

Temat : *Diagramy klas a implementacja kodu*

## Historia zmian

[illegible]

# 1. Cel laboratorium

Głównym celem laboratoriów jest nauka implementacji programu na bazie diagramów klas. Studenci zapoznają się z ideą wzorców projektowych poprzez użycie wzorców strukturalnych i kreacyjnych w małym projekcie.

*Czas realizacji laboratoriów wynosi 2 godziny.*

## 2. Zasoby

### 2.1. Wymagane oprogramowanie

Polecenia laboratorium będą dotyczyły programowania wzorców w języku Java. Potrzebne będzie środowisko dla programistów (JDK – Java Development Kit<sup>1</sup>) oraz zintegrowana platforma programistyczna (np. Eclipse<sup>2</sup>).

### 2.1. Materiały pomocnicze

Materiały dostępne w Internecie:

<http://idiotechie.com/uml2-class-diagram-in-java/>

<https://www.visual-paradigm.com/tutorials/roundtrip.jsp>

<http://www.vincehuston.org/dp/>

[http://en.wikipedia.org/wiki/Design\\_pattern\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Design_pattern_(computer_science))

## 3. Laboratorium

### 3.1. Zadanie pierwsze - Model dziedziny

Krysia wybiera się do Inverness w UK<sup>3</sup> na koncert swojego ulubionego zespołu. Zapowiada się dłuższa wyprawa, więc postanowiła zabrać ze sobą swoją ulubioną suszarkę. Rysio uświadomił jej, że bez zakupu specjalnej *prześciówki do gniazdka UK* nie będzie mogła jej użyć w trakcie swojej eskapady. Na szczęście Rysio zna sklep, w którym Krysia będzie mogła wyposażyć się w potrzebne urządzenie. Dodatkowo, może skorzystać z promocji i kupić prześciówkę za pół darmo pokazując w sklepie poprawny *diagram klas UML przedstawiający przedmiot zakupu i elementy współpracujące z nim*. Rysio nie umie rysować diagramów klas, ale polecił Krysi Ciebie, żebyś pomógł jej w tym zadaniu. Żeby ułatwić Ci zadanie, opisał prześciówkę dokładnie słownie:

„Urządzenia europejskie i brytyjskie są wyposażone we wtyczki o innych interfejsach. Prześciówka to urządzenie rodzaju UK. Dla urządzeń do niej podłączanych, jest ona po prostu gniazdkiem typu europejskiego.”

Krysia prosi, żebyś na diagramie zaprezentował też jej ulubioną suszarkę.

**Wskazówki:**

- W celu redukcji liczby klas na diagramie pokaż dziedziczenia wyłącznie na klasie reprezentującej prześciówkę i suszarkę.
- Klasy powinny reprezentować obiekty fizycznie (klasa Typ jest niepoprawna).
- Używaj pełnych nazw klas (klasa UK może oznaczać co najwyżej państwo).
- Diagram zacznij od zamodelowania prześciówki, a skończ na klasie suszarki.

---

<sup>1</sup> <http://java.sun.com/javase/downloads/index.jsp>

<sup>2</sup> <http://www.eclipse.org/>

<sup>3</sup> Po głosowaniu Szkotów w sprawie odłączenia od UK Inverness nadal jest częścią UK.

## 3.2. Zadanie drugie – Implementacja przejściówki

Przy zakupie nie dołączono instrukcji do przejściówki. Krysia nie zna za dobrze diagramów klas, ale świetnie pisze i czyta programy w języku Java. Ma kolejną prośbę. Wyjaśnij jej w jaki sposób powinna korzystać z przejściówki uzupełniając kod, który Krysia przygotowała.

Zaimplementowane testy pomogą Ci zweryfikować, czy kod wyjaśniający działa poprawnie. Testy to scenariusze podłączania suszarki w Polsce i UK do gniazdek 230 i 400 volt.

### 3.2.1. Pobranie kodu i przygotowanie do pracy

1. Do nowego folderu pobierz i rozpakuj (patrz opcja „Clone or download” → Download ZIP”) w interfejsie Githuba) strukturę przykładowego projektu z repozytorium svn:  
[https://github.com/iis-io-team/lab\\_hairdryer](https://github.com/iis-io-team/lab_hairdryer)
2. Projekt należy zaimportować do Eclipse IDE wybierając *File* → *Import...* → *General* → *Existing Projects into Workspace*. Następnie należy wybrać katalog jako *Root Directory* i kliknąć *Finish*.
3. Sprawdź czy w „Referenced libraries” projektu jest załadowana biblioteka *pioadalab.jar*.
4. Zapoznaj się z całą strukturą projektu (łącznie z klasami biblioteki *pioadalab.jar*).
5. Projekt powinien też zawierać biblioteki JUnit 4. Uruchom zestaw testów *edu.kis.pio.tests.HairDryerStoriesTest.java* (kliknij prawym przyciskiem na pliku, wybierz opcję *Run As* → JUnit Test).
6. **Przeanalizuj wynik i kody źródłowe testów.** Pytania kontrolne:
  - Jaki interfejs reprezentuje gniazdko typu UK?
  - Jaka metoda urządzenia zwraca jego wtyczkę?
  - W jaki sposób podłącza się wtyczkę do gniazdka?

### 3.2.2. Projekt przejściówki i implementacja

1. Na bazie diagramu klas w pakiecie *edu.kis.pio.triptouk* zaprojektuj nową klasę (bez implementacji metod) reprezentującą przejściówkę. Należy wziąć pod uwagę następujące cechy kodu Krysi:
  - zarówno urządzenia jak i gniazdka nie są klasami, ale interfejsami, w związku z tym relacja dziedziczenia powinna być interpretowana w Javie jako realizacja interfejsu. (Jakie słowo kluczowe Java powinno być użyte?)
  - wtyczka też jest interfejsem, ale możesz znaleźć klasę go implementującą. (Wykorzystaj w klasie przejściówki oba typy, tzn. zarówno interfejs jak i klasę).
2. Klasa przejściówki wymaga implementacji kilku metod. Implementacja ich powinna być bardzo prosta (po jednej linijce ;) ). Możesz zignorować metody *isBroken* i *isWorking*.
3. Wyjaśnij Krysi jak się używa przejściówki implementując statyczną metodę *plugEuropeanDeviceIntoUKSocket* w klasie *edu.kis.pio.triptouk.SocketProblemHelper*. Argumenty tej metody to:
  - *device* – urządzenie wyposażone we wtyczkę typu europejskiego, które chcemy podłączyć do gniazdka,
  - *socket* – gniazdko UK, do którego podłączyć chcemy urządzenie *device*.WSKAZÓWKA: Musisz utworzyć instancję przejściówki.
4. Zweryfikuj implementację uruchamiając testy.