

Asocjacji to grupa powiązań o tej samej semantyce i strukturze. (T)
Asocjacje skierowane są zawsze bardziej użyteczne od dwukierunkowych. (N)
Asocjacje z modelu obiektowego są implementowane w modelu relacyjnym jako tabele. (N)
Asocjacje skierowane są zawsze bardziej użyteczne od dwukierunkowych. (N)
Asocjacje występują bezpośrednio w języku Java. (N)
Architektura klient-serwer sprzyja redukcji ruchu sieciowego. (T)
Architektury warstwowe ułatwiają tworzenie współczesnego oprogramowania. (T)
Architektura klient-serwer wymaga mocnego komputera centralnego. (T)
Architektura mainframe nie wymaga mocnego komputera centralnego. (?)
Architektura systemu nie musi uwzględniać celów biznesowych. (N)
Atrybut klasowy ma zawsze stałą wartość. (N)
Atrybut opcjonalny zawsze musi mieć jakąś wartość. (N)
Atrybut wyliczalny może być usunięty bez szkody dla informacji zapamiętanych w systemie. (T)

Dobry programista nie musi używać komentarzy. (N)
Do zapisywania ograniczeń UML, można wykorzystywać tylko język OCL. (N)
Dziedziczenie dynamiczne występuje bezpośrednio w języku Java. (N)
Dziedziczenie typu disjoint występuje bezpośrednio w języku Java. (T)
Dziedziczenie wieloaspektowe występuje bezpośrednio w języku Java. (N)
Dziedziczenie, wielokrotne występuje bezpośrednio w języku Java. (N)

Ekstensja klasy nie występuje bezpośrednio w języku Java i dlatego musi być ręcznie zaimplementowana. (T)
Ekstensja klasy może być zaimplementowana w ramach tej samej klasy biznesowej lub korzystając z klasy dedykowanej. (T)

Implementacja asocjacji n-arnej w modelu relacyjnym wykorzystuje tabelę pośredniczącą. (T)
Implementacja asocjacji rekurencyjnej o licznosci „* - *” wymaga wykorzystania dwóch tablic lub dwóch kontenerów. (T)
Implementacja asocjacji kwalifikowanej w modelu relacyjnym może bazować na specyficznym doborze klucza obcego. (T)
Implementacja interfejsu przez klasę oznacza umieszczenie kodu tych metod w klasie. (N)
Implementacja kompozycji może odbywać się w oparciu o dziedziczenie. (N)
Implementacja ograniczenia {history} jest analogiczna do implementacji ograniczenia {bag}. (T)
Implementacja w modelu relacyjnym atrybutu klasowego nie jest zbyt prosta. (T)
Implementacja w modelu relacyjnym atrybutu opcjonalnego polega na dołożeniu dodatkowej kolumny do tabeli. (N)
Implementacja w modelu relacyjnym pojedynczy wiersz tabeli może reprezentować jedną osobę. (N)
Implementacja w modelu relacyjnym atrybutu wyliczalnego polega na usunięciu klucza głównego z tabeli. (?)
Interfejsy w języku Java umożliwiają zdefiniowanie metod. (T)
Istnieją dwa zasadnicze sposoby implementacji asocjacji za pomocą identyfikatorów lub korzystając z referencji (wskaźników). (T)

Jak pokazują testy, wydajność języka Javy jest ok. 100 razy słabsza od C++. (N)
Jednym z czynników wpływających na popularność systemów relacyjnych są aplikacje spadkowe. (T)
Jednym ze sposobów rozwiązania problemu niezgodności impedancji jest stosowanie dziedziczenia dynamicznego zamiast wielodziedziczenia. (N)
Jednym ze sposobów rozwiązywania problemu niezgodności impedancji jest wprowadzenie istotnych udogodnień z zakresu baz danych do języka programowania. (T)
Jednym ze sposobów obejścia problemu braku dziedziczenia w modelu relacyjnym jest dodanie dwóch kluczy głównych do każdej z tabel. (N)
Jeden ze sposobów obejścia problemu braku dziedziczenia wieloaspektowego w modelu relacyjnym, polega na zastąpieniu go dziedziczeniem dynamicznym. (N)
Jednym z głównych powodów korzystania z baz danych jest język zapytań. (T)
Dziedziczeniem dynamicznym. (N)
Jednym z głównych powodów korzystania z baz danych jest język zapytań. (T)
Jedną z technik optymalizacji, wykorzystywaną przez Hibernate, jest leniwa inicjalizacja obiektów. (T)
Jedną z wad wykorzystania JDBC w języku Java jest niezgodność impedancji. (T)

Klasa asocjacji z modelu obiektowego, jest implementowana w modelu relacyjnym jako tabela pośrednicząca. (T)
Klasa asocjacji umożliwia podanie dodatkowych informacji dotyczących asocjacji. (T)
Klasa jest zbiorem obiektów. (N)
Klasa jest wystąpieniem obiektu. (N)
Klasy kontenerowe języka Java umożliwiają przechowywanie tylko referencji do obiektów. (T)
Klasy szablonowe (generics) w języku Java mogą przechowywać m.in. typ Object. (T)
Klasa ObjectPlusPlus bezpośrednio wspiera wszystkie cechy kompozycji. (N)
Korzystanie z mechanizmu serializacji języka Java, do uzyskania trwałości ekstensji, oznacza pisanie wielu linii kodu. (N)
Konstrukcja „wszystkieCzesci.contains(czesc)” umożliwia sprawdzenie czy kontener „wszystkieCzesci” zawiera „czesc” (język Java i standardowy pojemnik). (T)

Mapowanie asocjacji w Hibernate wykorzystuje klasy pojemnikowe języka Java. (?)
Metoda obiektu ma dostęp do wszystkich obiektów danej klasy. (N)

Napisanie ograniczenia UML, zwykłym tekstem poprawia precyzję informacji. (?)

Ograniczenie {XOR} oznacza, że może istnieć tylko jedno powiązanie w ramach ograniczonych asocjacji. (T)
Ograniczenia w UML, są jedynym z mechanizmów rozszerzalności. (T)
Ograniczenie statyczne bazuje na poprzednim stanie elementu ograniczanego. (N)
Ograniczenie {subset} może dotyczyć jednej asocjacji. (N)
Ograniczenie {subset} może dotyczyć asocjacji. (T)
Ograniczenie {bag} dla asocjacji oznacza, że może istnieć wiele powiązań pomiędzy tymi samymi obiektami. (T)
Optymalna implementacja asocjacji kwalifikowanej, odbywa się w oparciu o pojemnik mapujący. (T)

„Pensja nie może zmaleć” jest przykładem ograniczenia dynamicznego. (T)
Plik mapujący wykorzystany w technologii Hibernate zawiera m.in. informacje dotyczące mapowanych atrybutów. (T)
Plik mapujący wykorzystywany w technologii Hibernate zawiera między innymi informacje dotyczące nadklas mapowanej klasy. (?)
Poważną wadą technologii LINQ jest brak możliwości korzystania z języka zapytań. (N)
Poważną wadą technologii Hibernate jest brak możliwości korzystania z języka zapytań. (N)
Programista języka Java może tworzyć własne klasy wyjątkowe. (T)
Przechodząc z modelu obiektowego na relacyjny, atrybuty złożone możemy zastąpić dodatkowymi tabelami. (T)
Przechodząc z modelu obiektowego na relacyjny, atrybuty proste zastępujemy kolumnami w tabelach. (T)
Przy implementacji ograniczenia {XOR} warto użyć jednej z wersji metody contains(...). (T)
Przy implementacji ograniczenia {subset} warto użyć metody containsKey(...) lub containsValue(...) z klasy pojemnika. (T)

Słowo kluczowe „break” języka Java służy do niszczenia obiektów. (N)
Słowo kluczowe catch języka Java służy do implementacji dziedziczenia. (N)
Słowo kluczowe „extends” języka Java służy do implementacji ekstensji. (N)
Sposoby obejścia braku dziedziczenia w modelu relacyjnym są zbliżone do tych wykorzystywanych przy przejściu z modelu pojęciowego do implementacyjnego. (T)

Tablice języka Java umożliwiają przechowywanie wielu elementów. (T)
Tablice języka Java umożliwiają łatwe wstawianie elementów. (N)
Technologia Hibernate umożliwia mapowanie asocjacji „* - *”. (T)
Technologia LINQ między innymi dodaje język zapytań do języka programowania. (T)
Trwałość ekstensji można symulować m.in. przez zapis/odczyt do/z pliku. (T)
Trwałość ekstensji jest bezpośrednio dostępną w języku Java. (N)

Wadą technologii Hibernate jest brak możliwości mapowania asocjacji. (N)
Wadą modelu obiektowego jest to, że aby odnaleźć powiązane obiekty trzeba (potencjalnie) przejrzeć całą ich ekstensję. (N)
W architekturze trójwarstwowej, warstwa pośrednicząca zajmuje się między innymi interfejsem użytkownika. (N)
W architekturze dwuwarstwowej, serwer nie zajmuje się udostępnianiem danych. (N)
W architekturze trójwarstwowej, warstwa pośrednicząca nie zajmuje się wymianą komunikatów, ponieważ jest to zadanie Webservice. (N)
W architekturze trójwarstwowej, warstwa pośrednicząca nie zajmuje się obsługą transakcji ponieważ jest to zadanie bazy danych. (N)
W celu wykorzystania pełni możliwości Hibernate, każda klasa musi zawierać identyfikator. (?)
W czasie implementacji kompozycji należy rozwiązać problem blokowania samodzielnego tworzenia części. (T)
W czasie implementacji kompozycji należy rozwiązać problem usuwania części przy usuwaniu całości. (T)
W czasie implementacji kompozycji, należy rozwiązać problem usuwania całości przy usuwaniu części. (N)
Większość rodzajów dziedziczenia można obejść za pomocą kompozycji. (T)
Widok logiczny architektury uwzględnia komponenty oraz ich interfejsy. (T)
Widok koncepcyjny architektury uwzględnia komponenty oraz ich interfejsy. (N)
W języku Java buforowanie odczytu danych ze strumienia ma na celu poprawę wydajności. (T)
W języku Java, buforowanie z plikami może odbywać się w oparciu o strumienie. (T)
W języku Java dodanie obsługi archiwów ZIP wymaga systemów skomplikowanej implementacji. (N)
W języku Java implementacja asocjacji o licznosci „* - *” wymaga użycia tablicy lub kontenera. (T)
W języku Java, implementacja dziedziczenia typu „incomplete” odbywa się za pomocą specjalnego słowa kluczowego „partial”. (N)
W języku Java komunikacja z plikami może odbywać się w oparciu o strumienie. (T)
W języku Java, klasa może implementować co najwyżej 3 interfejsy. (N)
W języku Java, obiekt klasy zewnętrznej ma bezpośredni dostęp do inwariantów obiektu klasy wewnętrznej. (N)
W języku Java ograniczenie UML „pensja nie może zmaleć” implementujemy w postaci specjalnej metody. (T)
W języku Java, technologia JDBC służy do implementacji dziedziczenia dynamicznego. (N)
W języku Java, wszystkiewołania metod SA domyślnie polimorficzne. (T)
W klasie ObjectPlus, metody pomocnicze służą między innymi do wyświetlania ekstensji klasy. (T)
W klasie ObjectPlusPlus, jeden zbiorczy pojemnik mapujący przechowuje wszystkie powiązania konkretnego obiektu. (?)
W klasie ObjectPlus, kontener służący do przechowywania wszystkich ekstensji, musi umożliwiać przechowywanie duplikatów referencji. (N)
Właściwie dobrana architektura systemu ułatwia kierowanie zespołem wykonawców. (T)
Właściwie dobrana architektura systemu sprzyja ponownemu użyciu. (T)
W przypadku niektórych rodzajów asocjacji, nazwy ról są wymagane. (T)
W przypadku implementacji ekstensji klasy w ramach klasy zewnętrznej metody klasowe muszą być oznaczone słowem static. (N)
W przypadku obejścia dziedziczenia dziedziczenia wieloaspektowego, jeden z aspektów realizujemy za pomocą klasycznego dziedziczenia udostępnianego przez język. (T)
W modelu relacyjnym pojedynczy wiersz tabeli może reprezentować jedną osobę. (T)
W modelu relacyjnym tabela pośrednicząca (dla relacji „* - *”) zawiera co najmniej dwa klucze obce. (T)
Wprowadzenie klas szablonowych (generics) do języka Java miało za zadanie zwiększenie bezpieczeństwa typologicznego. (T)
W przypadku implementacji ekstensji klasy w ramach klasy zewnętrznej, metody klasowe muszą być oznaczone słowem static. (N)
W przypadku implementacji ekstensji klasy w ramach klasy zewnętrznej, odpowiedni kontener musi być oznaczony słowem static. (N)
W przypadku implementacji ekstensji klasy w ramach tej samej klasy odpowiedni kontener musi być oznaczony słowem static. (T)
W przypadku implementacji ekstensji klasy w ramach tej samej klasy, metody klasowe muszą być oznaczone słowem static. (T)
W stosunku do architektury dwuwarstwowej, architektura trójwarstwowa posiada tak zwaną warstwę pośredniczącą. (T)
W technologii Hibernate, mapowanie asocjacji oraz atrybutów (powtarzalnych) wielowartościowych niczym się nie różni. (N)
Wykorzystanie asocjacji kwalifikowanej ma umożliwić przechowanie większej liczby informacji. (N)
Wykorzystanie kompozycji do opisu zależności pomiędzy klasami oznacza użycie specyficznych licznosci. (T)
Wykorzystanie mechanizmu serializacji języka Java oznacza, że zamiast tych samych obiektów, mamy takie same obiekty. (T)
Wykorzystanie serwera aplikacji zmniejsza ogólne koszty pielęgnacji oprogramowania. (T)

Zaletą obejścia dziedziczenia wielokrotnego wykorzystującego interfejsy, jest łatwość umieszczania w nich atrybutów. (?)
Zasadniczą zaletą obejścia dziedziczenia za pomocą kompozycji jest ogromna prostota implementacji. (N)
Zasadniczą wadą obejścia dziedziczenia za pomocą kompozycji jest poważnie zwiększone zużycie pamięci. (N)
Zjawisko określane jako niezgodność impedancji jest spowodowane połączeniem modelu obiektowego oraz relacyjnego. (T)
Zjawisko określane jako niezgodność impedancji jest spowodowane słabą wydajnością relacyjnych baz danych. (?)
Zasadniczą wadą obejścia dziedziczenia za pomocą kompozycji jest poważnie zwiększone zużycie pamięci. (?)
Zaletą obejścia dziedziczenia wielokrotnego wykorzystującego interfejsy, jest łatwość umieszczenia w nich atrybutów. (N)