

KURS PROGRAMOWANIA W JAVIE

DRZEWA OBLICZEŃ

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1.

Zdefiniuj publiczny interfejs `Obliczalny`, który będzie reprezentować obiekty, które umieją coś policzyć metodą `oblicz()`. Metoda ta ma zwracać wartość typu `Double`.

Zadanie 2.

Zdefiniuj publiczną i abstrakcyjną klasę bazową `Wyrazenie` reprezentującą drzewo obliczeń dla wyrażenia arytmetycznego. Klasa ta powinna implementować interfejs `Obliczalny` bez definiowania metody `oblicz()`.

Następnie zdefiniuj klasy dziedziczące po klasie `Wyrazenie`, które będą reprezentowały: liczbę, zmienną (zmienne mają być pamiętane w kolekcji `HashMap<String, Double>`), operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie i modulo) i porównania (równe, różne, mniejsze, większe, mniejsze-równe i większe-równe). W klasach potomnych zdefiniuj metody `oblicz()` oraz `toString()`. Obliczenie porównania ma zwracać wartość 1 albo 0, w zależności od tego czy warunek jest prawdziwy czy fałszywy. Przy obliczaniu dzielenia i modulo powinny być zgłaszane wyjątki, gdy dzielnik ma wartość 0. Konstruktory wymienionych klas powinny sprawdzać, czy ich argumenty są różne od `null`, a jeśli nie są to mają zgłosić odpowiedni wyjątek.

Napisz krótki program testowy, sprawdzający działanie obiektów tych klas. Niech twój program obliczy i wypisze wartość następujących wyrażeń (ustaw zmienną x na wartość -1):

```
3+5
7*11+2.72/3.14
(3*11-1)/(7+5)
((x+1)*x)/2!=0
(2*x+1)%(x+1)<1==0
```

Zadanie 3.

Zdefiniuj publiczną i abstrakcyjną klasę bazową `Instrukcja` reprezentującą drzewo obliczeń w programie. Klasa ta powinna implementować interfejs `Obliczalny` bez definiowania metody `oblicz()`.

Następnie zdefiniuj klasy dziedziczące po klasie `Instrukcja`, które będą reprezentowały: instrukcję blokową, deklarację zmiennej (zmienne zapamiętują w kolekcji `HashMap<String, Double>` i inicjalizują wartością 0), instrukcję przypisania wartości wyrażenia do zmiennej, instrukcje warunkowe (takie jak instrukcje `if` oraz `if-else` w języku C), instrukcje pętli (takie jak instrukcje `while` i `do-while` w języku C), instrukcję czytania ze standardowego wejścia oraz instrukcję pisania na standardowe wyjście. W klasach potomnych zdefiniuj metody `oblicz()` oraz `toString()`. Warunek w instrukcji warunkowej lub w pętli jest wyrażeniem (warunek jest prawdziwy tylko wtedy gdy wartość wyrażenia jest różna od 0). Instrukcja blokowa powinna być inicjalizowana dowolną liczbą instrukcji wewnętrznych (konstruktor ze zmienną liczbą argumentów); pozatym instrukcja ta ma spamiętywać jakie zmienne zostały utworzone w tym bloku i na końcu ma je usunąć (nie wolno tworzyć zmiennych o takich samych nazwach). Konstruktory wymienionych klas powinny sprawdzać, czy ich argumenty są różne od `null`, a jeśli nie są to mają zgłosić odpowiedni wyjątek.

Napisz krótki program testowy, sprawdzający działanie obiektów tych klas. Niech twój program sprawdzi, czy wczytana liczba jest pierwsza. Program ten może działać według następującego schematu:

```
var n;
read n;
if (n<2) write 0;
else
{
```

```

var p;
var wyn;
p = 2;
while (p*p<=n)
{
    if (n%p==0)
    {
        wyn = p;
        p = n;
    }
    p = p+1;
}
if (wyn>0) write 0;
else write 1;
}

```

Uwaga.

Wszystkie klasy i interfejsy umieść w pakiecie obliczenia.