

Obliczanie Pola Figur – Program CPP

Konrad Kukuła 14.

Spis Treści

Spis Treści.....	2
1. Cel projektu.....	3
1.1 Zakres działania – obsługiwane figury geometryczne.....	3
2. Wymagania funkcjonalne.....	3
3. Wymagania нефunkcjonalne.....	3
4. Opis technologii.....	4
5. Struktura kodu.....	4
5.1 – Lista funkcji w programie Funkcjonalnych.....	4
5.2 – Lista funkcji w programie Figur.....	5
6. Interfejs użytkownika.....	5
7. Testowanie i walidacja.....	5
8. Podsumowanie.....	5

1. Cel projektu

Celem projektu było stworzenie aplikacji konsolowej w języku **C++**, umożliwiającej użytkownikowi **obliczanie pól i obwodów podstawowych figur geometrycznych** na podstawie danych wejściowych. Program uwzględnia jednostki długości i powierzchni (cm, cm²) oraz waliduje dane wejściowe.

1.1 Zakres działania – obsługiwane figury geometryczne

Program obsługuje sześć podstawowych figur płaskich. Każda figura ma zaimplementowane wzory umożliwiające obliczenie **pola powierzchni** oraz **obwodu** na podstawie danych podanych przez użytkownika.

Figura	Dane wejściowe użytkownika	Zakres działań
Kwadrat	Bok a	Pole, obwód
Prostokąt	Długość a , szerokość b	Pole, obwód
Trapez	Podstawa a , podstawa b , wysokość h , dwa pozostałe boki	Pole, obwód
Trójkąt	Podstawa a , wysokość h , bok b , bok c	Pole, obwód
Równoległobok	Podstawa a , bok sąsiedni b , wysokość h	Pole, obwód
Sześciokąt foremny	Bok a	Pole, obwód

2. Wymagania funkcjonalne

- Obliczanie pola i obwodu: kwadrat, prostokąt, trapez, trójkąt, równoległobok, sześciokąt
- Obsługa jednostek (cm, cm²)
- Interaktywny interfejs użytkownika z menu
- Możliwość powtarzania obliczeń bez restartu programu
- Walidacja danych wejściowych (liczby, brak liter)

3. Wymagania niefunkcjonalne

- Czytelny i przejrzysty kod
- Możliwość łatwej rozbudowy o kolejne figury
- Wydajność obliczeniowa
- Intuicyjny interfejs użytkownika (menu, komunikaty)

4. Opis technologii

Aplikacja została napisana w języku C++ przy użyciu standardowej biblioteki wejścia/wyjścia.

Środowisko uruchomieniowe: dowolny kompilator C++ .

5. Struktura kodu

- Funkcja **main()**: uruchamia menu główne
- Funkcje figur: oddzielne funkcje dla każdej figury (np. kwadrat, trapez)
- Funkcje pomocnicze: **wczytajFloat**, **wczytajInt**, **WyczyscEkran**
- Walidacja danych wejściowych poprzez pętle sprawdzające typ danych

5.1 – Lista funkcji w programie Funkcjonalnych

Nazwa funkcji	Typ	Opis działania
main()	Główna	Uruchamia program, ustawia język i wywołuje menu główne
menu()	Logika interfejsu	Główna pętla programu z menu wyboru figur i obsługą kontynuacji/wyjścia
wyswietlMenu()	Pomocnicza	Wyświetla dostępne figury geometryczne
WyczyscEkran()	Pomocnicza	Czyści ekran konsoli w zależności od systemu operacyjnego
linia()	Estetyczna	Wyświetla separator w konsoli
wczytajFloat()	Walidacja	Wczytuje i sprawdza dane typu float, zabezpieczając przed literami
wczytajInt()	Walidacja	Wczytuje i sprawdza dane typu int, zabezpieczając przed literami

5.2 – Lista funkcji w programie Figur

Nazwa funkcji	Typ	Opis działania
kwadrat()	Obliczeniowa	Oblicza pole i obwód kwadratu
prostokat()	Obliczeniowa	Oblicza pole i obwód prostokąta
trapez()	Obliczeniowa	Oblicza pole i obwód trapezu
trojkat()	Obliczeniowa	Oblicza pole i obwód trójkąta
rownoleglobok()	Obliczeniowa	Oblicza pole i obwód równoległoboku
szesciokat()	Obliczeniowa	Oblicza pole i obwód sześciokąta foremnego

6. Interfejs użytkownika

Użytkownik wybiera figurę z listy (menu), podaje wymagane wymiary, a program oblicza pole i obwód. Na końcu użytkownik może zdecydować, czy chce kontynuować, czy zakończyć program. Wszystkie wyniki są prezentowane wraz z jednostkami.

7. Testowanie i walidacja

- Testy obejmowały różne wartości długości, w tym liczby zmiennoprzecinkowe i błędne dane (litery)
- Program zabezpiecza przed błędami typu (np. litery zamiast liczb)
- Interfejs został sprawdzony pod kątem przejrzystości i intuicyjności

8. Podsumowanie

Program pozwala na wybór figury, wprowadzenie danych, obliczenie pola i obwodu oraz ponowne wykonanie operacji bez restartu programu. Kod jest modularny i łatwy do rozszerzenia. Możliwym ulepszeniem jest dodanie wyboru jednostek.