

Integracja wybranych technologii programistycznych oraz technik programowania obiektowego w projektowaniu systemów informatycznych.

Autor:

Bączkowski Karol

Promotor:

dr inż. Paweł FIGAT



Integracja technologii obiektowych

- ❑ Integracja jest to całokształt działań zmierzających do scalenia różnych rozwiązań informatycznych.
- ❑ W miarę rozwoju nowych technologii informatycznych integracja nabiera coraz większego znaczenia.
- ❑ Wprowadzenie techniki obiektowej pozwoliło na rozpoczęcie badań nad integracją języków programowania.

Cel pracy

- ❑ Integracja obiektowych języków programowania: C# i Java oraz PHP i ActionScript.

Hipoteza

□ Czy i w jakim stopniu możliwa jest integracja języków Java i C# a także języka PHP i ActionScript ?

Zakres pracy

- ☐ Zaprezentowanie obiektowości
- ☐ Ujednolicony język modelowania UML
- ☐ Porównanie podejścia obiektowego z podejściem proceduralnym
- ☐ Charakterystyka integrowanych technologii obiektowych
- ☐ Zaprezentowanie praktycznych przykładów integracji technologii obiektowych

Obiektość

❑ Oprogramowanie obiektowe jest zbiorem współpracujących ze sobą obiektów posiadających swój stan, inaczej rzecz ujmując dane oraz zachowanie, czyli metody operujące na danym obiekcie.

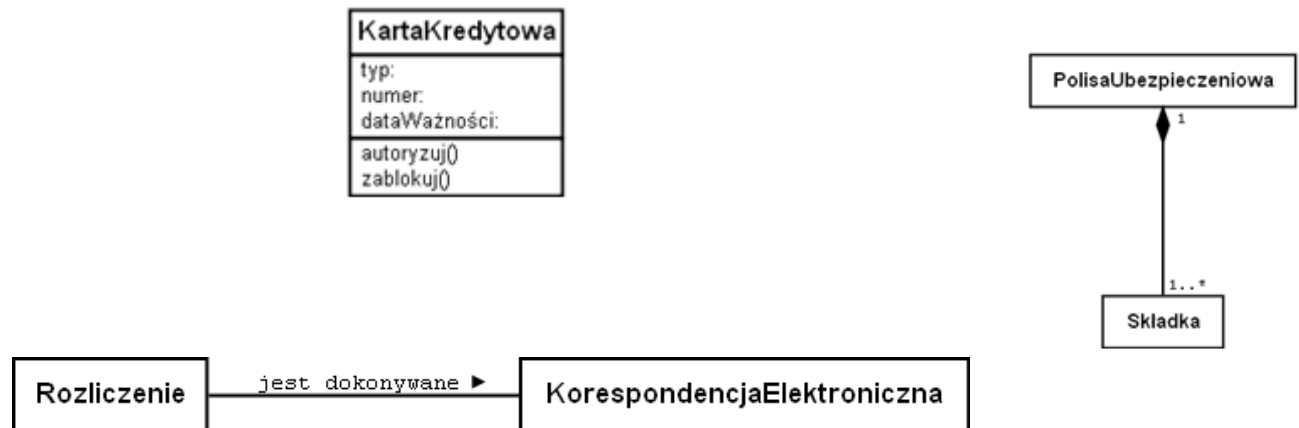
❑ Cechy obiektości:

- ❑ Abstrakcyjne typy danych (klasy)
- ❑ Dziedziczenie
- ❑ Polimorfizm
- ❑ Kapsułkowanie



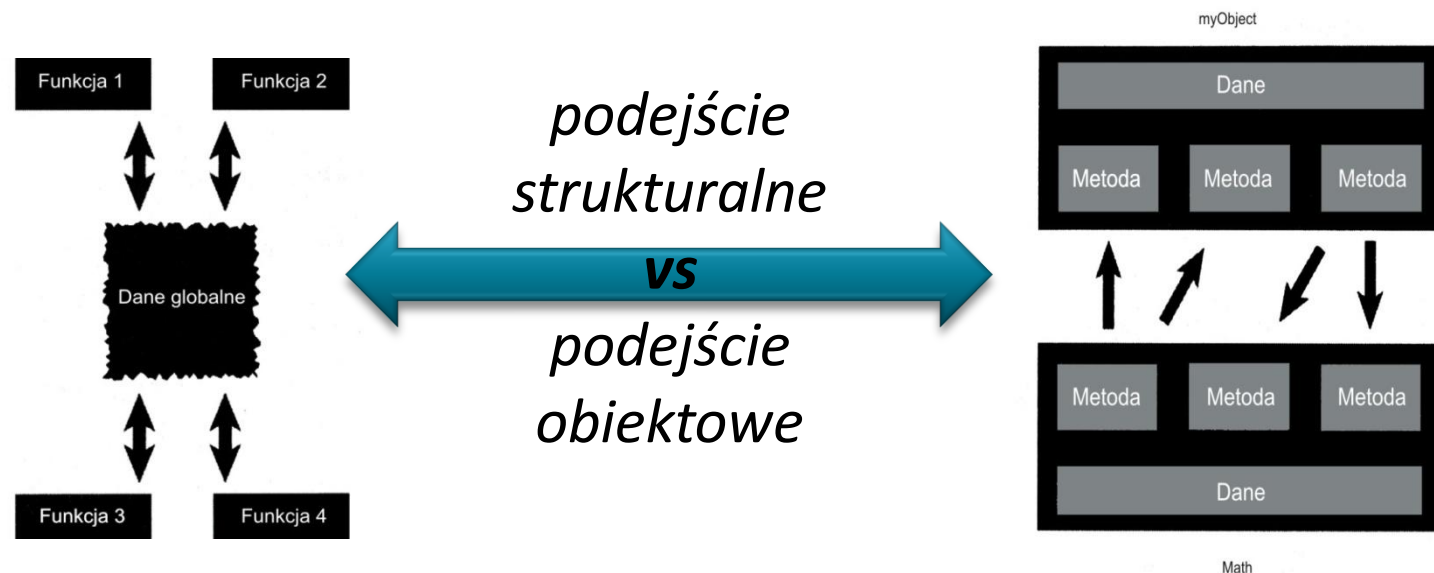
UML

- ❑ Ujednolicony język modelowania systemów informatycznych
- ❑ Zalety:
 - ❑ Możliwość modelowania statyki i dynamiki projektowanego systemu
 - ❑ Niezależny od stosowanej technologii projektowania obiektowego



Podejście obiektowe a strukturalne

- ❑ Programowanie obiektowe – atrybuty i zachowania w umieszczone są w jednym spójnym obiekcie
- ❑ Programowanie strukturalne – atrybuty i zachowania są od siebie oddzielone



Integrowane technologie

- ☐ Platforma .NET

- ☐ Wspólny system typów CTS
- ☐ Wspólna specyfikacja języka CLS
- ☐ Środowisko uruchomieniowe CLR
- ☐ Wspólnie powiązane biblioteki klas FCL
- ☐ Kompilacja kodu programu do kodu MSIL
- ☐ Kompilacja JIT

- ☐ Język C#

- ☐ Interfejsy
- ☐ Struktury
- ☐ Delegaty



Integrowane technologie

- ☐ Java jako język
 - ☐ Prosty
 - ☐ Zorganizowany obiektowo
 - ☐ Sieciowy
 - ☐ Niezawodny
 - ☐ Bezpieczny
 - ☐ Niezależny od architektury
 - ☐ Przenośny
 - ☐ Interpretowany
 - ☐ Wysokowydajny
 - ☐ Wielowątkowy
 - ☐ Dynamiczny



Integrowane technologie

☐ Język PHP

- ☐ Najpopularniejszy język skryptowy
- ☐ Zastosowanie – aplikacje internetowe
- ☐ Działa po stronie serwera



☐ Język ActionScript

- ☐ Zintegrowany ze środowiskiem Adobe Flash
- ☐ Zastosowanie – tworzenie dynamicznych aplikacji Flash
- ☐ Działa po stronie klienta



Integracja Java i C# - JNBridgePro

❑ Oprogramowanie stanowiące pomost w trakcie łączenia technologii Java i .NET.

❑ Możliwości:

❑ Przechwytywanie wyjątków jednej technologii w drugiej

❑ Zarządzanie cyklem życia obiektów obu technologii

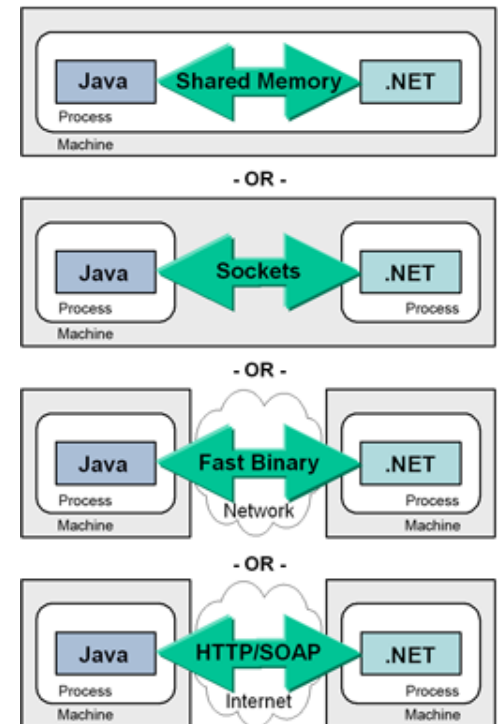
❑ Wsparcie dla mechanizmu oczyszczania pamięci

❑ Konwertowanie typów jednej technologii w tożsame typy drugiej technologii

Java side	Object & class level	.NET side
Java side	Objects	.NET rich-client GUIs
Java SE (J2SE)	Classes	ASP.NET
Java EE (J2EE)	Instance members	Microsoft BizTalk Server
J2EE app servers	Static members	SharePoint Server
EJBs	Fields	WinForm controls
JMS	Properties	WPF
JNDI	Methods	
Standalone JVMs	Generics	
AWT	Enums	
SWT	Events	
Swing	Transactions	

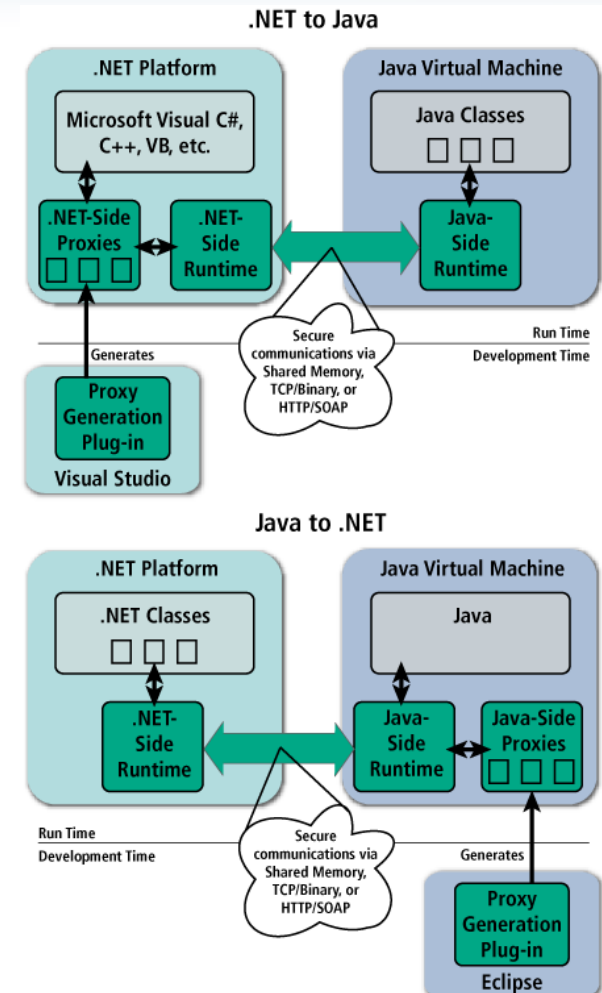
JNBridgePro – architektura połączenia

- ❑ Jedna maszyna wykorzystująca ten sam proces – pamięć współdzielona
- ❑ Jedna maszyna wykorzystująca różne procesy – gniazda
- ❑ Sieć lokalna – szybki protokół binarny
- ❑ Sieć Internet – protokoły HTTP/SOAP



JNBridgePro – działanie

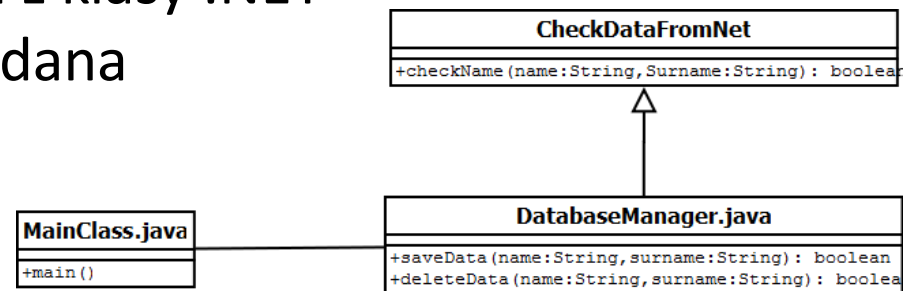
- ❑ Zestaw pośredników którzy zarządzają klasami i komunikacją pomiędzy technologią .NET i Java
- ❑ Automatyczne generowanie pośredników do wybranych klas, a także wszystkich klas wymaganych przez te klasy
- ❑ Utworzenie pośredników za pomocą Visual Studio, Eclipse, lub niezależne narzędzie producenta



Przykłady praktyczne

- ❑ Dziedziczenie klasy Java z klasy .NET

- ❑ Integracja w pełni udana



- ❑ Związek klasy .NET z klasą Java

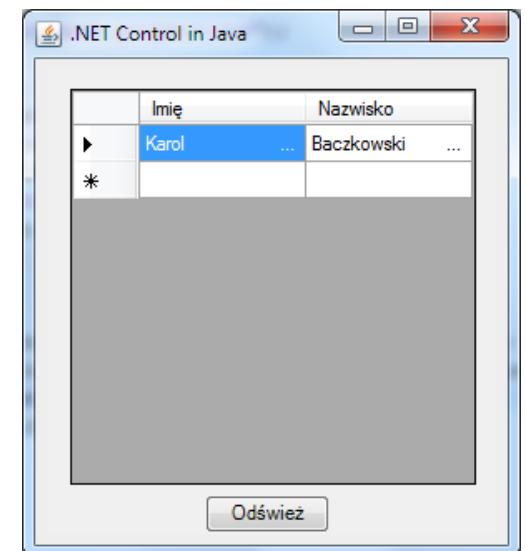
- ❑ Integracja w pełni udana

- ❑ Graficzna kontrolka technologii

- .NET wykorzystana w technologii Java

- ❑ Błędy narzędzia JNBridgePro podczas generowania pośredników

- ❑ Integracja w pełni udana



Integracja PHP i ActionScript

- ❑ Obsługa integracji ActionScript z PHP wbudowana w środowisko Flash
- ❑ Możliwość pobierania danych z PHP
- ❑ Możliwość wysyłania danych do PHP
- ❑ Wykorzystanie zalet pracy po stronie serwera i klienta w jednej aplikacji Flash



Przykłady praktyczne

- ☐ Zastosowanie wbudowanej funkcji PHP w ActionScript
 - ☐ Integracja w pełni udana
- ☐ Zapis danych Flash do bazy MySQL przy pomocy PHP
 - ☐ Problemy z polskimi znakami w aplikacji Flash
 - ☐ Integracja w pełni udana
- ☐ Weryfikacja danych Flash i pobranie danych z bazy MySQL przy pomocy PHP
 - ☐ Problemy z kodowaniem znaków w PHP i MySQL
 - ☐ Integracja w pełni udana

Podsumowanie 1/2

- ☐ Na podstawie przeprowadzonych badań, stwierdzono, że integracja technologii obiektowej Java i .NET oraz PHP i ActionScript jest możliwa.
- ☐ W przypadku technologii Java i .NET zrealizowano:
 - ☐ Dziedziczenie klasy jednej technologii w drugiej
 - ☐ Tworzenie obiektów jednej technologii w drugiej
 - ☐ Wykorzystanie graficznych kontrolek jednej technologii w drugiej
- ☐ W przypadku technologii PHP i ActionScript zrealizowano:
 - ☐ Wykorzystanie wbudowanych funkcji jednej technologii w drugiej
 - ☐ Pracę z bazą danych za pomocą integracji

Podsumowanie 2/2

- ☐ Pomimo udanego procesu integracji .NET oraz Java do utrudnień należy zaliczyć:
 - ☐ Zamknięty kod integrujący technologie
 - ☐ Oprogramowanie będące w ciągłej fazie rozwoju
 - ☐ Cenę oprogramowania integrującego
 - ☐ Brak literatury dotyczącej integracji
- ☐ W przypadku PHP i ActionScript:
 - ☐ Brak dobrego wsparcia dla polskich znaków podczas przekazywania danych w formie tekstu
 - ☐ Integracja możliwa jedynie w jedną stronę
 - ☐ Zamknięty kod w aplikacji Flash

Literatura 1/2

- ❑ Marcin Lis, „PHP5. Praktyczny kurs”, Helion, Gliwice, 2006
- ❑ Adobe Creative Team, „ActionScript 3.0 dla Adobe Flash CS4/CS4 PL Professional. Oficjalny podręcznik”, Helion, Gliwice, 2009
- ❑ Matt Weisfeld, „Myślenie obiektowe w programowaniu”, Helion, Gliwice, 2010
- ❑ Joseph Schmuller, „UML dla każdego”, Helion, Gliwice, 2001
- ❑ Michał Śmiałek, „UML 2.0 metody modelowania obiektowego”, Helion, Gliwice, 2005
- ❑ Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski, „Język UML 2.0 w modelowaniu systemów informatycznych”, Helion, Gliwice, 2005

Literatura 2/2

- ❑ Yourdon Edward, „Analiza obiektowa i projektowanie - przykłady zastosowań”, WNT, Warszawa, 2000
- ❑ Jesse Liberty, „Programowanie C#”, Helion, Gliwice, 2006
- ❑ Cay S. Horstmann, Gary Cornell, „Java Podstawy Wydanie VIII”, Helion, Gliwice, 2008
- ❑ Biała księga języka Java
<http://java.sun.com/docs/overviews/java/java-overview-1.html>
- luty 2011
- ❑ John Coggeshall, „PHP5. Księga eksperta”, Helion, Gliwice, 2005
- ❑ Roger Braunstein, Mims H. Wright, Joshua J. Noble, „ActionScript 3.0. Biblia”, Helion, Gliwice, 2009
- ❑ Oficjalna strona narzędzia JNBridgePro

Koniec

Dziękuję za uwagę