Opracowanie: dr inż. Łukasz Maliński, Politechnika Śląska, 2022

LABORATORIUM 6.2
REFAKTORYZACJA KODU

Poruszane zagadnienia z dziedziny programowania:

- Przekształcenia refaktoryzacyjne - **wyk. 17**
- Trójwarstwowy model aplikacji - **wyk. 16**
- Testowanie GUI - **wyk. 17**

Umiejętności do opanowania:

- dokonać refaktoryzacji zadanego kodu zgodnie z wytycznymi,
- napisać testy jednostkowe dla elementów GUI,
- dostosować strukturę własnej aplikacji do zadanego modelu warstwowego.

Oznaczenia odnośnie samodzielności pracy:

■ – **Ten problem koniecznie rozwiąż w pełni samodzielnie.** Możesz posługiwać się literaturą, wykładami i dokumentacją w celu sprawdzenia niuansów składniowych, ale nie szukaj gotowych rozwiązań i algorytmów. **Jest to bardzo ważne z punktu widzenia nauki i oszukiwanie przyniesie tylko poważne braki w późniejszych etapach nauki!**

▲ – Rozwiązując ten problem możesz posiłkować się rozwiązaniami zapożyczonymi od innych programistów, **ale koniecznie udokumentuj w kodzie źródło.** Nie kopiuj kodu bezmyślnie, tylko dostosuj go do kontekstu zadania. **Koniecznie musisz dokładnie zrozumieć rozwiązanie, które adoptujesz, gdyż inaczej wiele się nie nauczysz!**

● – Rozwianie tego problemu właśnie polega na znalezieniu i użyciu gotowego rozwiązania, **ale koniecznie podaj źródło.** Nie będzie wielkiej tragedii jeśli wykażesz się tylko ograniczonym zrozumieniem rozwiązania, gdyż nie musi być ono trywialne.

Zadanie domowe (ocena uwzględniana tylko po uzyskaniu 1,0 pkt. na zajęciach) [2,0 pkt.]

UWAGA: zaktualizuj repozytorium o nazwie Pamiętnik.

■ Przebuduj aplikację Pamiętnika, tak aby była zorganizowana zgodnie z modelem trójwarstwowym. Wyraźnie wydziel klasy GUI (warstwa prezentacji), klasę usług (warstwa usług), i klasy/funkcje przetwarzające dane (warstwa danych).

Wskazówki: Zadbaj aby klasy GUI zawierały tylko funkcjonalności faktycznie związane z prezentacją danych i zadawaniem akcji. Wydziel typowe „zlecenia" jaki zleca użytkownik i umieść je w formie metod w klasie warstwy usług.

Na przykład: zlecenie dodania nowych wpisów powinno wywoływać usługę (metodę) w klasie warstwy usług, która to powinna otworzyć nowe okno do wpisywania wpisów i odebrać z niego dane. Następnie warstwa usługo powinna dokonać aktualizacji zbioru wspomnień w warstwie danych.

Dla kontrastu, przewijanie wpisów w oknie przy pomocy klawiszy klawiatury, jest funkcjonalnością czysto związaną z prezentacją, więc nawet nie powinno na siłę z tej warstwy „wychodzić".

Funkcjonalności typowo back-end'owe, jak sortowanie, zapisywanie do pliku itd, powinny być w pełni obsłużone w warstwie danych.

■ Trzymając się architektury trójwarstwowej dodaj dwie nowe funkcjonalności:

1) Tworzenie kopii zapasowej zbioru wspomnień. Należy dać możliwość zapisu aktualnego zbioru wspomnień w osobnym pliku (innym niż plik domyślny), oraz możliwość odtworzenia zbioru wpisów na bazie wskazanego pliku.
2) Klasyfikacja wspomnień na *smutne*, *neutralne* i *radosne*, w raz z możliwością ustawienia, które z nich mają być wyświetlane (3 niezależne CheckBoxy). Selekcja wspomnień powinna być zadawana w warstwie prezentacji, ale dokonywana w warstwie danych.

Jeśli powyższe funkcjonalności zostały już wcześniej dodane jako własne propozycje (poprzednie zajęcia), to zaproponuj i zaimplementuj tutaj inne. Zadbaj aby funkcjonalności nie były czysto kosmetyczne i wymagały zaangażowania wszystkich trzech warstw.

Wskazówki: Zauważ, że nowe funkcjonalności są tak dobrane, aby angażować wszystkie 3 warstwy aplikacji. Przyjmij zasadę, że każda taka funkcjonalność, której działanie wykracza poza warstwę prezentacji powinna mieć reprezentację w warstwie usług.

W warstwie danych dobrze jest stworzyć dwie listy wspomnień: pełną i tą do wyświetlenia (może być węższa). Zastanów się jak to zrobić by niepotrzebnie nie duplikować danych.

■ Napisz test jednostkowy sprawdzający czy zaznaczenie jednocześnie klas smutne i radosne prawidłowo odfiltruje wspomnienia neutralne.

Wskazówki: Przygotuj na potrzeby testu bazę 9 wspomnień, po trzy na klasę. Upewnij się że w chwili testowania zbiór testowy znajduje się w warstwie danych. Zasymuluj programowo odhaczenie 2 checkboxów zadanych klas wspomnień. Odczytaj jaką listę wspomnień otrzyma warstwa prezentacji do wyświetlenia i sprawdź czy będzie zgodna z oczekiwaniami.

Pamiętaj, że testy są szczególnym typem funkcji/metod, które często muszą mieć dostęp do składników normalnie nie dostępnych. Nie krępuj się dla nich używać relacji przyjaźni. Staraj się nie korzystać funkcjonalności programu, które nie są testowane.

Dla przykładu można by pokusić się os tworzenie pliku zawierającego specjalne testowe wpisy i w czasie testów podmieniać go jako źródło danych. Problem jednak jest taki, że test filtracji wspomnień będzie wtedy zależny od funkcjonalności wczytywania wspomnień z pliku (zasada mówi że testy powinny być nie zależne i powinny testować jedną funkcjonalność na raz).

Ocenianie:

Przebudowa do 3W:	**0,5 pkt.**
Kopia zapasowa:	**0,5 pkt.**
Klasyfikacja wspomnień:	0,5 pkt.
Test jednostkowy:	0,5 pkt.

(**cele główne – wytłuszczone –** muszą być ocenione na min. 0,3 pkt.,
aby pozostałe cele cząstkowe były oceniane).