

Program (projekt)

75 punktów można zdobyć za aplikację zaprojektowaną z użyciem wzorców projektowych, w tym:

- 40 punktów można zdobyć za **projekt** aplikacji, prezentowany około 11. zajęć publicznie na zajęciach (maks. 8 wzorców x 5 pkt. za wzorzec - należy użyć co najmniej 1 wzorzec kreacyjny, strukturalny i czynnościowy); należy zaprezentować główne funkcjonalności aplikacji, diagram klas i opis użycia wzorców w formie prezentacji ładowanej potem w formacie PDF do Moodle; Uwaga: zamiast jednego wzorca można zaproponować własną propozycję niebanalnego rozwiązania problemu z użyciem standardowych technik programowania obiektowego (dziedziczenie, kompozycja, metody wirtualne, ...) - coś w rodzaju "własnego wzorca" Użycie 1 wzorca architektonicznego warte jest 10 punktów.
Uwaga na wzorzec Singleton: będzie sprawdzane, czy nie jest on używany do uzyskania globalnego dostępu do obiektu !
Dużo wzorców procentuje w tej części oceny ale może wygenerować dużo punktów ujemnych w drugiej (brak implementacji wzorca, błędy w zastosowaniu wzorców, nieuwzględnione uwagi dotyczące użycia).

Minimalnie trzeba zdobyć 15 punktów za projekt (diagram) klas

- 35 punktów można zdobyć za implementację projektu, w tym:
 - 20 pkt. za poprawność implementacji (bezbłędne działanie, zgodność z projektem, uwzględnienie uwag z projektu)
 - 5 pkt. za tzw. "przyjazność użycia" (prowadzący zajęcia może spróbować obsłużyć go samodzielnie, wpisując też niepoprawne dane) i jakość kodu (obniżenie punktacji: za "niesamoopisujący" się kod - bez komentarzy, z niepoprawnymi nazwami klas, metod, zmiennych; wielopiętrowe warunki 'if')
 - 10 pkt. za jakość dokumentacji (wg wymagań poniżej)

Zwłoka skutkuje karą 3 pkt za każdy dzień zwłoki.

Minimalnie trzeba zdobyć 20 punktów za implementację projektu.

Programy realizowane są w grupach 2-3 osobowych - NIE MA GRUP 1-OSOBOWYCH !!!

Język implementacji jest dowolny.

Aplikację w formie archiwum (zawierającego również źródła) wraz z elektroniczną wersją dokumentacji należy załadować do Moodle (lub innego wskazanego repozytorium) do dnia przedostatnich zajęć.

Programy będą prezentowane i oceniane na podstawie załadowanej wersji na 2 ostatnich zajęciach.

Wymagania dotyczące dokumentacji:

W dokumentacji należy zawrzeć :

1. Diagram klas budujących użyte wzorce (można go podzielić na części jeśli jest zbyt duży do prezentacji w całości). Można (czasem należy) pominąć klasy, które nie są istotne do przedstawienia wzorców (np. klasy interfejsu).
2. Krótki opis każdego wzorca obejmujący co najmniej:
 1. cel użycia;
 2. przyporządkowanie klas do ról wzorca;
 3. lokalizację wzorca w kodzie (pakiet, plik, ewentualnie numer linii jeśli plik jest długi i zawiera inne komponenty), zarówno jego definicję jak i użycie;
i nieobejmujący teoretycznego opisu wzorca.
3. Opis co najmniej jednego rozwiązania specyficznego dla użytej technologii i podać wykorzystane źródło wiedzy na ten temat (**EK4** - brak tego elementu powoduje niezaliczenie przedmiotu zajęć !).
4. Opis podziału pracy w zespole - niedopuszczalne są sformułowania typu "wszyscy pracowali równo" lub "podział pracy - po 50 %" - trzeba napisać, co kto konkretnie zrobił (najlepiej kto pisał poszczególne klasy, fragmenty systemu, ...). (**EK5** - brak tego elementu powoduje niezaliczenie przedmiotu zajęć !).
5. Krótką instrukcję użytkownika, góra 1-2 strony - najważniejsze (grupy) funkcjonalności.
6. Jeszcze krótszą instrukcję instalacji.
7. **Analizę możliwości rozbudowy programu z uwzględnieniem użytych wzorców**