

LABORATORIUM 5

Należy wykonać zadanie 1 lub 2.

1. Napisz program, który synchronizuje wiele wątków współdzielących wspólne zasoby. Program może zajmować się animacją figur. W programie zakłada się, że:
 - 1.1. zasobem współdzielonym jest powierzchnia bieżącego panela (obiekt typu JPanel lub pochodny) wstawionego do okna typu JFrame. Powierzchnia ta jest podzielona prostokątne obszary, zwane sektorami. Informacje o sektorach są przechowywane w wybranej kolekcji – zajęty sektor przechowuje informację o tym, że zawiera figurę.
 - 1.2. Wątkami są figury. W danym obszarze może znajdować się tylko jedna figura. Jeśli dany obszar jest zajęty, figura, która powinna być przemieszczona na zajęty obszar zostaje uśpiona. Zostaje ona obudzona w momencie, gdy figura zajmująca dotąd dany obszar, opuści go i przemieści się na do wolnego sektora. Podczas przemieszczania figur między sektorami bada się zajętość danego sektora kontrolując dane o sektorach umieszczone w kolekcji wspólnego zasobu – każda figura zajmując wolny sektor, wpisuje do komórki dane o jego zajętości, a zwalniając sektor i przemieszczając się do wolnego sektora anuluje zajętość dotychczas zajmowanego sektora.
 - 1.3. Figury można animować podobnie jak w programach z laboratorium 4.
2. Napisz komunikator wielowątkowy oparty na gniazdach Socket i ServerSocket. Klienci aplikacji mają dostęp do wspólnego zasobu, którym jest np. system zrealizowany w programie z laboratorium 3, umieszczony na serwerze aplikacji. Należy korzystając ze strumieni serializowanych ObjectOutputStream oraz ObjectInputStream przysyłać dane i umieszczać je w katalogu biblioteki. Należy tak synchronizować operacje na danych, aby nie dopuścić do jednoczesnego usuwania i wyświetlania tych samych danych przez dwóch różnych klientów.

Przykłady zadań na ocene celującą

1. Napisz komunikator wielowątkowy oparty na gniazdach Socket i ServerSocket.
 - 1.1. Klienci aplikacji mają dostęp do wspólnego zasobu, którym jest np. system zrealizowany w programie z laboratorium 3, umieszczony na serwerze aplikacji. System ten należy uzupełnić o klasy realizujące wypożyczalnię książek. Należy korzystając ze strumieni serializowanych ObjectOutputStream oraz ObjectInputStream przysyłać dane i umieszczać je w katalogu biblioteki. Należy tak synchronizować operacje na danych, aby nie dopuścić do jednoczesnego usuwania i wyświetlania tych samych danych przez dwóch różnych klientów.
 - 1.2. Należy wykonać dwa typy klientów: klienta maszyny wirtualnej Java (patrz projekt komunikator_3) oraz klienta jako aplet.
2. Napisz komunikator wielowątkowy oparty na protokole RMI.
 - 2.1. Klienci aplikacji mają dostęp do wspólnego zasobu, którym jest np. system zrealizowany w programie z laboratorium 3, umieszczony na serwerze aplikacji. System ten należy uzupełnić o klasy realizujące wypożyczalnię książek. Należy korzystając z metod zdalnych przysyłać dane i umieszczać je w katalogu biblioteki. Należy tak synchronizować operacje na danych, aby nie dopuścić do jednoczesnego usuwania i wyświetlania tych samych danych przez dwóch różnych klientów.
 - 2.2. Należy wykonać dwa typy klientów: klienta maszyny wirtualnej Java (patrz projekt komunikatorRMI_3) oraz klienta jako aplet.