

- I. Utwórz i wypełnij losowymi wartościami z przedziału 0 – 5 dwuwymiarową tablicę kwadratową `tab`. Następnie przygotuj statyczną metodę, sprawdzającą czy ta tablica ma wartości $a_{ij} = 0$ dla $i \neq j$, gdzie $i, j = 0, 1, \dots, n - 1$. Jeżeli nie jest spełniona powyższa zależność podniesiony zostanie wyjątek wskazujący, które elementy tablicy są nieprawidłowe np:

```
1 int [][] tab = {  
2     { 0, 0, 1},  
3     { 0, 1, 0},  
4     { 0, 1, 0}  
5 };
```

skutkuje błędem:

- 1 Tablica nie spełnia wymagań, błędy na pozycjach [0, 2], [2, 1]

- II. Zdefiniuj trzy metody, z których każda podniesie inny wyjątek:

- `podniesArrayIndexOutOfBoundsException()`;
- `podniesFileNotFoundException()`;
- `podniesException()`, z komunikatem *zła jednostka*.

Obsłuż wszystkie możliwe wyjątki w stosownej klauzuli `catch`

- III. Utwórz metodę `podajLiczbe` przyjmującą jako argument zmienną typu `String` i zwracającą wartość typu `int`, reprezentującą wartość dziesiętną liczby. Przyjmij że argument tej metody może przyjąć liczbę w systemie dwójkowym, ósemkowym lub szesnastkowym (zapisanych w formacie określonym przez twórcę programu). Jeżeli dostarczona wartość nie odpowiada przyjętemu wzorcowi podnieś wyjątek `NumberFormatException`.
- IV. Utwórz klasę `Rakieta`, zbudowaną z pól: `String nazwa` i `int wagaPaliwa`, konstruktora inicjującego pola oraz następujących metod: `zatankuj` - ładującą losową ilość paliwa, oraz `start` - rozpoczynającą procedurę startową. W ramach kontroli przedstartowej metoda `start` sprawdzi stan paliwa i jeżeli jest on mniejszy niż 1000, podniesiony zostanie wyjątek `Exception` z komentarzem 'start anulowany - za mało paliwa'.