

Zaawansowane Programowanie Obiektowe i Funkcyjne

Strumienie

Zadanie oceniane nr 3b
13-11-2023

Kod wstępny znajduje w repozytorium z przykładami (zpoif_2023_lessons). Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium.

"Statki powietrzne 3"



W dniu dzisiejszym należy poddać analizie i przetwarzaniu kolekcję statków powietrznych (samoloty, śmigłowce, szybowce i spadochrony) ze względu na ich własności.

Należy dostarczyć klasę dostarczającą rozwiązanie, która przyjmuje jako argument konstruktora dwie kolekcje generowanego przez klasę `GeneratorStatkowPowietrznych`. W zależności od wymogów zadania, raz pracuje na jednej, a raz na drugiej kolekcji. Należy również dostarczyć klasę demonstracyjną, której metoda prezentuje na konsoli efekt działania implementacji poszczególnych metod.

Używając mechanizmów opartych na strumieniach należy dostarczyć implementację następujących metod:

- 1) Metoda: `NapedzanyStatekPowietrzny` `getNajszybciejWznoszacySieStatek();`
zwraca napędzany statek powietrzny o największej prędkości wznoszenia.
- 2) Metoda: `StatekPowietrzny` `getSamolotONajwPowierzchniNosnej();`
zwraca samolot o największej powierzchni nosnej, gdzie poszukiwanie zaczyna się dopiero od 6-tego statku powietrznego, z kolekcji napędzanych statków powietrznych.
- 3) Metoda: `Smiglowiec` `getSmiglowiecONajmniejszejMasie();`
zwraca śmigłowiec o najmniejszej masie, którego typ nie zaczyna się na "Mi".
Poszukiwania należy zacząć dopiero od 4-go śmigłowca na liście bazowej. Korzystamy z kolekcji napędzanych statków powietrznych.
- 4) Metoda: `Set<StatekPowietrzny>` `getSamolotyLubSmiglowceBezPierwszych4`
Zwraca nie większy niż 3 zbiór jakichkolwiek samolotów lub śmigłowców o prędkości wznoszenia nie większej niż 15 metrów/s i o masie mniejszej niż 1300 kg, po pominięciu pierwszych czterech statków spełniających podane wyżej parametry.
- 5) Metoda: `public List<NapedzanyStatekPowietrzny>` `get4SmiglowceONajwiekszymZasiegu()`
zwraca co najwyżej 4 śmigłowce o średnicy wirnika nie mniejszej niż 15 metrów, które mają największy zasięg (z kolekcji napędzanych statków powietrznych). W poszukiwaniach nie uwzględniamy 3 ostatnich śmigłowców na bazowej liście napędzanych statków powietrznych.

- 6) Metoda: `Spadochron getSiedzeniowySpadochron()`
Spśród siedzeniowych spadochronów ratowniczych zwrócić ten o najwyższej wysokości minimalnej.
- 7) Metoda: `Map<Integer, Szybowiec> getMapaSzybowcowPerDokonalosc;`
Zwraca mapę szybowców (począwszy od 2-giego już po wyeliminowaniu pozostałych statków powietrznych z kolekcji wszystkich statków powietrznych) gdzie kluczem jest ich doskonałość. W przypadku wystąpienia duplikatów, wstawiamy ten o dłuższej nazwie typu.
- 8) Metoda: `double getSumePredkosciWznoszeniaSamolotow();`
Zwraca sumę prędkości wznoszenia nie więcej niż 5-ciu samolotów o masie nie większej niż 15000. Analiza rozpoczyna od 4-go statku powietrznego. Ostatni samolot o tej masie jest pomijany.
- 9) Metoda: `Map<String, StatekPowietrzny> getPosortowaneSmigowceLubSamoloty();`
Zwraca mapę zawierającą samoloty lub śmigłowce, zbudowaną z 10-ciu najlżejszych statków powietrznych. Analiza rozpoczyna się od 11-go statku powietrznego na liście wszystkich statków powietrznych. W momencie tworzenia mapy i wcześniej nie powinno być duplikatów. Kluczem jest typ statku powietrznego.
- 10) `List<String> zwrocNazwy();`
Metoda zwraca listę stringów zawierających wszystkie dane spadochronów ratowniczych. Lista ograniczona do dwóch elementów. Ostatni spadochron ratowniczy na liście bazowej nie jest uwzględniany, powtórzenia usunięte.
- 11) Metoda: `void modyfikujNazwy();`
Z $P=0.1$ zmienia nazwę statku powietrznego poprzez otoczenie jej nawiasami kwadratowymi, jeśli jego masa jest większa niż 5 ton. Robi to tylko dla 15 statków powietrznych spełniających ryterium masy. Operacje są wykonywane dla samolotów z wyłączeniem pięciu najszybszych.