PROJEKT: generator figur i wzorów geometrycznych w konsoli.

- **Cel projektu:** Program powstał jako ćwiczenie z zakresu pętli zagnieżdżonych oraz operacji na układzie współrzędnych w trybie tekstowym. Jego zadaniem jest rysowanie symetrycznych figur na podstawie parametrów podanych przez użytkownika.
- **Kluczowe funkcjonalności:** Dynamiczne skalowanie rozmiaru figury, walidacja błędnych danych wejściowych oraz przejrzyste menu wyboru kształtu.
- **Technologie:** C++17, biblioteka standardowa (`<iostream>`, `<iomanip>`).
- **Czego mnie to nauczyło:** Praca nad tym kodem pozwoliła mi lepiej zrozumieć zależności matematyczne w konsoli oraz pisanie zwięzłych funkcji pomocniczych.

Gotowy kod źródłowy (skopiuj i wklej do wxDev-C++)

Cpp

```
#include <iostream>
#include <limits>

using namespace std;

// Deklaracje funkcji pomocniczych
void wyswietlMenu();
int pobierzLiczbe(const char* komunikat, int min, int max);
void rysujOdwroconyPustyTrojkat(int wysokosc, char znak);
void rysujPelnaPiramide(int wysokosc, char znak);
void rysujPustyKwadrat(int bok, char znak);

int main() {
    int wybor = 0;
    int rozmiar = 0;
    char znak = '*';

    do {
        wyswietlMenu();
        wybor = pobierzLiczbe("Wybierz opcje (0-3): ", 0, 3);

        if (wybor == 0) {
            cout << "\nZamykanie generatora. Do zobaczenia w plenerze i
za biurkiem!\n";
            break;
        }

        rozmiar = pobierzLiczbe("Podaj wysokosc/rozmiar figury (3 -
25): ", 3, 25);
        cout << "Podaj pojedynczy znak do rysowania (np. *, #, +): ";
        cin >> znak;

        // Czyszczenie bufora na wypadek wpisania wiecej niz jednego
znaku
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
        cout << "\n--- WYNIK GENEROWANIA ---\n\n";

        switch (wybor) {
            case 1:
                rysujOdwroconyPustyTrojkat(rozmiar, znak);
                break;
            case 2:
                rysujPelnaPiramide(rozmiar, znak);
                break;
            case 3:
                rysujPustyKwadrat(rozmiar, znak);
                break;
        }

        cout << "\n-------------------------\n\n";

    } while (wybor != 0);

    cout << "Nacisnij Enter, aby zakonczyc program...";
    cin.get();
    return 0;
}

// Wyswietla menu glowne programu
void wyswietlMenu() {
    cout << "============================================\n";
    cout << "  GENERATOR FIGUR GEOMETRYCZNYCH (C++ / Konsola)\n";
    cout << "============================================\n";
    cout << "1. Odwrocony pusty trojkat\n";
    cout << "2. Pelna piramida\n";
    cout << "3. Pusty kwadrat (ramka)\n";
    cout << "0. Wyjscie z programu\n";
    cout << "--------------------------------------------\n";
}

// Bezpieczne pobieranie liczby z walidacja bledow wejscia
int pobierzLiczbe(const char* komunikat, int min, int max) {
    int wartosc;
    while (true) {
        cout << komunikat;
        cin >> wartosc;

        if (cin.fail() || wartosc < min || wartosc > max) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Blad! Wpisz poprawna liczbe z zakresu od " << min
<< " do " << max << ".\n";
        } else {
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            return wartosc;
        }
    }
}

// Rysuje odwrocony pusty trojkat
void rysujOdwroconyPustyTrojkat(int wysokosc, char znak) {
    for (int i = wysokosc; i >= 1; --i) {
        // Wciecie z lewej strony
```

```
        for (int spacja = 0; spacja < wysokosc - i; ++spacja) {
            cout << " ";
        }

        // Rysowanie krawedzi i srodka
        int szerokosc = 2 * i - 1;
        for (int j = 1; j <= szerokosc; ++j) {
            if (i == wysokosc || j == 1 || j == szerokosc) {
                cout << znak;
            } else {
                cout << " ";
            }
        }
        cout << "\n";
    }
}

// Rysuje pelna piramide
void rysujPelnaPiramide(int wysokosc, char znak) {
    for (int i = 1; i <= wysokosc; ++i) {

for (int spacja = 1; spacja <= wysokosc - i; ++spacja) {
            cout << " ";
        }
        for (int j = 1; j <= 2 * i - 1; ++j) {
            cout << znak;
        }
        cout << "\n";
    }
}

// Rysuje pusty kwadrat
void rysujPustyKwadrat(int bok, char znak) {
    for (int i = 1; i <= bok; ++i) {
        for (int j = 1; j <= bok; ++j) {
            if (i == 1 || i == bok || j == 1 || j == bok) {
                cout << znak << " ";
            } else {
                cout << "  ";
            }
        }
        cout << "\n";
    }
}
```