23 lutego 2017

Programowanie obiektowe

Lista 1.

Zaprogramuj dwa z poniższych zadań w języku w C, Pascalu czy w Pythonie, jednak **bez** użycia klas bibliotecznych C++ czy Pythona. W przypadku Pythona, gdzie jedynym sposobem definiowania typu jest deklaracja klasy, można poniższe zadania zaimplemntować wykorzystując krotki bądź listy, które będą przekazywane do funkcji jako argumenty.

Za każde zadanie na tej liście można otrzymać do 2 punktów.

Zadanie 1. Zadeklaruj typ `Kolekcja`, który może implementować albo *zbiór* przechowując elementy bez powtórzeń, albo *torbę*, przechowując elementy z powtórzeniami. Przyjmujemy, że przechowywane elementy są zadeklarowane deklaracją `typedef ... Elem` (`Elem` może być dowolnym typem). Zaimplementuj

- procedurę `void wstaw(Kolekcja **k, Elem e)`;
- funkcję `int rozmiar(Kolekcja **k)` zwracającą długość listy;
- funkcję `int szukaj(Kolekcja *k, Elem e)` zwracającą liczbę znalezionych elementów.

Przyjmujemy, że argumenty `e` są zmiennymi automatycznymi. Zmienna typu `Kolekcja` powinna być inicjowana tylko za pomocą dwóch procedur: `zbiór(Kolekcja **k)` albo `torba(Kolekcja **k)`. Sposób inicjacji determinuje działanie funkcji `wstaw` i `szukaj`.

Zadanie 2. Zadeklaruj typ `typedef Figura ...`, który może reprezentować figury geometryczne: punkt, koło lub kwadrat[1], wraz z ich położeniem w dwuwymiarowym układzie współrzędnych. Przyjmij, że pole `typfig` typu wyliczeniowego wyznacza rodzaj reprezentowanej figury geometrycznej. Zdefiniuj trzy procedury:

- `narysuj(Figura *f)` (wystarczy, że wypisze odpowiedni komunikat w trybie tekstowym);
- `przesuń(Figura *f, float x, float y)` (przesuwa figurę o zadany wektor);
- `int zawiera(Figura *f, float x, float y)` sprawdzającą, czy figura zawiera punkt o zadanych współrzędnych.

Podczas oceny programu wskaż miejsca, które wymagają modyfikacji, jeśli rozszerzymy typ `Figura` o możliwość reprezentacji trójkątów. Zadeklaruj również odpowiednie procedury inicjujące zmienne typu `Figura`.

Zadanie 3. Zaprojektuj własny typ `Zespolone`, tj. podaj odpowiednie struktury danych służące do przechowywania wartości tego typu, oraz nagłówki funkcji odpowiadających czterem standardowym operacjom arytmetycznym na liczbach zespolonych. Zaimplementuj dwie możliwe realizacje takich funkcji: w pierwszej funkcja zwraca wskaźnik do nowoutworzonego elementu tego typu; w drugiej funkcja modyfikuje drugi z argumentów.

Sposób reprezentacji liczb zespolonych jest dowolny; mogą być pamiętane w *postaci algebraicznej*, w *postaci trygonometrycznej* czy też w *postaci biegunowej*.

Zadanie 4. Zdefiniuj własny typ `DrzewoBinarne` reprezentujące drzewo binarnych poszukiwań. `DrzewoBinarne` przechowujące w węzłach struktury ustalonego typu `str` wraz z procedurami wstawiania i wyszukiwania elementów w drzewie. Zaprogramuj również funkcję `int rozmiar` zwracającą liczbę elementów drzewa.

[1] można przyjąć, że boki kwadratu są równoległe do osi układu współrzędnych

Powyższe zadania powinne być zaimplementowane w postaci modułu bądź biblioteki. Częścią rozwiązania powinien być krótki przykład (w odrębnym pliku) korzystający z zaimplementowanego modułu. Oceniane i punktowane są jedynie dwa zadania.
Przypomnienie: Rozwiązanie proszę wrzucić do serwisu SKOS do najbliższej środy do godz. 8:00

Marcin Młotkowski