

# Zadania z Informatyki

## Programowanie w C++

### 1. Wprow. do C++ – iostream, typy string i bool

1. Napisz program wypisujący tabliczkę mnożenia o  $m$  kolumnach i  $n$  wierszach (zwróć uwagę na wyrównanie wypisywanych wartości w kolumnach).
2. Napisz program rozwiązujący równania kwadratowe, współczynniki powinny być wczytywane z klawiatury, a program po wypisaniu wyników powinien pytać czy liczyć jeszcze raz (odp. tekstowe: „tak” lub „Nie” – program powinien ignorować wielkość liter).
3. Napisz program, który będzie generował losowe  $n$ -literowe hasło zawierające co najmniej jedną literę wielką, cyfrę i znak specjalny. W wygenerowanym hasle nie powinny sąsiadować dwie spółgłoski lub samogłoski.

Dla bardziej ambitnych:

4. Napisz program, który dla podanego przez użytkownika słowa znajdzie i wypisze wszystkie anagramy.

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

### 2. Strumienie, strumienie plikowe i stringowe

1. Napisz program rozwiązujący kilka równań kwadratowych, dane powinny być wczytywane z pliku, a wyprowadzane na ekran i do pliku.
2. Napisz program, który we wskazanym pliku zamieni pierwsze litery wyrazów na wielkie, a wynik umieści w nowym pliku.
3. Napisz program, który w wielowierszowym tekście wczytanym z klawiatury policzy wiersze, słowa, znaki „czarne” i spacje (wykorzystaj strumień stringstream).

Dla bardziej ambitnych:

4. Napisz program – książkę telefoniczną, który pozwoli na zarządzanie kontaktami. Dane powinny być zapisane z użyciem struktury i zapisywane w pliku binarnym. Program (z użyciem prostego menu) powinien pozwolić na: wyszukiwanie, dodawanie, modyfikację i kasowanie pozycji z ks. telefonicznej.

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

### 3. Funkcje (przeładowanie), tablice, wskaźniki i referencje

1. Napisz dwie funkcje o tej samej nazwie (`pierw`), które obliczą:
  - (a) pierwiastek kwadratowy liczby  $x$ ,
  - (b) pierwiastek  $n$ -tego stopnia z liczby  $x$ .(Zastosuj te funkcje w prostym programie, np. wypisującym pierwiastki kwadratowe i 3-go stopnia liczb od 1 do 5).
2. Napisz program (z użyciem funkcji) który wczyta (np. z pliku) elementy macierzy kwadratowej (o rozmiarze 3) oraz policzy:
  - (a) sumę elementów na diagonalu,
  - (b) iloczyn tej macierzy przez liczbę skalarną,
  - (c) iloczyn tej macierz przez samą siebie.Nie zapomnij o wypisaniu macierzy wynikowej.

Dla bardziej ambitnych: wczytaj z pliku elementy macierzy bez założenia jej rozmiaru (kolejne wiersze macierzy powinny być w kolejnych wierszach pliku).

3. Posługując się wskaźnikami do funkcji napisz program, który z tablicy wczytanych z pliku liczb obliczy:
- (a) sumę elementów,
  - (b) średnią arytmetyczną,
  - (c) średnią geometryczną.
- Każdy z tych podpunktów powinna realizować osobna funkcja, wywoływana za pomocą wskaźnika do funkcji.

**Dla bardziej ambitnych: rozszerz zadanie o szukanie wartości środkowej (mediany).**

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

#### 4. Klasy, obiekty

1. Utwórz klasę (o nazwie *punkt*) opisującą punkt w układzie kartezjańskim. Klasa ta powinna mieć funkcje składowe, które umożliwiają:
- (a) przypisanie współrzędnych punktu,
  - (b) wypisanie współrzędnych punktu,
  - (c) obliczenie odległości punktu od innego punktu (tej samej klasy),
  - (d) przesunięcie punktu o wektor  $[x,y]$
- Dla bardziej ambitnych: rozszerz zadanie o obliczenie współrzędnych punktu obroconego względem początku układu wsp. o kąt  $\alpha$**
2. Utwórz klasę opisującą okrąg w układzie kartezjańskim. Klasa powinna umieć zainicjować wartości opisujące okrąg (zarówno postaci punktu środka i dł. promienia, jak i wsp. środka i dł. promienia), wypisać informacje o sobie (środek, długość promienia), policzyć swoje pole, określić liczbę punktów wspólnych z obiektem takiej samej klasy. Skorzystaj z klasy *punkt* utworzonej w poprzednim zadaniu.

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

#### 5. Dziedziczenie, polimorfizm

1. (a) Utwórz klasę opisującą ogólne właściwości płaskiej figury geometrycznej (zdefiniuj wirtualne funkcje składowe liczące: pole, obwód oraz wypisujące informacje o figurze).
- (b) Utwórz klasy pochodne z niej (np. prostokąt, kwadrat) i zdefiniuj wirtualne funkcje składowe (pole, obwód).
- (c) Utwórz na podstawie (a) klasę opisującą figurę przestrzenną (objętość).
- Dla bardziej ambitnych: (d) Zdefiniuj klasę potomną opisującą figury przestrzenne (stożki, graniastosłupy) dziedzicząc po klasie z (c) oraz (b)**
2. Utwórz klasę do podawania danych z walidacją wartości (o nazwie np. *dana*).
- Klasa powinna zawierać funkcje: wypisującą przechowywaną wartość, wypisującą komunikat zachęty i wczytującą wartość z klawiatury, sprawdzającą, czy wartość jest poprawna (np. o nazwie *isvalid()*), a także podającą przechowywaną wartość *get()*.
- Klasa powinna zawierać konstruktor kopiujący, konstruktor z komunikatem zachęty.
- Na podstawie tej klasy (*dana*) utwórz klasy potomne do przechowywania wartości długości boków (nie mogą być ujemne), a także kątów (muszą być z zakresu 0-360)

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

## 6. Operatory, struktury danych, kontenery

1. Utwórz klasę opisującą wektor w układzie kartezjańskim, zaimplementuj operatory liczące: długość wektora (moduł), iloczyn wektora i liczby skalarnej, iloczyn skalarny z innym wektorem.
2. Wykorzystując szablon *vector* z STL utwórz własną klasę przechowującą stringi. Wczytaj do niej zawartość wybranego pliku tekstowego wiersz po wierszu, wypisz jego elementy, wysortuj je i znajdź wiersz najkrótszy.

Dla bardziej ambitnych:

3. Wczytaj plik tekstowy zawierający imiona. Policz powtarzające się słowa i wypisz ich statystykę. W programie posłuż się szablonem STL o nazwie *map* (tablicą asocjacyjną).

Aby zaliczyć ćwiczenie, wyślij list z załącznikami, zawierającymi postać źródłową (a nie skompilowaną!) napisanych dzisiaj programów do prowadzącego zajęcia (kopia do wykładowcy).

## 7. Zaliczenie