Programowanie obiektowe 8.04.2017
Kolokwium 1

Zadanie 1. (13 punktów)
Napisz funkcję, która przyjmie jako argumenty:

- referencję na dynamicznie zaalokowaną tablicę wskaźników do zmiennych typu `char`,
- rozmiar tej tablicy.

W tablicy, wartości części pól wskazują na niezerowe pozycje w pamięci, pod którymi zakładamy, że znajdują się poprawnie zaalokowane wartości typu `char`. Pozostałe pola wskazują na `NULL`. Funkcja powinna zmodyfikować tablicę tak, aby pozbyć się wartości wskazujących na `NULL`, pozostałe przesunąć tak, aby nie było przerw, a nadmiarowe miejsce zwolnić. Funkcja powinna zwrócić rozmiar tablicy po modyfikacji.

Przykład:
Wejściowa tablica zawiera wskazania na: `A NULL NULL X W NULL`.
Tablica po modyfikacji zawiera wskazania na: `A X W`, funkcja zwraca wartość 3.

Zadanie 2. (14 punktów)
Napisz klasę `WordReplacer` posiadającą:

- publiczny konstruktor przyjmujący w argumencie obiekt klasy `string`,
- publiczną metodę `replace` przyjmującą dwie wartości typu `unsigned int`, nie zwracającą wartości,
- publiczną, bezargumentową metodę `text` zwracającą obiekt klasy `string`.

Konstruktor w argumencie przyjmuje ścieżkę do pliku tekstowego. W pliku znajdują się słowa oddzielone pojedynczymi spacjami. Konstruktor powinien pobrać jego zawartość do obiektu swojej klasy. Metoda `replace` ma za zadanie zamienić ze sobą miejscami słowa o podanych numerach (indeksowanych od 1). Jeżeli słowa lub słów o podanym indeksie nie ma w napisie, należy zignorować wywołanie metody. Metoda `text` ma zwrócić zmodyfikowany napis.

Obiekt klasy `WordReplacer` powinien pracować na kopii danych z pliku. W wyniku jego działań, zawartość pliku nie może zostać zmieniona. Prywatne pola i metody można tworzyć dowolnie w zależności od potrzeb i wybranego sposobu rozwiązania. Jeżeli zaproponowane rozwiązanie będzie tego wymagało, należy dopisać destruktor.

Przykład:
W pliku znajduje się: Lorem ipsum dolor sit amet.
Po wywołaniu metody replace z argumentami 2 i 4, metoda `text` powinna zwrócić: Lorem sit dolor ipsum amet.

Zadanie 3. (13 punktów)
Napisz klasę abstrakcyjną `RegularPolyhedron` modelującą bryłę posiadającą wielokąt foremny w podstawie. Ma ona posiadać:

- prywatne, zmiennoprzecinkowe zmienne określające długość boku podstawy, pole postawy, wysokość bryły oraz prywatną,całkowitą zmienną określającą ilość boków postawy,
- publiczny konstruktor, przyjmujący w argumentach trzy wartości – długość boku, ilość boków i wysokość, dodatkowo obliczający wartość pola postawy,
- chronione gettery do pola podstawy i wysokości,
- publiczną, czysto wirtualną, bezargumentową metodę `volume` zwracającą wartość zmiennoprzecinkową.

Napisz dziedziczące po niej publicznie dwie klasy: `RegularPrism` i `RegularPyramid`. Pierwsza z nich ma implementować metodę `volume`, tak aby zwracała objętość graniastosłupa foremnego, druga – ostrosłupa foremnego. Umieść wszystkie trzy klasy w przestrzeni nazw o nazwie `poly`.

Napisz funkcję `main`, w której utworzone zostaną obiekty obu pochodnych klas i wywołana ich metoda volume, tak, aby zademonstrować ich polimorfizm. Do elementów znajdujących się w przestrzeni nazw `poly` można odwoływać się wyłącznie za pomocą operatora zasięgu (::). Program należy napisać zachowując podział na pliki nagłówkowe i źródłowe.

Przydatne wzory:

$a=\frac{s^2\cdot n}{4\cdot \mathrm{tg}\left(\frac{\pi}{n}\right)}$ - pole podstawy, $v=a\cdot h$ - objętość graniastosłupa, $v=\frac{1}{3}a\cdot h$ - objętość ostrosłupa,

gdzie: *s* – długość krawędzi postawy, *n* – ilość krawędzi postawy, *h* – wysokość bryły.

Funcja tangens (`double tan(double)`) przyjmuje kąty w radianach i znajduje się w pliku nagłówkowym `cmath`.