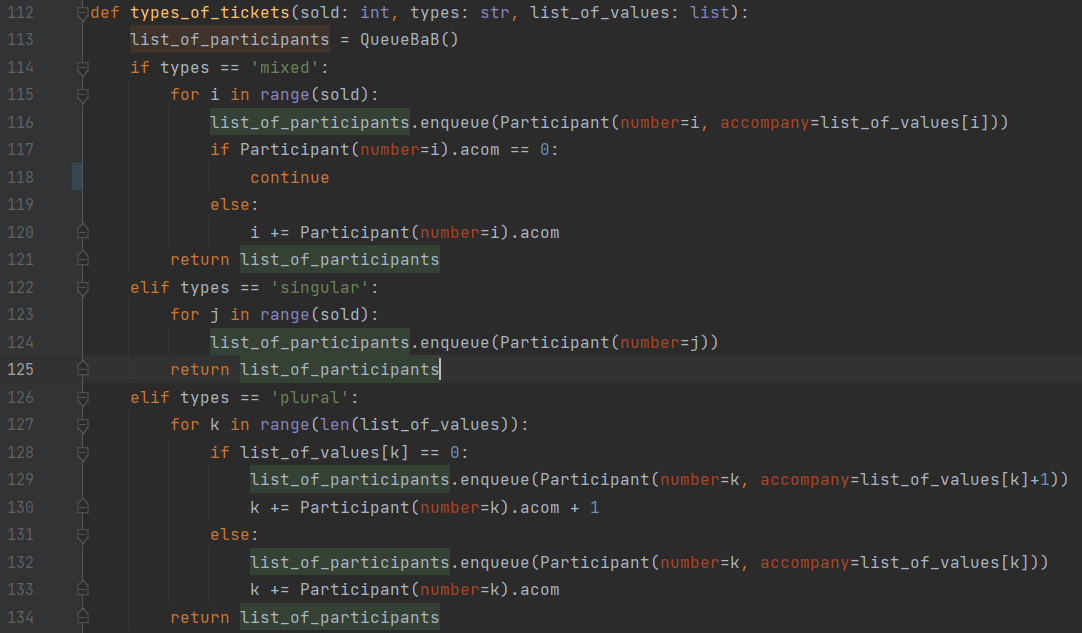
-gdy można zakupić zarówno bilety grupowe, jak i pojedyncze;

-gdy można zakupić bilety tylko grupowo.



Klasa *Participant* przyjmuje atrybuty *numer* i *acompany*, które oznaczają kolejno numer uczestnika w kolejce oraz ilość osób, z którymi uczestnik przyszedł.  
Klasa *Philharmonic* tworzy klasę, która opisuje ile drzwi jest przy zakupie biletu.

Korzystamy z modułu *numpy.random.normal*, który pozwala nam z danym prawdopodobieństwem określić iluosobowe bilety zostaną zakupione. Funkcja *list\_of\_normal\_dis\_values* tworzy listę o długości *lenght*(odpowiada ilości sprzedanych biletów) oraz parametry n i m odpowiednio oznaczają zakres prawdopodobieństwa oraz ilość liczb wyrzucanych przez funkcję.

W funkcji *types\_of\_tickets* tworzymy jedną kolejkę z uczestników i w zależności od listy zakupionych rodzajów biletów i przypisuje atrybuty obiektom z klasy *Participants*. Zostały wydzielone 3 przypadki: *„mixed”, „singular”* oraz *„plural”* (wśród zakupionych biletów są zarówno pojedyncze, jak i grupowe; tylko pojedyncze; tylko grupowe). Zwraca nam listę uczestników. 

Następnie przechodzimy do symulacji. Przyjmuje ona wartości takie jak: liczba osób zainteresowanych koncertem, liczba drzwi w filharmonii, typy biletów oraz ilości zakupionych biletów przez poszczególnych uczestnikow. Tworzymy kolejkę osób *participants* oraz w pętli tworzymy kolejne kolejki w zależności od liczby drzwi. Kolejkę *participants* dzielimy na mniejsze kolejki i tworzymy z nich listę. Kolejno przechodzimy przez każdą z nich i dopóki nie jest pusta, przekazujemy wartości z ilości zakupionycy biletów i tworzy listę jak długo trwa proces zakupu oraz wejścia do filharmonii. Na koniec sumuje wartości czasów i zwraca je.