

Wstęp

Dokument zawiera podstawowe informacje na temat ćwiczeń MAS.

Organizacja zajęć

1. Zajęcia prowadzimy stacjonarnie jednak nie ma problemu, aby wykorzystać elementy współpracy zdalnej. Mini-projekty oraz projekt końcowy można oddawać w formie zdalnej po wcześniejszym umówieniu się ze mną na taki odbiór
2. Zajęcia MAS skupione są na pracy samodzielnej. Ze względu na ograniczony czas i charakterystykę przedmiotu każde ćwiczenia skupione są na tym, aby mieli Państwo możliwość realizować mini-projekty oraz elementy projektu końcowego. Poza wiedzą zdobytą na wykładach będę przed każdym MP dokładnie omawiał jego zakres oraz informował Państwa o szczegółach oceny, aby mogli się Państwo lepiej przygotować
3. Podstawową formą komunikacji jest platforma MS Teams

Zaliczenie ćwiczeń

1. Dokładne wytyczne co do zasad zaliczenia (punktacje, zasady oceniania) znajdują Państwo na stronie mtrzaska.com w sekcji MAS informacje. W skrócie:
 - a. Należy zaliczyć mini-projekty tj. Uzyskać **ze wszystkich** minimum 50 punktów
 - b. Należy zaliczyć dokumentację na minimum 50 punktów
 - c. Należy zaliczyć projekt końcowy na minimum 50 punktów

Każdy z powyższych warunków MUSI być spełniony, aby móc zaliczyć ćwiczenia

2. Dokładne wytyczne co do zasad punktowania, elementów punktowanych znajdują się w dokumencie dr. Trzaski i obowiązuje wszystkie grupy niezależnie od prowadzącego
3. Ze względu na brak pierwszych zajęć **nie obowiązuje** państwa harmonogram przedstawiony w dokumencie dr. Trzaski. Terminy odbiorów itp. Ustalimy wspólnie

Mini-projekt 1

Pierwszy mini projekt obejmuje dwa podstawowe zagadnienia. Ekstensje wraz z trwałością oraz implementację pól. Ekstensje oraz trwałość przedstawione są na przygotowanym przeze mnie materiale wideo. Znajdą go Państwo w sekcji „Pliki” w grupie na Teams.

Atrybuty omawiam poniżej:

- * atrybut złożony - implementacja odbywa się za pomocą nowej klasy składającej się z plus. „Złożony” w nazwie oznacza, że ten atrybut nie jest opisywany przez pojedynczą wartość jak *int* czy *double* tylko przez zestaw atrybutów zawierających się w klasie. Przykład implementacji:

```
private Person person;
```

Gdzie Person to klasa:

```
public class Person {  
    private Long id;  
    private String firstName;  
    private String lastName;  
}
```

- * atrybut opcjonalny - atrybut który **nie jest** niezbędny/wymagany do utworzenia obiektu danej klasy. Wymagalność atrybutu może być definiowana po przez umieszczenie go w konstruktorze. W przypadku atrybutu opcjonalnego albo nie umieszczamy go w konstruktorze, a jeśli musimy to zrobić lub potrzebujemy to tworzymy drugi konstruktor BEZ tego atrybutu. W ten sposób zachowujemy opcjonalność. Proszę jednak pamiętać, że opcjonalność należy jeszcze obsłużyć. Brak wartości może generować problemy np. Jeśli o tym zapomnimy to możemy natrafić na wyjątek

- * Atrybut powtarzalny - to nic innego jak kolekcja. Wystarczy zastosować dowolny znany Państwu kontener. W przypadku JAVA może to być *List*, *Set*, *Map* i inne. W przypadku C# może to być np. *Dictionary*
- * atrybut klasowy - ten rodzaj atrybutu charakteryzuje się tym, że jego wartość *jest taka sama dla każdego wystąpienia obiektu danej klasy*. Oznacza to, że jego wartość jest niezależna od stanu obiektu. Możemy zaimplementować to z wykorzystaniem słowa kluczowego *static* lub przez wykorzystanie dodatkowej klasy, która będzie odpowiadać za przechowanie wspólnej wartości dla wszystkich wystąpień.
- * Atrybut pochodny - inaczej zwany wyliczalnym. Jego nazwa wynika z tego, że wartość wyliczana jest w momencie pobrania na podstawie aktualnego stanu obiektu. Realnie atrybut nie istnieje, a jedyne co robimy to tworzymy metodę dostępową. Przykład implementacji w JAVA:

```
private float getPrice() {
    return priceNetto * 1.22;
}
```

Przykład realizacji w C#:

```
private float price {
    get {
        return priceNetto * 1.22;
    }
}
```

Kluczowe tutaj jest to, aby wartość atrybutu była ZAWSZE wyliczana w momencie pobierania dlatego nie definiują Państwo typowego pola ani nie umieszczają Państwo tego atrybutu w kontrolerze. Nie ma też potrzeby tworzenia metody `setPrice(float priceNetto)`;

- * metoda klasowa - odpowiednik pola klasowego, ale dla metody
 - * Przeciążenie - implementacja tej samej metody w różnych wariantach. Przykład w JAVA:
- ```
public float getPrice() {
 return price;
}
public float getPrice(float vat) {
 return price * (1.0f + vat / 100.0f);
}
```
- \* Przesłonięcie - nadpisanie metody z nadklasy w podklasie. Idealnym przykładem jest metoda `toString()` którą mogą Państwo implementować w swoich klasach, a które nadpisują implementację tej metody z np. Klasy *Object*.

## Zasady odbioru

Każdy mini-projekt musi zostać „obroniony” przez studenta tzn. Musi on/ona zaprezentować kod, omówić każdy z punktów oraz wyjaśnić sposób implementacji. Z mojej strony często może paść pytanie „dlaczego tak to zostało zaimplementowane?”. Proszę nie traktować tego jako sugestii popełnienia błędu. Kluczowe przy odbiorach jest, aby w pełni Państwo rozumieli co i dlaczego implementują. Umiejętność wyjaśnienia własnych decyzji w oparciu o to co się dzieje na Państwa projekcie oraz w kodzie może mieć znaczący wpływ na ocenę.

Odbiory mogą odbywać się osobiście na terenie uczelni lub zdalnie za pomocą MS Teams. W drugim wariancie zdzwaniamy się, Państwo udostępniają ekran i przechodzimy przez cały kod. Od razu po odbiorze otrzymują Państwo ilość punktów, a więc cała wiedza na temat progresu jest dla Państwa jawna.

## Projekt końcowy

Wytyczne co do złożoności i realizacji projektu końcowego znajdują Państwo w dokumencie dr. Trzaski. Ze swojej strony dodam, że kluczowa jest użyteczność biznesowa systemu. Sugeruję

unikać systemów, które są narzędziem do wprowadzenia danych (nic poza formularzami). Państwa zadaniem jest stworzenie nietrywialnego przypadku użycia, które obejmuje konkretny proces biznesowy. Przykłady takich procesów:

- \* proces zakupowy w sklepie
- \* Proces rezerwacji
- \* Proces wynajmu

Jak widzą Państwo te procesy charakteryzują się pewną złożonością i potrzebą stworzenia zależności między różnymi klasami. Oczywiście każdy przypadek użycia można ze mną skonsultować do czego gorąco zachęcam.