Laboratorium
Projektowanie Obiektowe – Wzorce Projektowe

Temat : *Wzorzec behawioralny Command*
Wzorzec strukturalny Composite

Historia zmian

Data	*Wersja*	*Autor*	*Opis zmian*
19.04.2010	1.0	Tomasz Kowalski	Utworzenie dokumentu i wprowadzenie zadania laboratoryjnego.
04.04.2011	1.1	Tomasz Kowalski	Modyfikacja treści. Zmiana zestawu wzorców.
17.05.2011	1.2	Tomasz Kowalski	Drobne poprawki.
2.03.2012	1.3	Tomasz Kowalski	Uwzględnienie zmian w hierarchii sterowników
7.05.2013	2.0	Tomasz Kowalski	Aktualizacja w związku z reorganizacją laboratorium
9.4.2015	3.0	Dominik Żurek	Aktualizacja aplikacji i zmiana nazw na język angielski
6.10.2016	3.1	Tomasz Kowalski	Zmiana repozytorium z svn-a na github
29.5.2018	3.2	Tomasz Kowalski	Wydzielenie pytań
16.4.2019	4.0	Tomasz Kowalski	Zmiany dotyczące dziedziny problemowej.

1. Cel laboratorium

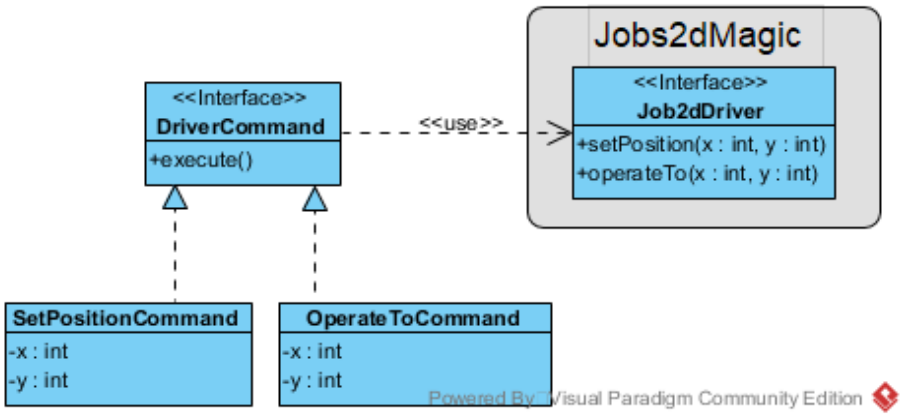
Głównym celem laboratoriów jest zapoznanie się z wzorcami projektowymi: *Command*, *Composite*. Należą one do różnych grup wzorców projektowych (behawioralne i strukturalne). Zajęcia powinny pomoc studentom rozpoznawać omawiane wzorce w projektach informatycznych, samodzielnie implementować wzorce oraz dokonywać odpowiednich modyfikacji wzorca w zależności od potrzeb projektu.

2. Laboratorium:

1. Laboratorium jest kontynuacją poprzednich laboratoriów dotyczących wzorca **adapter.** Rozwiązanie poniższych zadań należy utworzyć w nowych pakietach, których nazwy będą odpowiadać stosowanym wzorcom projektowym (np. *edu.kis.powp.command*).
2. Wyniki prac nad oprogramowaniem ***Jobs2dMagic*** uzyskane dzięki zastosowaniu wzorca projektowego adapter są do tej pory jak najbardziej pozytywne. Skłoniły to szefostwo do próby rozszerzenia oprogramowania ***Jobs2dMagic***. Łatwo jest dokonać specjalizacji adaptera do linii (*LineDrawerAdapter*) zaimplementowanego w poprzednim zadaniu. Niestety, nie jest to możliwe dla rzeczywistych sterowników urządzeń, do których implementacji nie mamy dostępu. Konieczne (i jak się okaże przydatne) jest rozdzielenie hierarchii sterowników od hierarchii klas dostarczających nową funkcjonalność (tzw. hierarchii abstrakcji). Szef rozważa wykorzystać w tym celu wzorzec **command**.
3. **UWAGA: pod koniec zajęć wyniki prac na laboratorium muszą być każdorazowo oznaczane w repozytorium jako osobny *release/tag*. Braki w tym zakresie są równoważne z brakiem obecności na zajęciach.**

   Diagramy i odpowiedzi na pytania *nie* powinny być dodawane do repozytorium tylko mają być przesłane w wiadomości mailowej do prowadzącego zajęcia.

3.1. Projektowanie hierarchii poleceń



Ilustracja 1: Diagram UML z przykładowym zastosowaniem wzorca command

1. Według diagramu na ilustracji 1 zaprojektuj i zaimplementuj hierarchię klas reprezentujących podstawowe funkcje urządzenia.
2. Wzorzec projektowy to półprodukt. Dopasuj (wg własnego pomysłu) hierarchię poleceń tak, żeby *odbiorcą* poszczególnych poleceń był zawsze obiekt należący do realizowanej w ramach poprzednich zadań hierarchii sterowników. Najlepiej, żeby implementacje *DriverCommand* nie były „na sztywno” związane z określonym sterownikiem, ale żeby mogły być wykorzystane w różnych kontekstach. Przetestuj opracowane rozwiązanie.

3. W hierarchii poleceń zaprojektuj klasę *ComplexCommand,* która umożliwiała by reprezentowanie **dowolnego** ciągu poleceń.
4. ***Pytanie*:** Jakie wzorce występują w hierarchii poleceń, poprzez fakt posiadania klasy *ComplexCommand*?
5. Zaimplementuj kilka „fabryk", które będą zwracały złożone polecenie rysujące figurę (np. prostokąt, okrąg). Dodaj do aplikacji *TestJobs2dPatterns* odpowiednie testy.
6. *Zaimplementuj klasę umożliwiającą generację złożonego ciągu poleceń, z których składają się przykładowe figury rysowane przez skrypty z klasy *FiguresJoe.* Za każdy świadomie wykorzystany do rozwiązania wzorzec projektowy otrzymasz * :).
7. ****Pytanie*:** Jakie jeszcze wzorce (jawnie nie nazwane do tej pory) można odnaleźć w projekcie?