Laboratorium 4 – **implementacja**

**Polecane materiały w zakresie przygotowania do zajęć:**

- repozytorium, system kontroli wersji:
  - https://git-scm.com/book/pl/v2/Pierwsze-kroki-Wprowadzenie-do-kontroli-wersji
  - http://pbiecek.github.io/Przewodnik/Programowanie/jak_korzystac_z_serwisu_github_i_waffle.html
  - https://docs.microsoft.com/pl-pl/visualstudio/ide/git-with-visual-studio?view=vs-2019
- testy jednostkowe:
  - https://helion.pl/ksiazki/testy-jednostkowe-zasady-praktyki-i-wzorce-vladimir-khorikov,tejeza.htm#format/d
  - https://docs.microsoft.com/pl-pl/visualstudio/test/writing-unit-tests-for-c-cpp?view=vs-2019
  - https://www.samouczekprogramisty.pl/testy-jednostkowe-z-junit/
  - https://javastart.pl/baza-wiedzy/testowanie-jednostkowe/junit
  - https://kobietydokodu.pl/17-testy-jednostkowe/
- TDD:
  - https://blog.onwelo.pl/czym-jest-technika-tdd-i-jak-wyglada-jej-cykl-zycia/
  - https://kobietydokodu.pl/niezbednik-juniora-test-driven-development/
  - https://www.samouczekprogramisty.pl/test-driven-development-na-przykladzie/
  - https://www.modestprogrammer.pl/test-driven-development-korzysci-ze-stosowania-tdd
  - https://helion.pl/ksiazki/tdd-sztuka-tworzenia-dobrego-kodu-kent-beck,tddszv.htm#format/e

**Przebieg zajęć:**

- Ocena diagramów sekwencji.
- **Rozpoczęcie (lub kontynuacja) implementacji projektu.** Prace nadal w zespołach 3-osobowych, które zostały ustalone na początku semestru. Temat projektu bez zmian. **Implementacja powinna być zgodna z wynikiem wcześniejszych analiz** w zakresie inżynierii wymagań, projektu struktury obiektowej (klas) oraz interakcji (pomiędzy obiektami).
- Dobór języka implementacji - wymaga się **obiektowego języka programowania** (C++/Java). Dobór środowiska programistycznego (IDE).
- Utworzenie konta w serwisie GitHub, instalacja Git'a. Integracja z wykorzystywanym środowiskiem IDE.
- **Wykorzystanie systemu kontroli wersji w projekcie**: gałęzie, zmiany, wersje, przesyłanie zmian pomiędzy gałęziami, scalanie zmian.
- Dobór narzędzia do testów jednostkowych, integracja z IDE. **Tworzenie testów jednostkowych**. Zastosowanie TDD (włączając refaktoryzację). Dokumentacja poszczególnych cykli TDD z wykorzystaniem repozytorium.

**Informacje o ocenianiu:**

- zgodnie z harmonogramem zajęć ocenianie prac w zakresie implementacji odbędzie się podczas ćwiczeń laboratoryjnych numer 5 (kolejne spotkanie). Ocena będzie w zakresie od zera do 40 punktów. Szczegóły oceniania ukazane są w pliku **Harmonogram oraz zasady oceniania**, który umieszczony jest w kursie UPEL.

**Przed rozpoczęciem zajęć nr 5 należy umieścić w odpowiednim zadaniu kursu następujące pliki:**

- **info.pdf** zawierający podstawowe dane: temat (tytuł) projektu, adres URL repozytorium (np. https://github.com/mikeal/watch), skład zespołu, nr grupy dziekanatowej,
- ***.vpp**, czyli projekt w Visual Paradigm, który zawiera wszystkie wymagane w trakcie semestru diagramy w ostatecznych wersjach,
- ***.zip** zawierający kody źródłowe w ostatecznej wersji (pliki z kodem np. *.java, *.cpp, *.h) oraz testy jednostkowe.
- sprawozdanie.pdf, który będzie odnosił się do prac wykonanych w trakcie implementacji. W pliku tym należy zawrzeć następujące cześći:
  - **język implementacji i IDE** - informacje o wybranym języku programowania, IDE, czy ewentualnie innych wykorzystywanych narzędziach (framework'ach, bibliotekach). Krótkie uzasadnienie decyzji w tym zakresie.
  - **wykorzystanie repozytorium** - informacje o wykorzystaniu repozytorium, czyli wykonywanych commit'ach, tworzonych branch'ach, pull requist'ach, scalaniu zmian (gałęzi), code review. Opisowi powinny towarzyszyć rysunki, czyli fragmenty zrzutów ekranu (strona projektu w witrynie GitHub) potwierdzających wykonanie operacji.
  - **informacje o zaimplementowanych testach jednostkowych** - opis prac i ich efektów. Ile testów? Czy dotyczą całego projektu, czy część? Narzędzia, które były wykorzystywane, etc.
  - **TDD** - informacje dotyczące zastosowania TDD. Czy w ogóle było, czy nie było wykorzystane to podejście? Jeśli było, to przedstawić poszczególne cykle łącznie z refektoryzacją. Wskazać części implementacji, w której był zastosowany TDD. Ukazać odniesienie w historii repozytorium.

**Powyższe części powinny być pisane zwięźle i konkretnie, ale bez pomijania istotnych szczegółów.**