Programowanie Obiektowe
Kolekcje, wyjątki i generyki
Zadanie oceniane nr 2b
25-04-2023

Po zakończeniu pracy konieczne jest wgranie zmian do repozytorium (add + commit + push) w katalogu o nazwie w stylu: zadanie_oceniane_2b. Fakt wgrania plików do swojego repozytorium można sprawdzić samodzielnie logując się (via www) na swoje konto i sprawdzając czy pojawiły się tam wszystkie zmiany.

"Terminal"

W dniu dzisiejszym oprogramowanie które stworzymy symulować będzie process obsługi samolotu pasażerskiego. W użyciu będą kolekcje, mapy ale **nie "surowe" tablice**. Zatem lecimy...

Prace do wykonania:

1. **Stworzyć klasy i ich hierarchie odzwierciedlające wspomniane poniżej elementy (nazewnictwo dobrać wedle uznania)**

   - Pasażer (bilet, bagaże główne (2-3) bez powtórzeń posortowane po id malejąco po każdym wstawieniu,
     bagaże podręczne (1-2) indeksowane z dużą ilością usunięć,
     oddajBagazeGłówne() - zwraca bagaże główne usuwając je jednocześnie z kolekcji,
     getBagazePodreczne() - zwraca bagaże podręczne,
     usunBagazPodreczny(...) - usuwa dany bagaż podręczny

     oraz

   - Bagaż główny (id, tylko przedmioty legalne (2-10) – unikalne przedmioty)
   - Bagaż podręczny (id, wszystkie przedmioty (2-10) – unikalne przedmioty pamiętające kolejność dodania,
     pobierzPrzedmiotyZBagazu(...) - zwraca przedmioty z bagażu do badania

     oraz

   - Przedmiot nielegalny (masa 5-28, getStopienNielegalnosci() - zwraca masę pomnożoną przez stałą 10)
   - Przedmiot legalny (getStopienNielegalnosci() – zawsze zwraca 0)
   - Kiełbasa (masa 1-2, getStopienNielegalnosci() - zwraca masę pomnożoną przez stałą 5)) Jest ona również przedmiotem nielegalnym.

     Bilet posiada następujące informacje: imię, nazwisko i jakiś losowany mało istotny numer loteryjny.

     Należy stworzyć generator biletów używany podczas inicjalizacji pasażera, dla którego stworzony zostanie bilet z niepowtarzalną kombinacją imienia i nazwiska, co stanowi jego unikalną cechę, więc trzeba o to jakoś zadbać. Dwóch Janów Kowalskich nie przejdzie. Losowanie kolejnej wejściówki do skutku, aż uzyskamy imię i nazwisko którego jeszcze nie było nie jest tu porządanym rozwiązaniem więc trzeba to zrobić lepiej. Można to utożsamiać z losowaniem bez zwracania.

     Zakładamy że miarą równości bagażu jest takie samo id. W pozostałych przypadkach liczy się fizyczność – czyli dwie instance są tożsame jeśli ich nośnikiem jest ten sam obiekt.

   - Bramka bagażowa
     Jest w stanie przetrzymywać niepowtarzalne bagaże główne w relacji z instancjami biletu, które do niej dodano. Posiada metodę zdajBagaz(), która przyjmuje referencję na bilet i bagaż i wrzuca go do tej struktury. Bramka chce udostępniać tylko tą metodę na zewnątrz ukrywając swą tożsamość.

   - Bramka bezpieczeństwa
     Jest w stanie przechowywać informację o nielegalnych przedmiotach, które pobrała z bagażu podręcznego jako relację pasażera i indeksowanej kolekcji tych właśnie przedmiotów. Posiada metodę boolean skanujBagaż(...), która wywołuje na jego

przedmiotach metodę getStopienNielegalnosci(). Jeśli jest on większy od 0 (to od razu zwraca false i dodaje do nielegalnych) dodatkowo sprawdza w dobrze znany aczkolwiek mało elegancki sposób (przymknę na to oko) czy przypadkiem nie jest to kiełbasa. Jeśli tak, to jest to sytuacja w obliczu której metoda nie może kontynuować swojego działania. W dobrze znany sposób informuje o tym wszystkich dookoła przekazując informację na zewnątrz. Bramka chce udostępniać tylko tą metodę na zewnątrz ukrywając swą tożsamość.

- Operator bramki bezpieczeństwa
  Nie zna się na obsłudze bramki bagażowej.
  Ma dostęp do bramki bezpieczeństwa (właściwie czegoś co udostępnia metodę skanujBagaże) i wystawia metodę sprawdzPasazera(), który przekazany jest jako argument. Pobiera od niego listę bagaży podręcznych i po kolei przekazuje do bramki bezpieczeństwa. Wypisuje na konsolę informację o skanowaniu bieżącego bagażu. Wie że metoda ta w dobrze znany sposób może odmówić współpracy, wtedy odpowiednio to obsługuje tak żeby sprawa była załatwiona na miejscu i tutaj pisząc na konsoli: "Kielbasie mówimy stanowcze nie!!!". Jeżli bagaż posiada coś nielegalnego to jest on usuwany z bagaży podręcznych pasażera.

- Operator bramki bagażowej
  Nie zna się na obsłudze bramki bezpieczeństwa.
  Ma dostęp do bramki bagażowej (właściwie czegoś co udostępnia metodę zdajBagaz(...)). Pobiera bagaże od pasażera i przekazuje je do bramki bagażowej. Wypisuje na konsolę informację o zdaniu bieżącego bagażu.

- Terminal
  Posiada operatorów bramek, niepowtarzalną kolekcję z pasażerami. Metoda go() przeprowadza pasażerów przez wszystkie bramki.

2. **Uczynić Terminal parametryzowalnym**
Dodać do Terminala pole z obiektem zajmującym się reklamą o klasie/typie który nie może być znany w momencie implementacji ale będzie on wskazany dopiero podczas tworzenia maszyny. W momencie instancjonowania Terminala zakładamy że typem tym będzie obiekt stworzonej klasy Neon spoza hierarchii, który będzie utworzony podczs instancjonowania terminala. Reprezentacja tekstowa Neonu zwraca znany napis na murze niedaleko Dworca Centralnego: "Lataj razem z nami..." Poprawić odpowiednie miejsca w kodzie (łącznie ze światem).

3. **Zademonstrować działanie**
Uruchomić Terminal i przekazać do niego 50 pasażerów.