

Programowanie obiektowe – zadanie oceniane (2/4)

23-04-2018

Należy wysłać rozwiązanie w 90 minut po rozpoczęciu.
Liczba punktów do zdobycia: 25.

1. Wysyłanie rozwiązania

Należy spakować projekt w archiwum zip i nadać mu nazwę: login.zip

Należy wysłać email:

To: bednarzm@student.mini.pw.edu.pl

Subject: [PO] Zadanie2/4 2018

Załączniki: login_studenta.zip

2. Opis ogólny

Należy stworzyć aplikację symulującą pociąg pospieszny, zawierający charakterystyczne elementy dla tego środka transportu takie jak: konduktor, lokomotywa, wagon i oczywiście pasażer. Należy używać kolekcji i map, wszędzie, gdzie jest to potrzebne i użyteczne. W aplikacji trzeba posłużyć się również typami generycznymi i wyjątkami.



3. Opis szczegółowy

Część A (17 pkt).

Celem tego etapu jest stworzenie aplikacji posiadającej hierarchię potrzebnych klas będącej prostym modelem pociągu, zdolnej do zainicjowania samej siebie i wypełnienia pasażerami.

Pociąg zawiera następujące elementy:

➤ Pociąg

Klasa zawierająca wszystkie elementy takie jak:

- ➔ Lokomotywa – nic o niej nie musi być wiadomo, poza tym, że implementuje konfigurowalny interfejs. Klasa Pociąg ma być przygotowana na konfigurację typu.
- ➔ Kierownik pociągu – jeden obiekt
- ➔ Kilku konduktorów – nie ma znaczenia ich kolejność, nie mogą się dublować
- ➔ Wagony – kolekcja ta ma umożliwiać odwoływanie się do wagonów po ich indeksie, który nie może się zmieniać. Ich ilość powinna być łatwo zmienialna i zawiera się w losowym przedziale 5-15.

➤ Lokomotywa

Jest to obiekt, który implementuje interfejs zawierający dwie metody:

- ➔ go()
- ➔ stop()

➤ Wagon

Zawiera elementy takie jak:

- ➔ Pewna ilość przedziałów (6-10) w których siedzieć będą pasażerowie, konieczna jest możliwość dostępu do danego przedziału po indeksie. Pojemność każdego przedziału jest 6 lub 8 miejsc. To wagon przy tworzeniu tych obiektów ma decydować, ile osób wejdzie do każdego z nich. Wszystkie przedziały w danym wagonie mają mieć taką samą pojemność.
- ➔ Korytarz na którym pasażerowie mogą kontynuować podróż.

Powinien umożliwiać:

- ➔ Dostęp do przedziałów
- ➔ Informacje o ilości pasażerów aktualnie nim podróżujących
- ➔ Informacje o ilości bagażów znajdujących się w przedziałach

➤ Przedział

Jest to element wagonu.

Zawiera:

- ➔ od 6 do 8 miejsc. Miejsca mają być podatne na indeksowanie, tzn.. musi istnieć możliwość pobrania pasażera po danym indeksie.
- ➔ Półkę na bagaże. Nie ma ograniczeń na ilość bagażu na niej złożonego. Natomiast bagaże mają być ułożone w porządku: pierwszy jest najgłębiej, a ostatnio włożony najpłycej. Zdejmowanie odbywa się w odwrotnej kolejności. (jeśli bagaż x był włożony jako pierwszy, to zostanie zdjęty jako ostatni)

➤ Korytarz

Może na nim przebywać dowolna liczba pasażerów.

- Konduktor
 - Obiekt ten posiada następujące cechy:
 - ➔ Ma dostęp do wagonów, przedziałów, pasażerów oraz bagażów.
- Kierownik pociągu
 - Jest animatorem całego przedsięwzięcia. W tej części (część A) dokonuje załadunku pasażerów do wybranego wagonu.
- Pasażer
 - Jest to obiekt, który podróżuje koleją. Posiada elementy takie jak:
 - ➔ Bilet
 - ➔ Pewną liczbę sztuk bagażu (1-3)
- Bilet
 - Pusty obiekt.
- Peron
 - Obiekt zawiera 500 pasażerów, którzy chcą jechać pociągiem. Nie może być tam zdublowanych pasażerów ale musi być zachowana kolejność dodania do peronu.
- Bagaż
 - Obiekt, który posiada referencję do Pasażera (jeśli do niego należy).

Akcja:

- Należy zainicjować pociąg zgodnie z w/w wytycznymi.
- Kierownik pociągu pobiera pasażerów z peronu i każe im poszukać sobie miejsca poprzez wywołanie metody na każdym z pasażerów, której argumentem jest wylosowany wagon.
- Pasażer, któremu kierownik kazał poszukać sobie miejsca (po przekazaniu mu referencji na wagon):
 - ◆ listuje przedziały i w każdym sprawdza, czy jest tam wolne miejsce (liczba siedzących w przedziale jest mniejsza niż jego pojemność).
 - ◆ Jeżeli znalazł przedział w którym jest wolne miejsce, to "dodaje się" do niego i na półce bagażowej wstawia wszystkie swoje bagaże.
 - ◆ Jeżeli przelistuje wszystkie przedziały w tym wagonie i nie znajdzie żadnego wolnego, to stoi na korytarzu (dodaje się do tej kolekcji).

Część B (4 pkt).

Celem niniejszego fragmentu jest ożywienie pociągu i implementacja naturalnych procesów, które mu towarzyszą. Powinien być on już załadowany i gotowy do jazdy.

Kierownik pociągu (po załadunku pociągu) wykonuje następujące akcje:

- Uruchamia pociąg wywołując metodę go() na interfejsie implementowanym przez coś, co ma być lokomotywą.
 - ◆ Raz na 20 wywołań metody go() zdarza się sytuacja, która nie powinna się wydarzyć (której Kierownik Pociągu normalnie nie oczekuje):
 - Coś, co ma będzie lokomotywą zepsuło się
 - Prowadzący lokomotywę (zakładamy, że nie ma go w hierarchii obiektów), nie jest w stanie jej poprowadzić.
- Wtedy Kierownik Pociągu wypisuje na konsoli tekst: "Koniec jazdy!" i opróżnia pociąg z pasażerów i bagaży.

- Jeżeli udało się ruszyć, to losuje konduktora, któremu daje dostęp do wagonów wywołując na nim metodę: `sprawdzajBilety()`. Konduktor listuje wszystkie wagony i przedziały, które zawierają i sprawdza bilety. Pasażer powinien udostępniać metodę `getBilet()`.
- Jeden na 20 pasażerów nie ma biletu. Jest to sytuacja niestandardowa i nieoczekiwana w normalny trybie, więc proces musi być przerwany. Kierownik Pociągu:
 - ◆ zatrzymuje pociąg - `stop()`
 - ◆ usuwa pasażera
 - ◆ daje sygnał odjazdu – `go()`
 - ◆ każe konduktorowi ponownie sprawdzać bilety
- Po zakończeniu sprawdzania wszystkich biletów zatrzymuje pociąg (`stop()`), bo skład dojechał do końca podróży.
- Na zakończenie na konsoli pojawia się podsumowanie zawartości pociągu:
np. wagon 1, przedział 1, liczba pasażerów: 3, liczba

Część C (4 pkt).

Udziwnienia.

Istnieje rodzina konduktorów: dobry konduktor i zły konduktor. Jest ich do 5-ciu. Losowa liczba jednych i drugich.

Istnieje rodzina pasażerów: Dorosły, Dziecko i Student. Wśród pięciuset pasażerów ma ich być odpowiednio: 400, 50 i 50.

Kierownik pociągu losuje z zestawu konduktorów tego, który pójdzie sprawdzać bilety.

Dobry konduktor:

- W przypadku braku biletu wyrzuca tylko dorosłego. Resztę zostawia w spokoju pisząc na konsoli: "No no no!"

Zły konduktor:

- W przypadku braku biletu wyrzuca dorosłego i studenta. W przypadku dziecka wypisuje na konsoli: "No no no!"

Konduktorzy mają być przechowywani w samosortowalnej kolekcji. Najpierw dobrzy, potem źli.