

Kontrola promów kosmicznych

Przygotuj następującą hierarchię wyjątków:

1. WyjątekUsterka i WyjątekTransportu dziedziczą z Exception.
2. WyjątekKolizja dziedziczy z WyjątekUsterka.
3. WyjątekBrakTlenu i WyjątekBiznesowy dziedziczą z WyjątekTransportu.

Wszystkie utworzone wyjątki posiadają konstruktor z parametrem typu String, w którym przekazują krótki opis powstałego błędu.

Przygotuj interfejs PojazdKosmiczny, który posiada deklarację następujących metod:

- getIloscTlenu() wyrzucający wyjątek WyjątekBrakTlenu,
- zuzycieTlenu(), z argumentem określającym ile tlenu zużyto.

Przygotuj następujący zestaw klas:

1. Klasa Rakieta implementująca interfejs PojazdKosmiczny. W niej należy zaimplementować metody zawarte w interfejsie.
2. Klasa PromKosmiczny dziedzicząca po klasie Rakieta.
3. Klasa KomputerKontrolny (klasa zawierająca metodę main)

Dobierz modyfikatory klas w taki sposób, aby: nie można było utworzyć żadnych obiektów klasy Rakieta i nie można było dziedziczyć od PromKosmiczny.

Pola i metody:

1. Rakieta:
 - pola: iloscTlenu, iloscZalogi, miejscePrzylotu, dniPozaZiemia
 - metody:
 - odpowiednie setery i gettery
 - toString() – „ładnie” sformatowana informacja o stanie rakiety
 - metoda getIloscTlenu() wyrzuca wyjątek z informacją: "Rakieta rozpoczęła przyspieszony powrót na Ziemię", gdy ilość tlenu spadnie do wartości wynoszącej minimum 100,
 - metoda zuzycieTlenu() ma, przy wywołaniu, zmniejszać ilość tlenu o wartość utrataTlenu,
 - metoda odliczająca kolejny dzień poza Ziemią
 - konstruktor parametryczny, do którego przekazujemy wartości wszystkich zmiennych, poza dniPozaZiemia, na początku = 0). Wyrzuca wyjątek WyjątekTransportu gdy ilość załogi składa się z mniej niż 5 osób (info: „Wczoraj kapitan miał urodziny, nie polecimy”) i WyjątekBrakuTlenu gdy ilość tlenu jest mniejsza niż 500 (info: „Mamy wyciek tlenu, polecimy, gdy tylko znajdzie się taśma klejąca”).

2. PromKosmiczny:

- pola: maxLadownosc, załadowanie (dobrać modyfikatory tych)
- metody:
 - odpowiednie gettery, setery oraz toString(),
 - konstruktor parametryczny. Wyrzuca WyjątekBiznesowy gdy załadunek jest mniejszy niż połowa ładowności (info: „Dorzućcie jeszcze towaru”).

3. KomputerKontrolny:

- metody:
 - główna metoda aplikacji, czyli main(),
 - losowyWypadek(prawdopodobienstwo) wyrzucająca wyjątki: WyjątekKolizja i WyjątekUsterka, Podczas wywołania ma wyrzucić losowo z pewnym prawdopodobieństwem jeden z wyjątków, świadczący o zaistnieniu wypadku. Info:
 - i. „Zderzenie z Marsjanami, bez paniki”
 - ii. „Problem z silnikiem grawitacyjnym, na zwykłym będziemy lecieć 1353 lata...”

W funkcji main() należy:

1. Utworzyć tablicę typu String zawierającą nazwy kilku potencjalnych miejsc do lądowania.
2. Utworzyć kolekcję Map, której kluczem jest nr kolejnej rakiety, a wartością rakieta.
3. Utwórz kolekcję (lista), w której elementami będą rakiety.
4. Włóż do kolekcji Map 10 promów kosmicznych (w pętli). Uwagi:
 - minimalna ilość tlenu w rakiecie wynosi 1000,
 - ilość załogi max 10 osób
 - miejsce przylotu wylosować z tablicy utworzonej w podpunkcie 1,
 - max ładowność zawsze 500
 - załadunek losowy
 - W sekcji catch() obsługi wyjątków należy wyświetlić odpowiedni komunikat i nie tworzyć danej rakiety(pamiętając, że **rakiet łącznie ma być 10**).
5. Elementami z kolekcji Map uzupełnij kolekcję utworzoną w podpunkcie 3. Użyj do tego foreach.
6. W pętli przeprowadzić symulację przebywania w przestrzeni kosmicznej. (Zagnieżdżona pętla: dopóki jakaś rakieta jest jeszcze na liście, przetwarzamy wszystkie)
Należy (używając utworzonej listy i iteratora listowego):
 - a. dla wszystkich rakiet sprawdzić aktualny stan tlenu i obniżyć go o wartość 70. Gdy komuś zabraknie tlenu należy usunąć go z listy
 - b. wywołać losowyWypadek z prawdopodobieństwem 10% i wypisać informację z wyjątku
 - c. zwiększyć ilość dni poza ziemią o 1
 - d. wypisać informacje o rakietach,

Punktacja:

- Przygotowanie hierarchii wyjątków i interfejsu - 1 pkt
- Przygotowanie klas - 4 pkt
- Funkcja main - 5 pkt