
A Zasady współpracy

Ocena rozwi za

- 3.0 – 25÷40 punktów
- 3.5 – 41÷65 punktów
- 4.0 – 66÷80 punktów
- 4.5 – 81÷100 punktów
- 5.0 – 101÷130 punktów

Warunki zaliczenia przedmiotu

Student uzyska ocen zaliczaj c (3.0) o ile dostarczone przez niego rozwi zania dla ka dego z poni szych zada zostan ocenione na co najmniej 5 punktów:

- New I/O
- Client/Server
- Web Applications – wprowadzenie
- jedno z wybranych (1) Remote Method Invocation – lub (2) Java Messaging Service
- Web Services – wprowadzenie

1 New I/O – współdzielony zmapowany plik

Stwórz dwie proste aplikacje – względnie jedną aplikację działającą w dwóch różnych trybach – które komunikują się wzajemnie poprzez współdzielony plik, którego zawartość jest zmapowana na pamięć operacyjną.

Pierwsza aplikacja powinna zapisywać dane do pliku – np. dwie liczby całkowite. Druga powinna umożliwiać odczyt danych ze zmapowanego pliku oraz dodatkowo wykonywać dowolną operację – np. dodawać odczytane wartości. Wynik przetwarzania powinien być przez drugą aplikację wyprowadzany na ekran.

2 Client/Server – własny protokół typu requests-response

Stwórz rozwiązanie typu klient/serwer oparte o własny prosty tekstowy (komunikacja jest czytelna dla człowieka) protokół umożliwiający wykonanie następujących operacji:

- przesyłanie treści otrzymanej wiadomości z powrotem do klienta (*echoing messages*);
- dodawanie wartości dwóch liczb przesłanych przez stronę klienta i przekazanie z powrotem wyniku.

3 Web Applications – wprowadzenie

Stwórz prosty serwlet Java™, którego funkcjonalność polega na dodaniu wartości dwóch liczb całkowitych przekazanych w postaci parametrów do formularza oraz wyświetleniu wyniku poniżej wspomnianego formularza. Użytkownik aplikacji powinien mieć możliwość przekazania parametrów do serwletu zarówno za pomocą metody GET, jak i POST.

Twój serwlet powinien być odporny na wprowadzenie przez użytkownika danych w niepoprawnej formie, względnie danych niekompletnych.

4 Web Applications – Model-View-Controller

Zmodyfikuj architekturę rozwiązania powstałego w ramach zadania 3, tak by była ona zgodna z koncepcją Model-View-Controller dla aplikacji web omówionej w ramach wykładu z przedmiotu z TPR.

5 Web Applications – AJAX

Zmodyfikuj rozwiązanie zadanie 3 lub 4, tak aby wynik operacji dodawania był wyliczany natychmiast po wprowadzeniu danych wprowadzonych w pola formularza – bez konieczności nacisnięcia przycisku “Submit” lub naciśnięciu <Enter>.

Wyliczenie powinno być oparte o mechanizm AJAX – tzn. powinno być uruchamiane przez skrypt napisany w JavaScript w chwili, gdy fokus opuszcza jedno z pól tekstowych.

6 Web Applications – dost p do bazy danych

Stwórz prost aplikacj web realizuj c podstawowe operacje typu CRUD (Create-Retrieve-Update-Delete) na baize danych o poni szym schemacie.

Aplikacja powinna składa si z dwóch głównych ekranów umo liwiaj cych:

- przegl danie listy u ytkowników;
- dodanie/edycja konkretnego u ytkownika – u ytkownik aplikacji jest przekierowany do ekranu edycji po klikni ciu hiperł cza lub przycisku umieszczonego obok rekordu reprezentuj cego danego u ytkownika na li cie.

UWAGA: Twoje rozwi zanie powinno wykorzystywa mechanizm rółła danych (DataSource) wspieraj cy pule połącze do bazy danych.

USERS	
PK	<u>USER_ID</u>
	USER_NAME USER_PASSWORD

7 Web Applications – Filtry

Zmodyfikuj własne rozwiązanie stworzone w ramach zadania 3, 4 lub 5, tak aby (1) nagłówek oraz (2) stopka generowanych stron web były dodawane przez omówiony w trakcie wykładu mechanizm filtrów.

8 RMI – echo treści przesyłanych komunikatów oraz dodawanie wartości

Stwórz rozwiązanie typu klient/serwer oparte o technologii RMI realizujące operacje omówione w zadaniu 2, czyli:

- echo treści komunikatów otrzymanych od klienta;
- dodawanie wartości argumentów liczbowych.

9 Web Services – Wprowadzenie

Rozszerz funkcjonalno usługi SampleService umieszczonej w przykładach dołączonych do wykładu o metod Add dodaj c dwa składniki przekazane jako argumenty wywołania.

10 Web Services – Wywołania asynchroniczne

Rozszerz funkcjonalno usługi TimeoutSampleService umieszczonej w przykładach dołączonych do wykładu o metodzie TimeoutAdd. Funkcjonalno nowej metody – podobnie jak metody Add omówionej w zadaniu 9 – polega na dodaniu wartości dwóch parametrów. Różnica w działaniu między metodami omówionymi w zadaniu 9 a TimeoutAdd polega na ustawieniu opóźnienia (np. 10 sekund) w wysłaniu odpowiedzi do aplikacji klienckiej.

Zaimplementuj asynchroniczne wywołanie metody TimeoutAdd po stronie klienckiej, tak aby wywołanie wspomnianej metody nie skutkowało “zamrożeniem” aplikacji klienckiej.

11 Web Services – zwykłe i „odporne” wyzwalanie akcji

Rozszerz funkcjonalno usługi `TriggerActionSampleService`, której kod źródłowy znajduje się w materiałach wiczeniowych dla wykładu, o metod `AddTriggerAction`.

Funkcjonalno nowej metody `b` dzie odpowiadała metodzie `Add` omówionej w poprzednich zadaniach. W odróżnieniu od wcześniejszych wariantów metody `Add` `AddTriggerAction` `b` dzie umieszczała wynik dodawania w logu po stronie serwerowej zamiast przekazywać bezpośrednio do aplikacji klienckiej. Zapis w logu jest rozumiany jako zapis do pliku lub wyprowadzenie komunikatu poprzez standardowe wyjście (`System.out`) lub standardowy błąd (`System.err`) procesu serwera.

Rozszerz funkcjonalno usługi `RobustTriggerActionSampleService` umieszczonej w materiałach pomocniczych do wykładu dodając metod `AddRobustTriggerAction`. Komunikacja z usług `RobustTriggerActionSampleService` powinna odbywać się w oparciu o wzorzec wymiany komunikatów `ROBUST-IN-ONLY` umożliwiający przekazywanie informacji o błędzie stronie wywołującej.

Podobnie jak w przypadku metody `AddTriggerAction` metoda ta również `b` dzie wyprowadza wynik dodawania na standardowe wyjście lub błąd. Po wyprowadzeniu wyniku metoda powinna również zgłaszać wyjątek, który ze względu na użycie wzorca wymiany komunikatów `ROBUST-IN-ONLY` powinien być przekazany do strony klienckiej, gdzie następnie powinien być obsłużony poprzez wyprowadzenie komunikatu o błędzie za pomocą okna dialogowego.

12 Enterprise Java Beans – komponenty sesyjne

Stwórz prosty komponent sesyjny realizujący podstawowe operacje arytmetyczne: (1) dodawania, (2) odejmowania, (3) mnożenia i (4) dzielenia wykonywane na dwóch argumentach.

Dokonaj wdrożenia stworzonego komponentu na wybranej implementacji kontenera EJB.

Stwórz prostą aplikację kliencką umożliwiającą weryfikację poprawności działania stworzonego komponentu.