

KURS JĘZYKA JAVA

DRZEWA WYRAŻEŃ

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1.

W pakiecie `obliczenia` zdefiniuj interfejs `Obliczalny`, reprezentujący obiekty, na których można coś policzyć metodą `oblicz()`. Zadaniem tej metody ma być w klasach implementujących ten interfejs wykonanie obliczeń i zwrócenie wyniku jako wartości typu `double`.

W pakiecie `obliczenia` zdefiniuj też abstrakcyjną klasę `Wyrazenie` reprezentującą wyrażenie arytmetyczne. Klasa ta ma implementować interfejs `Obliczalny` (nie definiuj tej metody w klasie, gdyż jeszcze nie wiadomo co należy policzyć) — będzie to więc baza dla innych konkretnych klas realizujących określone zadania w wyrażeniu. W klasie `Wyrazenie` umieść dwie statyczne metody ze zmienną liczbą argumentów, które będą realizowały zadanie sumowania i mnożenia wyrażeń:

```
abstract class Wyrazenie {
    // ...
    /** metoda sumująca wyrażenia */
    public static double suma (Wyrazenie... wyr) {
        /* ... */ }
    /** metoda mnożąca wyrażenia */
    public static double iloczyn (Wyrazenie... wyr) {
        /* ... */ }
}
```

Następnie zdefiniuj klasy dziedziczące po klasie `Wyrazenie`, które będą reprezentowały kolejno liczbę, zmienną, stałą, operacje arytmetyczne (dodawanie, odejmowanie, mnożenie i dzielenie oraz jednoargumentowe operacje zmiany znaku na przeciwny i odwrotności) i popularne funkcje matematyczne (sinus, cosinus, potęgowanie, logarytmowanie, itp). Klasy te powinny być tak zaprojektowane, aby można z nich było zbudować drzewo wyrażenia: obiekty klas `Liczba` lub `Zmienna` to liście, a operatory to węzły wewnętrzne w takim drzewie. We wszystkich klasach tych podyfiniuj metody `toString()` oraz `equals(Object)`.

W klasie `Zmienna` zdefiniuj statyczne pole finalne do pamiętania zbioru wszystkich zmiennych w programie (pary identyfikator–liczba). Do przechowywania zmiennych możesz wykorzystać klasę `Zbior` z poprzedniego zadania. Odczytywanie wartości zmiennej ma polegać na zidentyfikowaniu pary w tym zbiorze i odczytaniu wartości związanej z identyfikatorem.

Klasa `Stala` ma reprezentować takie stałe wartości jak stała Archimedesesa π , stała Eulera e czy stała złotego podziału ϕ , które są często używane w wyrażeniach arytmetycznych.



Zadanie 2.

Uzupełnij swoje zadanie o krótki program testowy napisany poza pakietem `obliczenia`. Program ma rzetelnie sprawdzić działanie obiektów reprezentujących wyrażenie arytmetyczne.

W programie testowym skonstruuj drzewa obliczeń, wypisz je metodą `toString()` a potem oblicz i wypisz otrzymane wartości. Przetestuj swój program dla następujących wyrażień:

```

3+5
2+x*7
(3*11-1)/(7+5)
arctan(((x+13)*x)/2)
pow(2,5)+x*log(2,y)

```

Na przykład wyrażenie $7+x*5$ należy zdefiniować następująco:

```

Wyrażenie w = new Dodaj(
    new Liczba(7),
    new Mnoz(
        new Zmienna("x"),
        new Liczba(5)
    )
);

```

Ustaw na początku programu testowego zmienną `x` na wartość `2.0`.

Zadanie 3.

Dopisz w plikach źródłowych w pakiecie `obliczenia` komentarze dokumentacyjne. Stwórz także plik `package-info.java` z komentarzem do całego pakietu.

Uwaga 1.

W swoich programach nie czytaj ani nie analizuj danych ze standardowego wejścia. Drzewa wyrażień zdefiniuj na stałe w programie testowym.

Uwaga 2.

Program należy skompilować i uruchomić w zintegrowanym środowisku programistycznym *NetBeans* albo *IntelliJ!* Wygeneruj też na podstawie komentarzy dokumentacyjnych dokumentację do zawartości całego pakietu **obliczenia**.