Kolokwium poprawkowe z Programowania Obiektowego

5 czerwca 2009

Celem zadania jest stworzenie systemu umożliwiającego prowadzenie aukcji. System powinien działać w turach oraz posiadać interfejs umożliwiający wykonanie następujących operacji:

- przejście do kolejnej tury,
- wyszukanie aktywnych (niezakończonych) aukcji, których opis zawiera określony napis,
- wyszukanie zakończonych aukcji, których opis zawiera określony napis,
- wyszukanie aukcji, które wystawił użytkownik o danym loginie,
- wyszukanie aukcji, w których zwyciężył użytkownik o danym loginie,
- przyznanie „punktów sympatii" twórcy przez zwycięzcę aukcji i zwycięzcy przez twórcę.

Każdy użytkownik systemu może zapytać o aktualną cenę aukcji, wziąć udział w aukcji (zalicytować) oraz sprawdzić jej historię (listę wszystkich ofert złożonych do tej pory). Aby zalicytować, użytkownik musi określić cenę, za którą chce kupić dany przedmiot. Cena ta musi być wyższa od dotychczasowej. Dodatkowo użytkownik musi mieć wystarczającą liczbę „punktów sympatii". Jeżeli te warunki nie są spełnione, próba licytacji kończy się porażką. Pytanie o cenę lub licytacja aukcji zakończonej również kończy się porażką.

Użytkownik tworzący aukcję podaje jej opis, cenę początkową, czas trwania aukcji (liczbę rund, przez które będzie ona aktywna) oraz minimalną liczbę „punktów sympatii", które musi posiadać użytkownik chcący wziąć udział w tej aukcji. Niektóre typy aukcji mogą wymagać dodatkowych parametrów.

Poza listą aukcji system powinien prowadzić rejestr użytkowników. System musi pamiętać imię, nazwisko, login oraz liczbę „punktów sympatii" danego użytkownika. „Punkty sympatii" są przyznawane pomiędzy twórcą a zwycięzcą aukcji. Każdy z nich może przyznać drugiemu 1, 0 lub -1 punkt w danej aukcji.

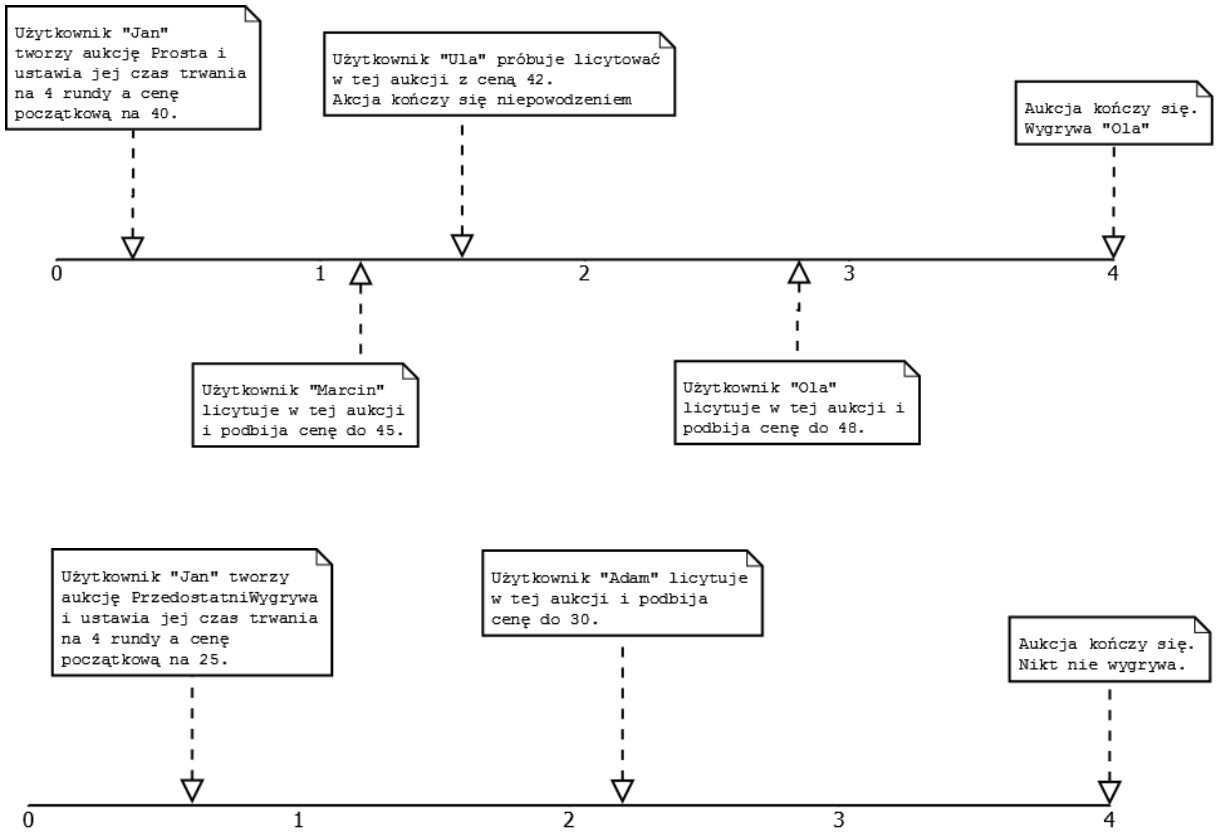
W pierwszej wersji system powinien umożliwiać tworzenie następujących typów aukcji:

- Aukcja Prosta – jak sama nazwa wskazuje jest to najprostszy typ aukcji. Wygrywa uczestnik, który złożył najwyższą ofertę przed końcem ostatniej rundy aukcji.
- Aukcja CzasoWydłużająca – w tym typie aukcji również wygrywa uczestnik, który złożył najwyższą ofertę. Jednak w przeciwieństwie do aukcji prostej każde złożenie poprawnej oferty wydłuża czas aukcji o określoną liczbę rund.
- Aukcja DrugiWygrywa - jest to eksperymentalny typ aukcji. Wygrywa ją nie uczestnik, który złożył najwyższą ofertę lecz ten, który złożył drugą co do wielkości. W tym typie aukcji próba zalicytowania przez tego samego użytkownika dwa razy z rzędu kończy się porażką.

Polecenia:

(1) Zaprojektuj i opisz hierarchię klas potrzebnych w tym zadaniu.
(2) Zaimplementuj metody i konstruktory potrzebne w tym zadaniu.

Przykłady:



Uwagi:

Nie jest częścią zadania stworzenie kodu uruchamiającego system ani przeprowadzającego licytacje, tylko stworzenie i zaimplementowanie interfejsu systemu umożliwiającego to.

Można założyć, że lista użytkowników jest przekazywana do systemu w konstruktorze. Nie trzeba implementować dodawania i usuwania użytkowników.

Przydatne metody w klasie String:

`public int indexOf(String str)` – daje indeks, od którego rozpoczyna się pierwsze wystąpienie słowa „str” w danym słowie lub wartość -1, jeżeli „str” nie jest podsłowem danego słowa

Przydatne metody w klasie ArrayList<E>:

`public boolean add(E e)` – dodaje element „e” na koniec listy

`public E get(int index)` – daje element znajdujący się na pozycji „index”

`public boolean remove(Object o)` – usuwa pierwsze wystąpienie obiektu „o” z listy. Daje true, jeśli obiekt „o” występował na liście, false w przeciwnym przypadku.

`public int size()` – daje liczbę elementów listy.