

Technologia programowania - 2020

Zadanie 1 Tworzysz aplikację obliczającą pole i obwód figur geometrycznych. W aplikacji jest zdefiniowany interfejs z metodami *getArea()* i *getPerimeter()*, który jest implementowany przez klasy *Rectangle* oraz *Circle*. Chcesz wykorzystać zewnętrzną klasę *Square*, która posiada inny interfejs niż Twój. Wykorzystaj wzorzec **Adapter** do zaadaptowania klasy *Square*.

Zadanie 2 Tworzysz aplikację, która pobiera informację z wielu portali społecznościowych i prezentuje je w jednym miejscu zgodnie z Twoimi preferencjami. Oczywiście każdy portal wymaga innej interpretacji metody *getMessage()* odpowiedzialnej za pobieranie wpisów. Ponadto chcesz mieć wpływ na treść wiadomości: np. obcinania wiadomości dłuższych niż 100 słów, cenzurowanie określonych słów. Wykorzystując wzorzec **Decorator** zaproponuj rozwiązanie, które umożliwi dynamiczne i elastyczne rozszerzenie funkcjonalności metody *getMessage()*. Podaj przykład sytuacji w której kolejność dekorowania ma wpływ na końcową postać wiadomości.

Zadanie 3 Piszysz program do automatycznego generowania raportów na temat klientów instytucji finansowej. Raporty mogą być potrzebne do różnych celów i w zależności od celu powinny zawierać jedynie wybrane informacje o klientach (np. nazwisko, adres, wiek, telefon, płeć, stan konta). Ponadto Twój program ma współpracować z wieloma systemami i potrzebnych będzie wiele różnych reprezentacji raport, dla których trudno ustalić wspólny interfejs (np. HTML, XML). W oparciu o wzorzec **Builder** zaproponuj rozwiązanie.

Zadanie 4 Z użyciem wzorca **State** stwórz automat posiadający trzy stany. Każdy ze stanów powinien implementować przejście do stanu poprzedniego oraz następnego. Ma przykład stanem poprzednim w stosunku do stanu drugiego (2) jest stan pierwszy (1) a następnym stan trzeci (3), itd.. Automatem powinien sterować obiekt *Context* posiadający metody, które będą wywoływały metody przejść zaimplementowane dla poszczególnych stanów. Zaimplementuj w pętli sytuację n przejść przez wszystkie stany automatu, czyli: 123 123 123 i tak n razy. W pętli powinna być tylko instrukcja: *context.goNext()*.

Zadanie 5 Tworzysz narzędzie do automatycznego generowania raportów. Wszystkie raporty składają się z trzech części: header, footer oraz body. Trudno jest z góry przewidzieć jaką reprezentacja raportu będzie potrzebna (np. plain text, XML, HTML), ale możesz założyć, że klasa reprezentująca daną część raportu jest zgodna z góry ustalonym interfejsem (np. każda klasa nagłówka musi mieć metody ustalające tytuł i datę raportu jako odpowiednio sformatowany String). Zaproponuj rozwiązanie umożliwiające oddzielenie w kodzie programu reguł wyboru i tworzenia elementów raportu od pozostałej części logiki. Jaki wzorzec wykorzystasz?