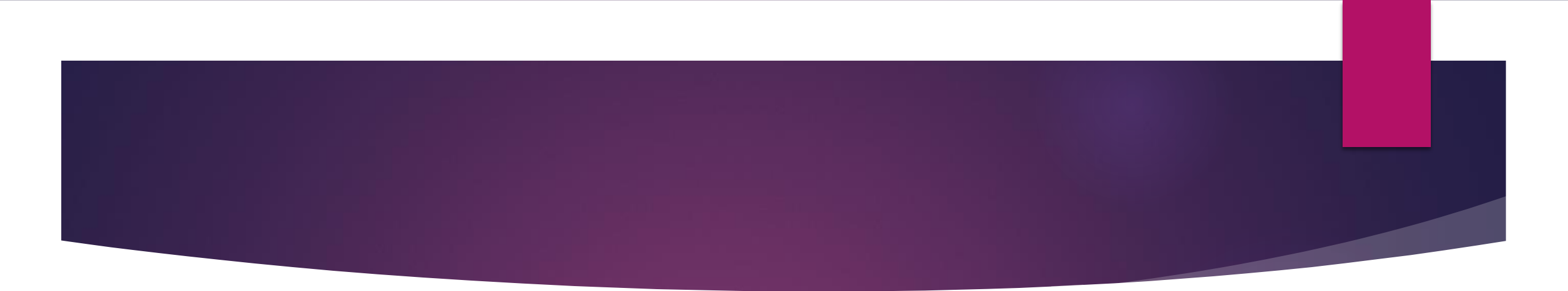


Wzorce reaktywne. Java

EE 8

Czym jest programowanie reaktywne?

- ▶ Przez długi czas aplikacje przetwarzały żądania w sposób synchroniczny.
- ▶ W procesie synchronicznym każde wywołanie funkcji lub zasobu działa sekwencyjnie. Jeżeli zadanie wymaga dużej ilości czasu na wykonanie (np. Żądanie I/O - wejścia/wyjścia), to blokuje wykonywanie innych zadań.
- ▶ Programowanie reaktywne to paradygmat programowania, w którym kładzie się nacisk na tworzenie asynchronicznych i nie blokujących się komponentów. Rdzeniem programowania reaktywnego jest strumień danych, który możemy obserwować i reagować na niego.
- ▶ Programowanie reaktywne najczęściej wykorzystuje się w dużych projektach oraz ma większą złożoność od rozwiązań klasycznych.

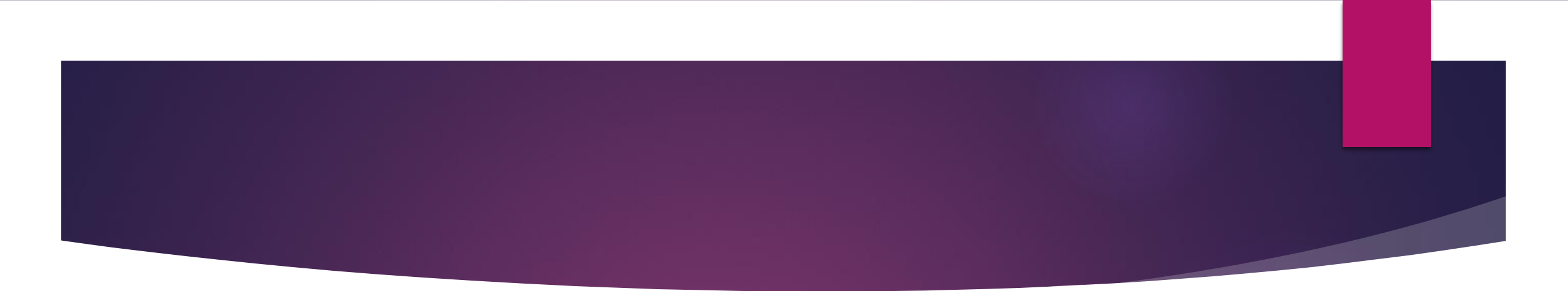
- 
- ▶ W programowaniu reaktywnym elementy reagują na zdarzenia, następnie gdy użytkownik żąda jakiegoś zasobu, to uruchamia zdarzenie, które obsługuje strumień danych.
 - ▶ Gdy to zdarzenie się rozpocznie, element lub zadanie reaguje na strumień danych i przetwarza swój algorytm.
 - ▶ Dzięki temu aplikacja internetowa może bez problemu przetwarzać dużą ilość danych.

Paradygmat reaktywny – cztery zasady

- ▶ Paradygmat działa w oparciu o następujące cztery zasady, aplikacje powinny być:
 - Elastyczne (Elastic): Reaguje na wymagania. Aplikacje mogą wykorzystywać wiele rdzeni i wiele serwerów do przetwarzania żądań.
 - Odporne (Resilient): Reaguje na błędy. Aplikacje mogą reagować na awarie i błędy w oprogramowaniu, sprzęcie i sieciach oraz naprawiać je.
 - Oparte na wiadomościach (Message-driven): Reaguje na zdarzenia. Aplikacje składają się z menedżerów zdarzeń, które są asynchroniczne i nieblokujące, w przeciwieństwie do wielu synchronicznych wątków.
 - Responsywne (Responsive): Reaguje na użytkowników. Aplikacje powinny zapewniać szybką i spójną reakcję.

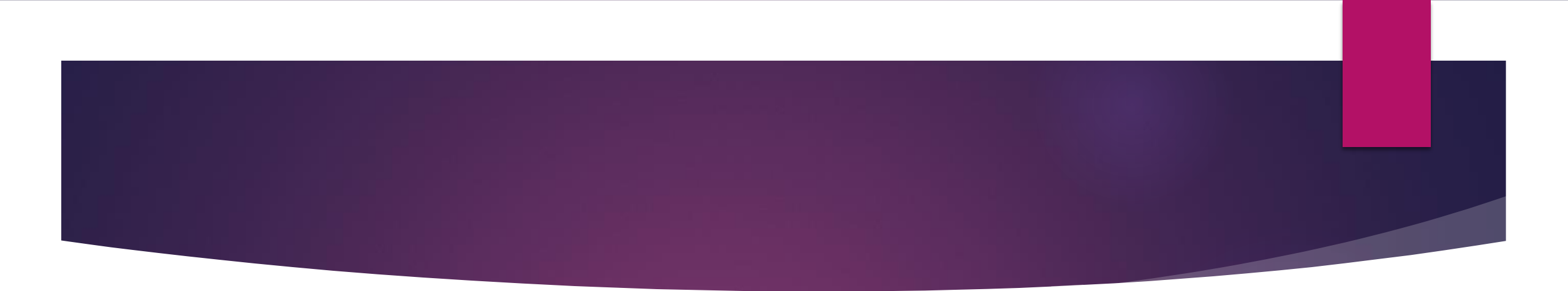
Zdarzenia CDI

- ▶ Wraz z rozwojem programowania reaktywnego zostały wprowadzone nowe narzędzia umożliwiające tworzenie systemów przy użyciu programowania reaktywnego. Jednym z takich rozwiązań jest zdarzenie CDI, które może uruchomić zdarzenie synchroniczne i blokujące lub zdarzenie asynchroniczne i nie blokujące przy użyciu CDI.
- ▶ W podejściu blokującym jeden wątek obsługuje żądanie po czym czeka na odpowiedź np. z bazy danych. Dopiero jak otrzyma odpowiedź to może wrócić do klienta.
- ▶ W podejściu nieblokującym jeden wątek odpowiada za obsługę żądania i kiedy pojawi się żądanie deleguje on obsługę do puli wątków, które są za to odpowiedzialne.

- 
- ▶ Zdarzenie CDI zostało stworzone przy użyciu wzorca obserwatora, umożliwiając tworzenie i uruchamianie zdarzenia do oddzielnych komponentów wykonujących przetwarzanie. Obserwator reaguje na wywołane zdarzenia.
 - ▶ Zdarzenie w CDI to mechanizm specyfikacji CDI, który może być wykorzystywany we wszystkich warstwach aplikacji. Jednak użycie tego mechanizmu jest zalecane w przypadku pracy z warstwą prezentacyjną. Wynika to z faktu, że CDI koncentruje się głównie na warstwie prezentacyjnej, a jej zakresy są bezpośrednio związane z interakcją HTTP i sesją HTTP.

Asynchroniczna metoda EJB

- ▶ Asynchroniczna metoda EJB pozwala klientowi wywołać metodę i otrzymać jej zwrot zaraz po wywołaniu metody. Zwrot metody jest w kontroli asynchronicznego wywołania reprezentowanego przez obiekt `Future<T>`. Klient może kontrolować wykonanie metody asynchronicznej. Działania te mogą anulować metodę inwokacji, sprawdzić, czy inwokacja jest zakończona, sprawdzić, czy inwokacja uruchomiła wyjątek, oraz sprawdzić, czy inwokacja została anulowana.

- 
- ▶ Metoda asynchroniczna EJB nie wykorzystuje paradygmatu programowania reaktywnego, ale jest procesem asynchronicznym, który umożliwia skrócenie czasu odpowiedzi na zapytanie klienta. Zaleca się stosowanie go na warstwie biznesowej. Korzyścią z zastosowania tego mechanizmu jest to, że możemy korzystać również z innych mechanizmów EJB, takich jak kontrola transakcji.

Różnica między EJB a CDI

- ▶ Metoda EJB działa na zasadzie jeden do jednego pomiędzy wywołującym a wywoływanym
- ▶ CDI działa na zasadzie jeden do wielu
- ▶ Metoda EJB nie wykorzystuje wzorca obserwatora
- ▶ CDI wykorzystuje wzorzec obserwatora

REST service

- ▶ REST service to mechanizm specyfikacji JAX-RS, który jest w stanie stworzyć usługę REST, która może wykonywać przetwarzanie asynchroniczne.
- ▶ Asynchroniczny REST service jest asynchronicznym procesem, który ułatwia przetwarzanie wątków
- ▶ W żądaniu wysłanym do serwera nowy wątek może zostać wywołany w celu przetworzenia zadania nieblokującego
- ▶ JAX-RS wspiera asynchroniczne przetwarzanie w API klienta i API serwera
- ▶ Dzięki temu mechanizmowi kontrola procesu jest zwracana natychmiast po wysłaniu żądania, dzięki czemu klient nie musi długo czekać na wykonanie innych zadań

- ▶ API klienta może być zakończone poprzez asynchroniczne wywołanie, które zwraca obiekt `Future<T>`, gdy tylko żądanie zostanie wykonane, co pozwala klientowi kontrolować to wywołanie i zastosować do niej akcje.
- ▶ Można to zrobić poprzez programowanie reaktywne i może zwrócić obiekt `CompletionState<T>`, gdy tylko żądanie zostanie wykonane.
- ▶ Oznacza to, że można kontrolować inwokację i wybrać metodę wywołania zwrotnego, wykonywaną w zależności od etapu.



Dziękujemy za uwagę

PREZENTACJA WYKONANA PRZEZ:

ERNEST PERŻYŁO

PAWEŁ ZEGARSKI

KRYSTIAN ŻEŃCZAK