laboratorium: zadanie 6 *termin: 11–14 kwietnia 2011 r.*

KURS JĘZYKA C++

DRZEWA OBLICZEŃ

Instytut Informatyki Uniwersytetu Wrocławskiego *Paweł Rzechonek*

Zadanie 1.

Zdefiniuj klasę `para`, która będzie przechowywać pary klucz–wartość, gdzie klucz jest typu `string` a wartość typu `int`. Klucz powinien być polem publicznym i niemodyfikowalnym, a wartość polem ukrytym, które można odczytać za pomocą gettera i zmodyfikować tylko za pomocą settera.

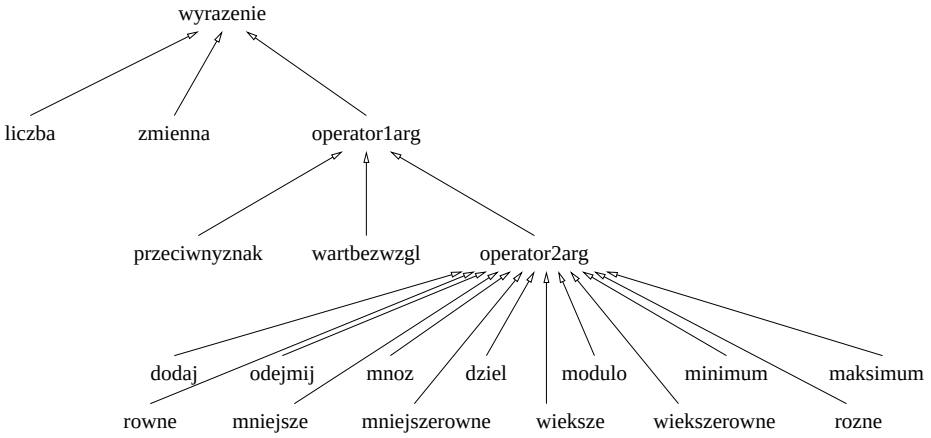
Zdefiniuj potem klasę `slownik`, która będzie implementowć w postaci listy dwukierunkowej zbiór obiektów typu `para`. Klasa `slownik` powinna oferować podstawowe operacje słownikowe: wyszukiwanie (metoda `szukaj()`), wstawianie (metoda `wstaw()`), usuwanie (metoda `usun()`) i policzenie wszystkich elementów w zbiorze (metoda `ile()`). Oprócz funkcji składowej `szukaj()` zdefiniuj operator indeksowania, który będzie alternatywą dla tej metody.

Pamiętaj, aby klasa `slownik` była tylko opakowaniem dla homogenicznej dynamicznej struktury danych opartej na węzłach. W klasie `wezel` zaimplementuj metody do wstawiania pary klucz–wartość, usuwania pary i wyszukiwania w liście wartości według zadanego klucza.

Zadanie 2.

Zdefiniuj abstrakcyjną klasę bazową `wyrazenie`, reprezentującą wyrażenie arytmetyczne. W klasie tej umieść deklaracje abstrakcyjnych metod `oblicz()` (jej zadaniem w klasach potomnych będzie obliczanie wartości wyrażenia i przekazywanie wyniku typu `int`) oraz `opis()` (ta metoda ma zwracać napis reprezentujący wyrażenie).

Następnie zdefiniuj klasy dziedziczące po klasie `wyrazenie`, które będą reprezentowały kolejno liczbę (stała całkowitoliczbowa), zmienną (zmienne pamiętaj w statycznym zbiorze asocjacyjnym, czyli na zdefiniowanej wcześniej liście par klucz–wartość), operacje arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie i modulo oraz jednoargumentowa operacja zmiany znaku na przeciwny) i porównania (wynikiem porównania ma być jak w języku C liczba 0 albo 1 odpowiadająca wartościom logicznym *false* albo *true*). 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. W klasach potomnych zdefiniuj metody `oblicz()` oraz `opis()`.

Zadanie 3.

Na koniec napisz krótki program testowy, sprawdzający działanie obiektów tych klas. W swoim programie skonstruuj drzewa obliczeń, wypisz je metodą `opis()` a potem oblicz i wypisz ich wartości metodą `oblicz()`. Przeprowadź testy dla następujących wyrażeń:

```
3+5
2+x*7
(3*11-1)/(7+5)
((x+1)*x)/2
2*x+1<0
```

Na przykład wyrażenie `2+x*7` należy zdefiniować następująco:

```
wyrazenie *w = new dodaj(
    new liczba(2),
    new mnoz(
        new zmienna("x"),
        new liczba(7)
    )
);
```

Ustaw na początku programu testowego zmienną `x` na wartość –3.