

- I. [1 pkt.] Utwórz program wyświetlający jako pierwszy i ostatni komunikat odpowiednio *początek* i *koniec*. Następnie pomiędzy tymi poleceniami utwórz dziesięcioelementową tablicę i odwołaj się do elementu o indeksie 15. Zapewnij, aby pomimo błędu program osiągnął i wyświetlił ostatni komunikat (*koniec*).
- II. [1 pkt.] Utwórz program wyświetlający jako pierwszy i ostatni komunikat odpowiednio *początek* i *koniec*. Upewnij się że na dysku C w katalogu głównym nie ma pliku `1.txt`, następnie utwórz obiekt klasy `FileInputStream` podając jako parametr ścieżkę `C:\\1.txt`. Zapewnij aby pomimo błędu program osiągnął i wyświetlił ostatni komunikat (*koniec*).
- III. [1 pkt.] Zdefiniuj trzy metody, z których każda podniesie inny wyjątek:
- `podniesArrayIndexOutOfBoundsException()`;
 - `podniesFileNotFoundException()`;
 - `podniesException()`, z komunikatem *zla jednostka*.
- Obsłuż wszystkie możliwe wyjątki w stosownej klauzuli `catch`
- IV. [1 pkt.] 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'.
- V. [1 pkt.] Utwórz plik tekstowy i wpisz do niego ciąg znaków 'Ala ma kota, a kot ma Ale'. Następnie wczytaj cały ten tekst do jednej zmiennej typu `String` i wyświetl na ekranie.
- VI. [1 pkt.] Utwórz program sprawdzający, czy w pliku `pierwsze.txt` znajdują się wyłącznie liczby pierwsze, gdzie przez liczby rozumiemy ciągi cyfr oddzielone znakami białymi.