

# Modelowanie i Analiza Systemów Informatycznych (MAS)

Egzamin

Imię	Nazwisko	Nr indeksu	Grupa

Wypełnione podczas ostatniego wykładu

Zestaw B

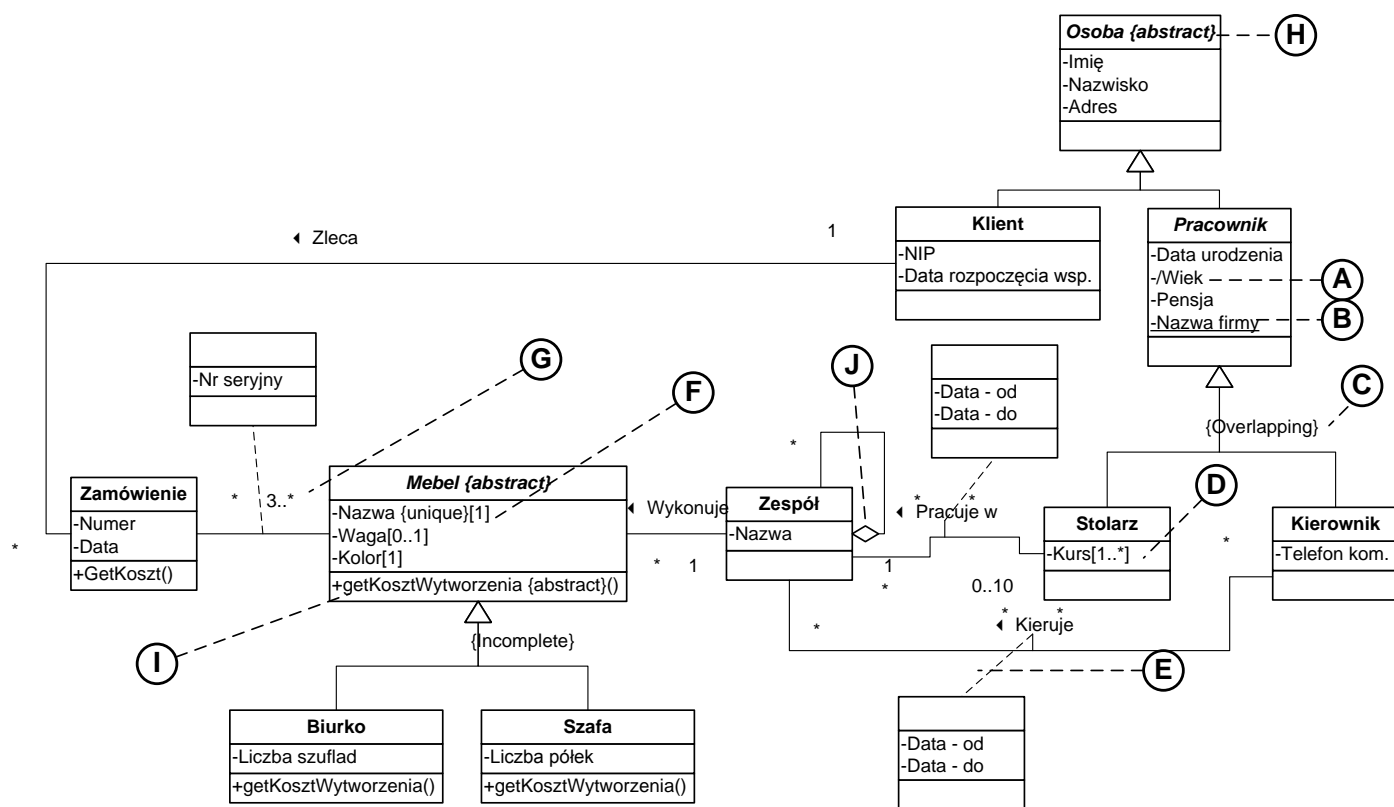
Wszystkie odpowiedzi są potwierdzone przez wykładowcę

1. Pytania testowe. Proszę zaznaczyć prawdziwe twierdzenia (litera **T**) lub nieprawdziwe (litera **N**) stawiając odpowiednią literę przy kwadracie. Prawidłowa odpowiedź oznacza +1 pkt., błędna -1 pkt, brak odpowiedzi 0 pkt (całość: min. 0 pkt., maks 20 pkt.).

- N** W modelu relacyjnym, w przypadku kolumny typu INT, wartość null oznacza 0.
- N** W klasie ObjectPlus, kontener służący do przechowywania wszystkich ekstensji, może być łatwo zaimplementowany przy użyciu klasy Vector.
- T** Technologia zwana refleksją (ang. *Reflection*) może posłużyć do odczytania budowy/struktury klasy.
- T** Przy implementacji ograniczenia {XOR} warto użyć jednej z wersji metody contains(...) znajdującej się w klasie pojemnikowej.
- T** Metoda implementująca ograniczenie {subset} powinna uwzględniać nazwę roli asocjacji.
- N** Korzystanie z deklaratywnego podejścia do tworzenia GUI wymaga dużego zaangażowania ze strony programisty.
- T** W popularnych językach programowania (np. Java) ograniczenia nie występują bezpośrednio.
- T** Ograniczenie {bag} dla asocjacji oznacza, że powiązania pomiędzy tymi samymi obiektami są uporządkowane.
- N** Implementacja atrybutu złożonego może polegać na zdefiniowaniu nowego typu.
- N** Zjawisko określane jako niezgodność impedancji jest związane ze słabą wydajnością relacyjnych baz danych.
- T** Plik mapujący wykorzystywany w technologii Hibernate zawiera m. in. informacje dotyczące asocjacji mapowanej klasy.
- T** Asocjacje n-arną implementujemy wykorzystując klasę pośredniczącą.
- T** Implementacja agregacji w modelu relacyjnym jest dokładnie taka sama jak asocjacji.
- T** Jednym ze sposobów kształtowania wysokiej użyteczności jest obserwacja sposobu pracy użytkowników.
- N** Agregacja z modelu obiektowego, jest implementowana w modelu relacyjnym jako dwie tabele pośredniczące i 4 klucze obce.
- T** Mechanizm serializacji języka Java, ułatwia uzyskanie trwałości ekstensji.
- T** Głównym powodem implementacji asocjacji za pomocą identyfikatorów (zamiast referencji) jest chęć uniezależnienia obiektów od siebie (z punktu widzenia JVM).
- N** W przypadku implementacji ekstensji klasy w ramach klasy zewnętrznej, odpowiedni kontener musi być oznaczony słowem static.
- T** Dobre GUI, w celu pokazania wykluczających się opcji wykorzystuje "radio buttons".
- T** Niektóre nowe wersje aplikacji nie koncentrują się na dodawaniu nowych funkcji, ale na ułatwieniu dostępu do już istniejących.

Źródło: <http://www.mtrzaska.com/mas-egzamin>

2. Dla podanego poniżej diagramu analitycznego proszę nazwać elementy oznaczone literami od **A** do **J** oraz krótko opisać sposoby ich implementacji (maks. 30 pkt.):



<b>A</b>	Wyliczalny Java: Settery, gettery C#: Property	<b>F</b>	Atrybut unialny, obowiązkowy [1] Metoda przechowująca nazwy Przeglądanie ekstensji
<b>B</b>	Atrybut klasowy Słowo kluczowe static	<b>G</b>	Asocjacja wiele-wiele Metody manipulujące powiazaniami sprawdza ograniczenie
<b>C</b>	Overlapping Do jednej klasy i dyskryminator	<b>H</b>	Klasa abstrakcyjna Uzywamy słowa kluczowego Abstract
<b>D</b>	Powtarzalny Kolekcja, w konstruktorze trzeba dodac jeden element [1..*]	<b>I</b>	Metoda abstrakcyjna Trzeba zaimplementowac w podklasach
<b>E</b>	Klasa asocjacji Klasa posrednicza	<b>J</b>	Agregacja rekurencyjna Dwa kontenery/kolekcje w ramach tej samej klasy