

# Modelowanie i Analiza Systemów informacyjnych (MAS)

## Egzamin

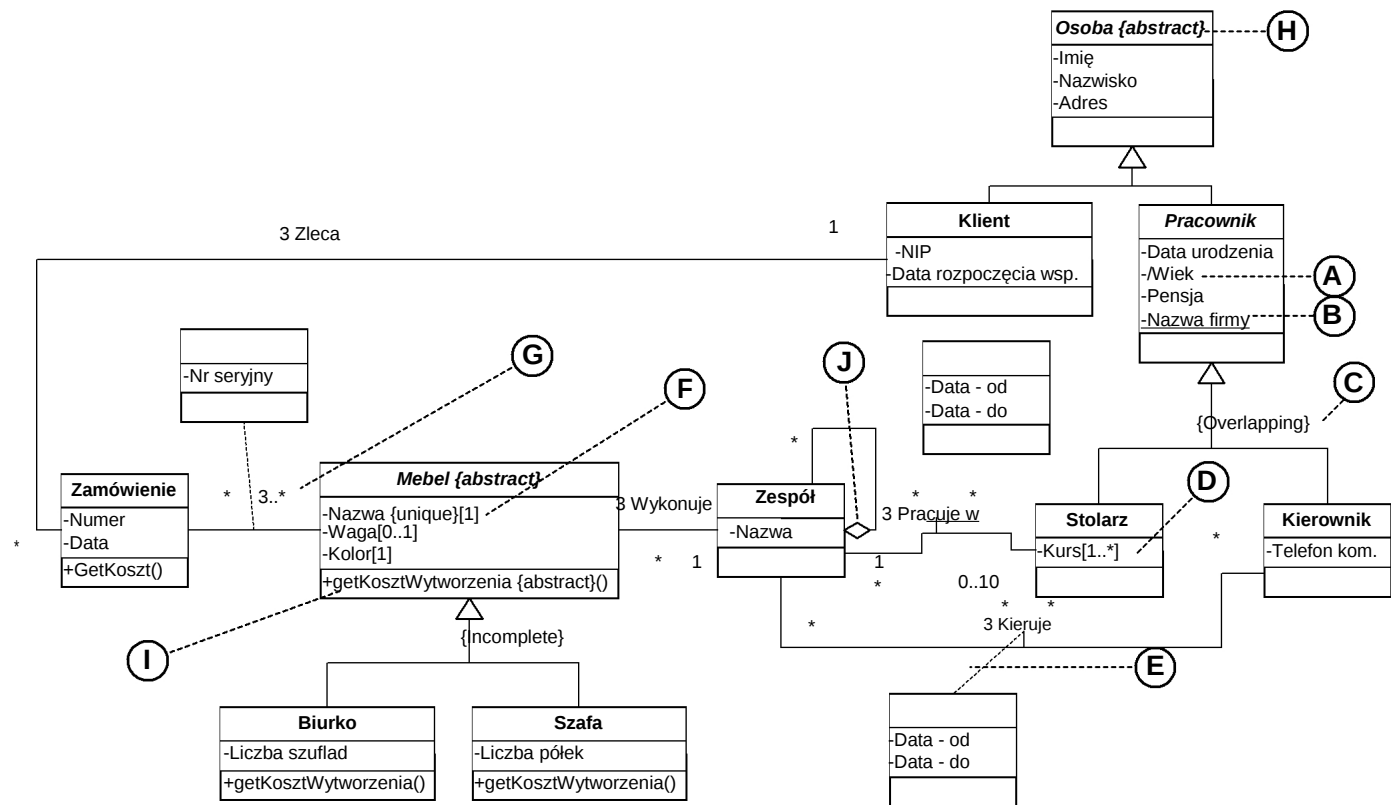
Imię					Nazwisko					Nr indeksu			Grupa															
T	R	O	L	O	L	O				P	W	N	A	G	E					U	M	A	D	?	O	R	L	Y

### Zestaw B

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.).
- 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. **N**
  - 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. **T**
  - 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. **T**
  - Zjawisko określane jako niezgodność impedancji jest związane ze słabą wydajnością relacyjnych baz danych. **N**
  - Plik mapujący wykorzystywany w technologii Hibernate zawiera m. in. informacje dotyczące asocjacji mapowanej klasy. **T**
  - Asocjację 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. **T**
  - Agregacja z modelu obiektowego, jest implementowana w modelu relacyjnym jako dwie tabele pośredniczące i 4 klucze obce. **N**
  - 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). **T**
  - 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. **T**

Źródło: [http://pjwstk.mykhi.org/6sem/MAS/MAS\\_e\\_2011\\_grB\\_odp.pdf](http://pjwstk.mykhi.org/6sem/MAS/MAS_e_2011_grB_odp.pdf)

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



<p><b>A</b> <b>Atrybut wyliczalny.</b></p> <p>Dostęp do niego odbywa w oparciu o metody settery i gettery. Specjalne traktowanie atrybutu zaimplementowane jest w ciele metody.</p>	<p><b>F</b> <b>Atrybut unikalny.</b></p> <p>W konstruktorze Meble sprawdzamy czy w jej ekstensji jest już obiekt z daną nazwą. Dobrze stosować HashMap, gdyż szybko przeszukuje po kluczach unikalnych.</p>
<p><b>B</b> <b>Atrybut klasowy.</b></p> <p>public static String getNazwa_firmy()</p>	<p><b>G</b> <b>Liczności:</b></p> <p>Jak wiele -&gt; to np. ArrayList, jak 1 -&gt; to referencja do jednego obiektu, jak więcej niż jeden ArrayList w konstruktorze, to musimy zapewnić, że przynajmniej 3 obiekty będą w liście</p>
<p><b>C</b> <b>Dziedziczenie typu overlapping.</b></p> <p>Sposoby obejścia: zastąpienie całej hierarchii dziedziczenia jedną klasą wykorzystanie agregacji lub kompozycji</p>	<p><b>H</b> <b>Klasa abstrakcyjna.</b></p> <p>Nie ma bezpośrednich wystąpień obiektów. class abstract Osoba{ }</p>
<p><b>D</b> <b>Atrybut powtarzalny.</b></p> <p>Wiele wartości dla takiego atrybutu przechowujemy w jakimś kontenerze lub zwykłej tablicy. Rodzaj wybranego kontenera może zależeć od sposobu pracy z takim atrybutem, np. czy częściej dodajemy elementy, czy może raczej odczytujemy.</p>	<p><b>I</b> <b>Metoda abstrakcyjna</b></p> <p>Nie zawiera ciała metody w klasie abstrakcyjnej tylko w klasie konkretnej. public abstract double getKosztWytworzenia();</p>
<p><b>E</b> <b>Asocjacja z atrybutem.</b></p> <p>Zamieniamy na asocjację z klasą pośredniczącą. Wadą takiego podejścia jest utrudniony dostęp do obiektów docelowych. Można zaimplementować metody pomocnicze, które po wywołaniu z klasy pośredniczącej zwrócą nam obiekt docelowy.</p>	<p><b>J</b> <b>Agregacja.</b></p> <p>Agregacje implementujemy dokładnie tak samo jak klasyczne asocjacje</p>