

Programowanie - kolokwium III

13 czerwca 2016

Uwaga! Rozwiązania umieść w katalogu imie_nazwisko_nrindeksu. Przed oddaniem zadań usuń wszystkie pliki ze skompilowanym programem. Podczas kolokwium można korzystać z własnych programów i notatek, dokumentacji w Internecie, strony ćwiczeń oraz strony wykładu. Nie wolno się komunikować! Podejrzenie o ściąganie skutkuje zakończeniem kolokwium z wynikiem 0 pkt.

Zad. 1. (11 pkt)

Mścisław jest studentem II roku i właśnie przypomniał sobie, że jeszcze nie zaliczył wszystkich ćwiczeń Pracowni Technik Pomiarowych... Postanowił szybko nadrobić zaległości i na dodatkowym terminie pracowni przystąpił do ćwiczeń. Pierwszym ćwiczeniem, jakie musi wykonać, jest budowa filtra dolnoprzepustowego R-C. O ile pojemność kondensatora podano w instrukcji, o tyle opór rezystora zapisano za pomocą kodu kolorowego na oporniku... Student nie posiada umiejętności rozszyfrowania kodu kolorowego. Niemniej jednak w jego głowie narodziła się myśl, genialny pomysł – zmierzy opór opornika multimetrem uniwersalnym! Mścisław wykonał szereg pomiarów, a wyniki zapisał w pliku: http://www.fuw.edu.pl/~mkuich/programowanie_2016/ko1_3_2016/dane.dat Ze zdziwieniem stwierdził, że wyniki tych pomiarów są różne. Pomóż zdesperowanemu Mścisławowi – napisz program, który wyznaczy wartość średnią i odchylenie standardowe jego pomiarów.

Program powinien zawierać:

- dowolny strumień wejścia, którym wczyta plik
- sprawdzenie, czy poprawnie otwarto plik
- zamknięcie pliku
- dowolną tablicę, w której umieści dane
- funkcję, która oblicza wartość średnią
- funkcję, która oblicza odchylenie standardowe.

W funkcji `main` otwórz plik i wczytaj dane. Następnie wywołaj funkcje, oblicz średnią i odchylenie standardowe oraz wypisz wynik na ekranie.

Wskazówki: Unikaj kopiowania obiektów, czyli tablicę z danymi przekaż do funkcji za pomocą referencji. Wartość średnia opisana jest wzorem:

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

Odchylenie standardowe jest opisane wzorem:

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)^2}$$

Zad. 2. (14 pkt)

Napisz klasę `Slowo`, która zawiera jedno pole chronione `wiadomosc` typu `string` oraz metodę publiczną: `Wpisz_wiadomosc()`, która wpisuje wiadomość ze standardowego wejścia do pola `wiadomosc`. Następnie napisz klasę `Szyfr`, która będzie dziedziczyła po klasie `Slowo` w sposób publiczny. Klasa pochodna `Szyfr` powinna zawierać jedno pole prywatne `szyfr` i dwie metody publiczne: `Zaszyfruj()` i `Wypisz_szyfr()`. Metoda `Zaszyfruj()` wpisuje do pola `szyfr` zawartość pola `wiadomosc` w sposób zaszyfrowany (w tym zadaniu – każdy znak przesuwamy o 5), a metoda `Wypisz_szyfr()` wypisuje na ekran zawartość pola `szyfr`. W funkcji `main()`, stwórz obiekt klasy `Szyfr`, wpisz wiadomość, zaszyfruj ją i wypisz szyfr na ekran. Pamiętaj, aby przed zaszyfrowaniem sprawdzić, czy wiadomosc została wpisana przez użytkownika – jeśli nie wywołaj metodę `Wpisz_wiadomosc()`.

Przykładowe działanie programu:

Podaj wiadomosc: Odnaleziono bozon Higgsa

Twoja wiadomosc: Tisfqjntst%gtts%Mnllxf

Wskazówki: Przydatne mogą okazać się metody z klasy `string`: `size()` i `resize()`. Jeśli chcesz uniknąć pomijania spacji możesz skorzystać z funkcji `getline(cin, wiadomosc)`, gdzie strumieniem wejściowym jest standardowe wejście `cin`, a `wiadomosc` jest obiektem typu `string`.

Zad. 3. (2 pkt - dla chętnych)

Uwaga! Aby otrzymać punkty za to zadanie, problem musi być rozwiązany idealnie (tzn. oceniam binarnie 2 pkt lub 0 pkt).

Napisz program, który jako argument funkcji `main` przyjmie wartość oporu opornika w Ohmach. Następnie program wywoła konstruktor klasy `ResistorColour` i utworzy obiekt tej klasy. Na tym obiekcie należy wywołać funkcję `PaintCode()`, która narysuje w konsoli kod kolorowy. Funkcje rysujące kod kolorowy zostały oprogramowana w pliku:

http://www.fuw.edu.pl/~mkuich/programowanie_2016/ko1_3_2016/ResistorColour.cpp.

Zapisz ten kod, sporządź plik nagłówkowy dla tej biblioteki, załącz ją w swoim programie i wykorzystaj do wyświetlenia kodu kolorowego opornika o zadanym oporze. *Wskazówki:* Typ wyliczeniowy `enum` nie musi być umieszczony w pliku nagłówkowym. Pamiętaj o sprawdzeniu, czy użytkownik podał opór opornika przy uruchamianiu programu.