

L06 Wzorce projektowe 3

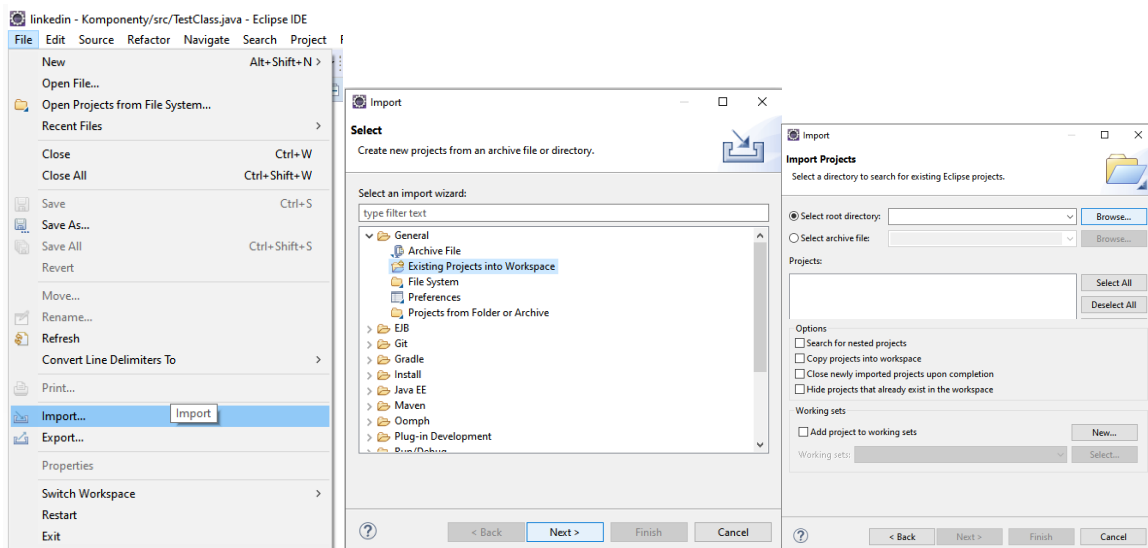
Ćwiczenia laboratoryjne

Cel

Celem ćwiczeń laboratoryjnych jest nabycie umiejętności korzystania ze wzorców projektowych w implementowaniu w języku java prostych przykładów projektów. Ćwiczenia są inspirowane książką „Rusz głową! Wzorce projektowe” wyd. Helion 2011.

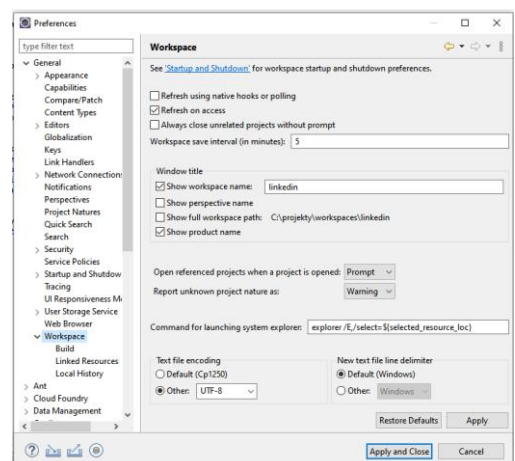
Przygotowanie

Ze strony kursu proszę pobrać spakowany plik z kodem startowym, rozpakować do wybranego katalogu i zaimportować z tego katalogu do Eclipse istniejące projekty: FabrykaCzekolady, SuperPilot.



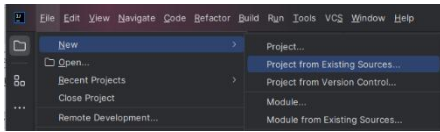
W treści zadań pozostawiono polskie znaki w nazwach klas itp., żeby wygodniej można było posługiwać się tym tekstem. Kod początkowy nie zawiera jednak polskich znaków. Rekomendowane jest, żeby rozwiązanie również nie zawierało polskich znaków.

Gdyby w miejscu polskich znaków pojawiły się krzaki, należy zmienić kodowanie polskich znaków w eclipse. W menu Window opcja Preferences i następnie > General, > Workspace i Text file encoding: UTF-8.



Środowisko IntelliJ

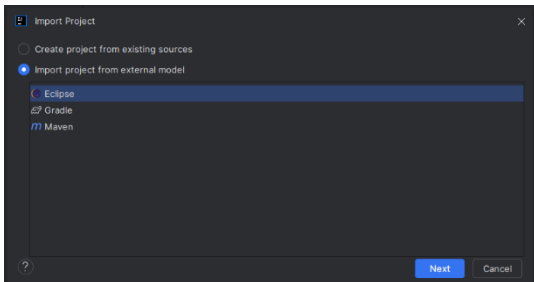
W IntelliJ uruchamianym po raz pierwszy należy utworzyć projekt bez nazwy aby uzyskać dostęp do pełnego menu. Następnie w menu wybrać *File/New/Project from Existing Sources...*.



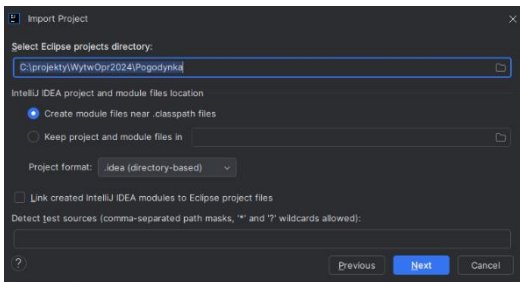
Alternatywnie można od razu, jako pierwszy krok, wybrać tę opcję za pomocą skrótu: *Ctrl+Shift+A*.

Potem wybrać katalog z projektem.

Następnie w oknie dialogowym wybrać opcję *Import project from external model* i *Eclipse* jako źródło projektu.



Kolejnym krokiem jest zaimportowanie projektu wraz z ustawieniami.

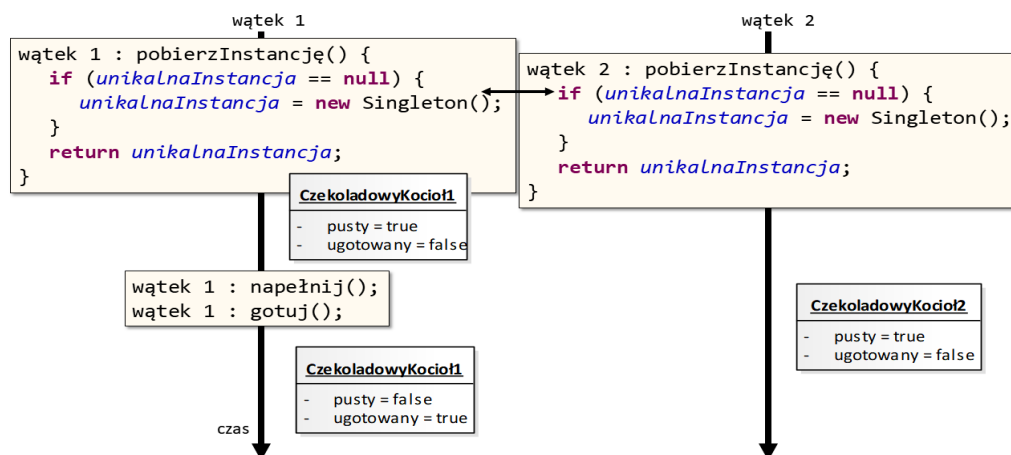


W kolejnych oknach dialogowych proszę naciskać *Next* i *Create*.

Sprawozdanie

Jako wynik pracy na zajęciach proszę przesłać pojedyncze archiwum w formacie „zip” zawierające projekty Eclipse’a/IntelliJ z rozwiązanymi zadaniami – czyli wystarczająco pliki źródłowe.

Zadanie 1 Fabryka Czekolady



Proszę zaimplementować w projekcie FabrykaCzekolady wzorec projektowy Singleton tak, aby klasa CzekoladowyKocioł miała tylko jedną instancję na różne sposoby przedstawione na wykładzie:

1. Proszę skopiować klasę CzekoladowyKocioł, FabrykaCzekolady i FabrykaCzekoladyWątki z pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady` do pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady.singleton.klasyczny`. Należy zaimplementować w klasie CzekoladowyKocioł wzorec singleton w wersji klasycznej, usunąć obecnie istniejącą klasę Singleton, a następnie przetestować za pomocą aplikacji FabrykaCzekolady oraz FabrykaCzekoladyWątki.
2. Proszę skopiować klasę CzekoladowyKocioł, FabrykaCzekolady i FabrykaCzekoladyWątki z pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady` do pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady.singleton.synchro`. Należy zaimplementować w klasie CzekoladowyKocioł wzorec singleton w wersji synchronizowanej (powolnej), usunąć obecnie istniejącą klasę Singleton, a następnie przetestować za pomocą aplikacji FabrykaCzekolady oraz FabrykaCzekoladyWątki.
3. Proszę skopiować klasę CzekoladowyKocioł, FabrykaCzekolady i FabrykaCzekoladyWątki z pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady` do pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady.singleton.eager` i zaimplementować w klasie CzekoladowyKocioł wzorec singleton w wersji zachłannej (pożerającej zasoby), usunąć obecnie istniejącą klasę Singleton, a następnie przetestować za pomocą aplikacji FabrykaCzekolady oraz FabrykaCzekoladyWątki.
4. Proszę skopiować klasę CzekoladowyKocioł, FabrykaCzekolady i FabrykaCzekoladyWątki z pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady` do pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady.singleton.double_lock` i zaimplementować w klasie CzekoladowyKocioł wzorec singleton w wersji z podwójnym blokowaniem (złożonej), usunąć obecnie istniejącą klasę Singleton, a następnie przetestować za pomocą aplikacji FabrykaCzekolady oraz FabrykaCzekoladyWątki.
5. Proszę skopiować klasę CzekoladowyKocioł, FabrykaCzekolady i FabrykaCzekoladyWątki z pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady` do pakietu `pl.edu.zut.po.wzorcy.fabryka_czekolady.singleton.jbloch` i zaimplementować w klasie CzekoladowyKocioł wzorec singleton w wersji Joshu'y Blocha (wykorzystującej implementację typu wyliczeniowego enum), usunąć obecnie istniejącą klasę Singleton, a następnie przetestować za pomocą aplikacji FabrykaCzekolady oraz FabrykaCzekoladyWątki.

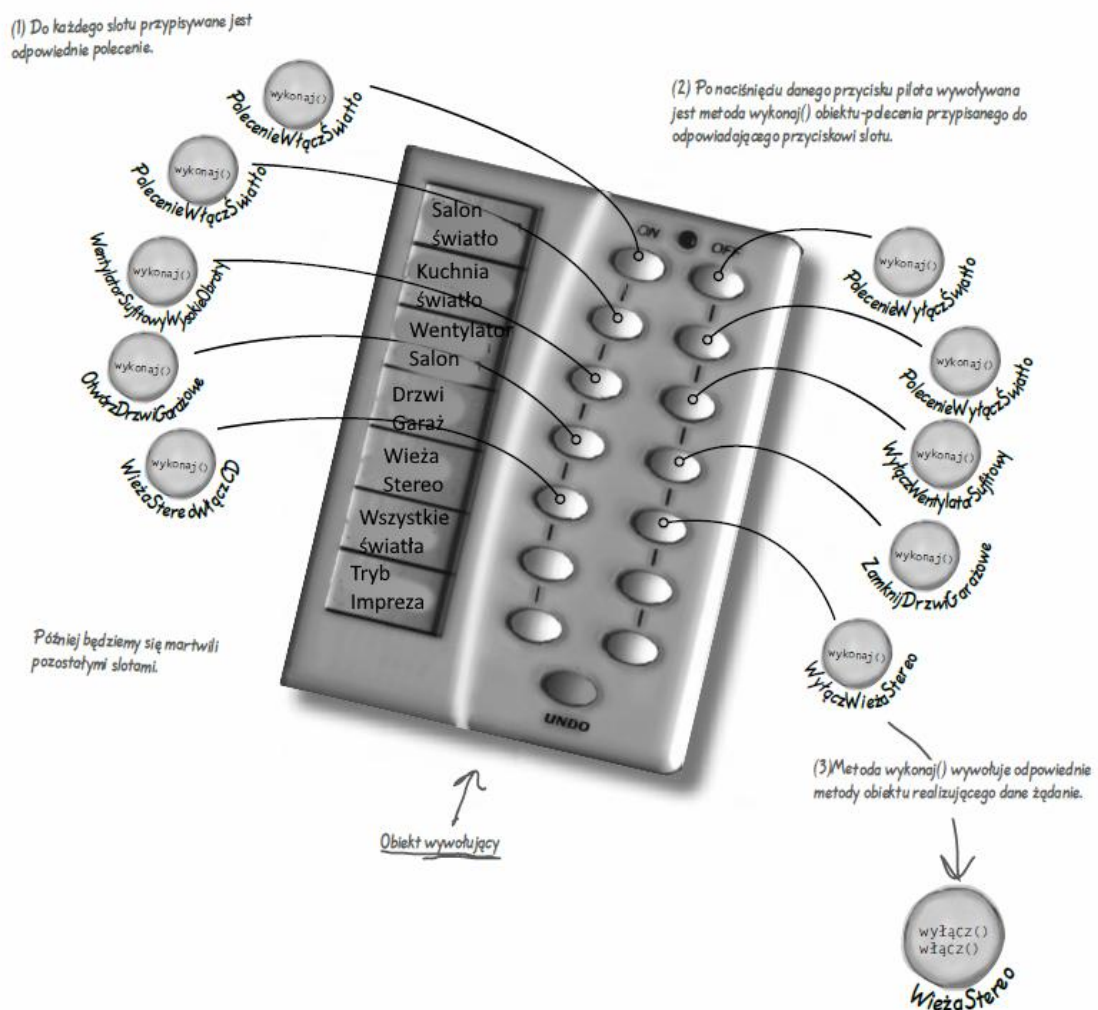
Zadanie 2 SuperPilot - wzorzec Polecenie

Projekt SuperPilot zawiera następujące pakiety:

pl.edu.zut.po.wzorce.pilot – MiniPilot, MiniPilotTest – przykładowy jednoprzyciskowy pilot i jego klasa testująca, SuperPilot, SuperPilotTest – klasy Super Pilota, testy do implementacji.

pl.edu.zut.po.wzorce.pilot.polecenia – Interfejs Polecenie, przykładowe polecenie włączające światło, tu mają znaleźć się polecenia do sterowania urządzeniami.

pl.edu.zut.po.wzorce.pilot.sterowniki – w pakiecie znajdują się klasy sterujące urządzeniami domowymi: drzwiami garażu, jacuzzi, oświetleniem, telewizorem, wentylatorem i wieżą stereo.



Rysunek 1 Polecenia do zaprogramowania w SuperPilocie

1. Proszę zaimplementować zgodnie z wzorcem Polecenie, polecenia włączające i wyłączające urządzenia zgodnie z Rysunkiem 1, wzorując się na klasie `MiniPilotTest` uzupełnić implementację `SuperPilotTest`.
2. Proszę zaimplementować klasę `SuperPilotTestWycofaj` i przetestować możliwość wycofania ostatniego polecenia.
3. Proszę zaimplementować, używając klasy `MakroPolecenie`, klasę testującą `SuperPilotTest` z możliwością włączenia i wyłączenia wszystkich światel w domu oraz sterowania trybem Impreza: przyciemnione oświetlenie, włączona wieża stereo i telewizor, przełączenie całego zestawu na DVD oraz przygotowanie kąpieeli w jacuzzi.