Programowanie Obiektowe
Laboratorium 2

Obiektowa implementacja problemu.
Wzorce projektowe

Sprawozdanie **w formacie PDF** proszę wysłać przez platformę Teams lub na adres mailowy podany przez prowadzącego zajęcia (wyłącznie w przypadku problemów z oprogramowaniem Teams). Sprawozdanie powinno zawierać **imię i nazwisko autora/autorki**, **numer albumu** i **datę**. Również **nazwy przesyłanych plików** (sprawozdanie, archiwa ZIP z kodem itp.) muszą zawierać **nazwisko i imię autora/autorki**.

Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z:

- praktyczną realizacją dziedziczenia i implementacji interfejsu w języku Java,
- implementacją wzorca projektowego Dekorator.

Zadanie 1. (3,5 pkt)

Naszym celem będzie stworzenie hierarchii klas i interfejsów reprezentujących napisy ujęte w dowolnie rozbudowaną kombinację nawiasów: zwykłych, klamrowych itp. Przykład takiego napisu można zobaczyć pod treścią tego zadania. Taką kombinację nawiasów będziemy nazywać *otoczeniem* tekstu. Napis musi udostępniać jednakowe API niezależnie od tego, jakie (i czy w ogóle) posiada on otoczenie. W szczególności powinien dysponować metodą `String pobierzZawartość()`, zwracającą tekst ujęty w odpowiednią kombinację nawiasów. Tego typu zadanie programistyczne najlepiej rozwiązać, używając wzorca strukturalnego **Dekorator**.

Sugeruje się utworzenie następujących klas i interfejsów (podane nazwy są tylko przykładowe):

- `Napis`. Interfejs wymuszający na implementujących go klasach posiadanie metody `pobierzZawartość()` zwracającej `String`.
- `ZwykłyTekst`. Tekst pozbawiony nawiasów.

- `Otoczenie`. Podstawowym zadaniem tej klasy jest przechowywanie referencji do *zawartości*, którą może być zarówno inne `Otoczenie`, jak i `ZwykłyTekst`. Sugeruje się, by `Otoczenie` było klasą *abstrakcyjną*, po której będą dziedziczyły konkretne otoczenia, różniące się rodzajem nawiasów.
- `Nawiasy`, `Klamry` itp. Klasy konkretne dziedziczące po klasie `Otoczenie`. Ich metoda `String pobierzZawartość()` musi pobierać *zawartość* i otaczać ją nawiasami odpowiedniego rodzaju.

Przy implementacji klas należy zadbać o hermetyzację.

Po odpowiedniej implementacji opisanych klas i interfejsów poniższy ciąg poleceń:

```
ZwykłyTekst tekst = new ZwykłyTekst("Tekst do ozdobienia");
Napis udekorowany = new Nawiasy(new Klamry(new Nawiasy(tekst)));
System.out.println(udekorowany.pobierzZawartość());
```

powinien spowodować wypisanie w konsoli:

```
({(Tekst do ozdobienia)})
```

W funkcji `main` należy przetestować tworzenie i wypisywanie tekstów zarówno zwykłych, jak i udekorowanych różnymi kombinacjami nawiasów.

Zadanie 2. (1 pkt)

W powyższym przykładzie obiekty klasy `Otoczenie` (i potomnych) były tworzone na podstawie obiektu klasy implementującej interfejs `Napis`. Teraz należy uzupełnić klasę `Otoczenie` (i potomne) o konstruktor pozwalający tworzyć takie obiekty na podstawie obiektu `String`, np.

```
Napis udekorowany2 = new Nawiasy("Nawiasem mówiąc...");
```

W funkcji `main` należy przetestować stworzony konstruktor.