|  |
| --- |
| Merito University |
| Programowanie Obiektowe |
| Ćwiczenia 2 - zadania |

|  |
| --- |
| Autor: Lesław Pawlaczyk  2025/06/15 |

Spis treści

[Rozdział 1 – Zadania do zrealizowania 2](#_Toc132844718)

[Rozdział 2 – Ocenianie 3](#_Toc132844719)

[Bibliografia 4](#_Toc132844720)

# Rozdział 1 – Zadania do zrealizowania

**Zadania podstawowe**:

1. Dla rozwiązania zadań nr 1 oraz 2 z Ćwiczeń 1 należy
2. Dodać diagram klas dla interfejsów z bilbioteki SampleHierachies.Interfaces.
3. Dla rozwiązania zadań nr 1 oraz 2 z Ćwiczeń 1 należy dodać diagram klas i połączeń z interfejsami dla biblioteki SampleHierachies.Data
4. Dla rozwiązania zadań nr 1 oraz 2 z Ćwiczeń nr 1 dodać diagram użycia (Use Case Diagram) dla jednego z zaimplementowanych ekranów.
5. Dodać rozwiązania zadań nr 1 oraz 2 Ćwiczeń nr 1 dodać diagram sekwencji wywołań (Sequence Diagram) dla renderowania jednego z zaimplementowanych ekranów.

**Zadania dodatkowe**:

1. Dla zadań nr 1 oraz 2 z Ćwiczeń 1 dodać diagram komponentów dla całego rozwiązania. Diagram powinien być ogólny bez nadmiarowych szczegółów.
2. Dla rozwiązania zadań na ocenę dst z Ćwiczeń nr 2 dodać diagram komponentów całego rozwiązania. Diagram powinien być ogólny bez nadmiarowych szczegółów.
3. Dla rozwiązania zadań na ocenę dst z Ćwiczeń nr 2 dodać diagram przypadków użycia dla klasy ScreenDefinitionService.
4. Dla rozwiązania zadań na ocenę dst z Ćwiczeń nr 2 dodać diagram sekwencji dla wybranej metody klasy SettingsService.

Do tworzenia diagramów można posłużyć się np.:

1. <https://www.figma.com/figjam/diagramming-tool/> (tutorial użycia: <https://www.youtube.com/watch?v=4NzlbkcgQqY&ab_channel=Figma>)
2. <https://draw.io> (tutorial: <https://www.youtube.com/results?search_query=how+to+draw+uml+diagrams+in+draw.io>)
3. lub dowolnym innym wybranym narzędziem (może nawet być MsPaint).

# Rozdział 2 – Ocenianie

**Rozwiązania powinny być indywidualne**.

1. Wykonać zadania od 1 do 4 na ocenę co najmniej dst.
2. Wynik w postaci linku do repozytorium na GitHub oraz sprawozdanie w PDF. Sprawozdanie w PDF powinno się nazywać: ProgramowanieObiektowe\_Zadanie3\_20230420\_Rozwiazanie\_ImieNazwisko\_XXXXX.pdf, gdzie ImieNazwisko zastąpić swoim imieniem i nazwiskiem a XXXXX numerem albumu.
3. Rozwiązanie powinno być linkiem do repo na GitHub, gdzie będzie pokazane PDFy albo BMP w wysokiej rozdzielczości z diagramami i wskazaniami w nazwie, którego zadania dotyczą.

**Strona tytułowa**:

1. Imię i nazwisko.
2. Adres email.
3. Numer albumu.
4. Data.
5. Link do repo na Github.

**Sposób oceniania**:

1. Ocena 3.0 – zadania nr 1 do 4
2. Ocena 3.5 – zadania nr 1 do 5.
3. Ocena 4.0 – zadania nr 1 do 6.
4. Ocena 4.5 – zadania nr 1 do 7.
5. Ocena 5.0 – zadania nr 1 do 8.

**Oceniane będą**:

1. Jakość kodu.
2. Komentarze.
3. Poprawność działania programu.

**Czas do wykonania**:

1. Do 2025/06/15.

**Uwaga:**

Przy zauważeniu, że kod powtarza się z kodem innego studenta rozwiązanie zostanie uznane za plagiat i ocenione na ocenę „ndst”.

# Bibliografia

1. <https://github.com/palles77/ProgrammingObjectOriented/raw/main/UmlTutorial.zip>
2. <https://github.com/palles77/ProgrammingObjectOriented/raw/main/UmlSpecification.pdf>
3. <https://uml.org>
4. <https://draw.io>
5. <https://www.microsoft.com/en-ww/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>
6. <https://www.modelio.org/index.htm>
7. <https://www.softwaretestinghelp.com/uml-diagram-tutorial/>