Programowanie obiektowe

wersja 2009/2010

Laboratorium **2.**

Zagadnienia

1. Wskaźniki i tablice – powtórzenie

   Definiowanie wskaźników, mechanika ruchu wskaźnika, arytmetyka wskaźników, dynamiczna alokacja pamięci, stałe wskaźniki, wskaźniki do stałych, stałe wskaźniki do stałych, tablice wskaźników, wskaźniki na tablice, wskaźniki do funkcji, tablica wskaźników do funkcji, tablice wielowymiarowe, wskaźniki na wskaźniki.
2. Formatowanie obiektowego wejścia i wyjścia

   Strumienie wejścia/wyjścia, operatory `<<` *i* `>>`*, klasa* `ios`*, sterowanie formatem, manipulatory bezargumentowe i parametryzowane, operacje wejścia i wyjścia na plikach.*
3. Wprowadzenie do programowania obiektowego

   Hermetyzacja, składniki klasy, prototyp funkcji.
4. Wprowadzenie do UML – reprezentacja klasy
5. Definiowanie widoczności (+, -, #), opis atrybutów, opis operacji.

Ćwiczenie 1. (Wskaźniki i tablice)

1. Omów wskaźnik typu **`void`** oraz operację rzutowania (dotyczy wskaźników);
2. Niech **`char *wskCh = "Tekst"`**; Wyświetl adres, na który wskazuje `wskCh`;
3. Wyjaśnij różnicę między przesyłaniem argumentu do funkcji przez referencję a wskaźnik;
4. Zaproponuj funkcję, która na podstawie typu tablicy zwróci jej rozmiar (ilość elementów);
5. Niech istnieje **`double tab[5]`**. Co oznaczają: `tab+1; *tab+1; &tab+1; *(tab+1); tab[0]+1; &tab[0]+1; tab++;`
6. Niech **`double *w`** wskazuje na **`tab`**. Co oznaczają: `w+1;w+=1; *w+1; &w+1; *(w+1); w[0]+1; &w[0]+1; w++; *(&w[2]-1);`
7. Wyjaśnij dodawanie, odejmowanie oraz porównywanie wskaźników;
8. Czy można wskaźnik ustawić sam na siebie? Odpowiedź uzasadnij;
9. Zapisz tablicę jedenastu wskaźników do funkcji (przyjmujących w argumencie stały wskaźnik do stałej typu **`char`**), które zwracają tablicę wskaźników na tablicę elementów typu **`double`**;

10. Zapisz prototyp funkcji wywoływanej beż żadnych argumentów, która zwraca wskaźnik na funkcję wywoływaną bez żadnych argumentów, a zwracającą typ **`double`**;
11. Zdefiniuj i przetestuj działanie wskaźnika na wskaźnik na wskaźnik na typ **`short`**;
12. Wyjaśnij pojęcie wycieku pamięci oraz obiecaj, że w swoich programach unikniesz tego błędu;
13. Porównaj sposoby dynamicznej alokacji tablicy w C i C++.

Zadanie 1. (Formatowanie obiektowego wejścia i wyjścia)

1. Napisz program, który wczyta kilka liczb rzeczywistych do tablicy TAB (tworzonej dynamicznie – rozmiar podaje użytkownik), następnie wyprowadzi te liczby w wyrównanej kolumnie. W programie kontroluj, aby zadany rozmiar tablicy nie przekroczył wartości MAX (zdefiniowanej przez użytkownika);
2. Zredaguj precyzję wyprowadzanych liczb tak, aby była ona zależna od życzenia użytkownika;
3. Sformatuj wyprowadzane liczby tak, aby miały one jednakową zadaną liczbę cyfr po kropce oraz aby kropki dziesiętne były pisane w jednej kolumnie;
4. Wyprowadzaj każdą liczbę dwukrotnie: w notacji zwykłej i naukowej Np. w postaci `TAB[2] = 3.20  3.20E+00`;

Dodatkowe*

5. Zaprojektuj i zaimplementuj eksperyment, w którym porównany zostanie czas wyprowadzania tekstu na konsolę przy pomocy strumieni `cout` z `cerr`
6. Napisz program, który z pliku wczyta serię liczb rzeczywistych znajdujących się w wierszach. Następnie zapisze te liczby w notacji naukowej do pliku wyjściowego. Ponadto w pliku wyjściowym na początku powinno pojawić się krótkie podsumowanie (informacja, skąd dane były czytane oraz ile wierszy – liczb – zostało wczytanych). Plik wyjściowy należy tworzyć automatycznie.

Zadanie 2. (Modelowanie i implementacja klasy `Data`)

Zaprojektuj i zaimplementuj klasę `Data`, która będzie reprezentowała datę. Opisz znaczenie atrybutów i działanie wszystkich zaproponowanych metod/funkcji. Zgodnie z opisem stwórz odpowiedni kod. Podstawowe funkcje należy zaimplementować, dla pozostałych wystarczą prototypy. Klasę przedstaw na diagramie UML.

Zadanie 3. (Modelowanie klasy `Zesp`)

Zaprojektuj klasę `Zesp`, która będzie reprezentować liczby zespolone. Klasę przedstaw zgodnie z diagramem UML. Opisz znaczenie atrybutów (przynajmniej 2) i działanie wszystkich zaproponowanych metod/funkcji (przynajmniej 3). (Klasy nie trzeba implementować).

Zadanie 4. (Quiz)

Zredaguj 5 pytań testowych (np. jednokrotnego wyboru, wielokrotnego wyboru, dopasowanie odpowiedzi itp.), które pozwolą sprawdzić wiedzę nabytą podczas wykonywania zadań 1, 2 i 3.

Zaliczenie ćwiczenia

1. Kody programów (Zadania 1 i 2) muszą być czytelnie sformatowane i opatrzone odpowiednimi komentarzami.
2. Raport (Zadania 2, 3 i 4) w formie drukowanej nie może przekroczyć jednej spójnej kartki A4.

Życzę owocnej pracy