

Polimorfizm, tablice obiektów

1. Przykład

Dana jest abstrakcyjna klasa bazowa **TFigure**.

```
class TFigure
{
public:
    virtual char *getName() const = 0;

    virtual double area() const = 0;
    virtual double perimeter() const = 0;
};
```

Dziedziczy z niej klasa **TSquare** mogąca obliczać pole powierzchni i obwód kwadratu. Klasa **TSquare** posiada jeszcze dodatkowe metody i konstruktory.

```
class TSquare : public TFigure
{
public:
    TSquare();
    TSquare(double side);

    char *getName() const;

    void setSide(double side);
    double getSide() const;

    double area() const;
    double perimeter() const;

private:
    double side;
};

TSquare::TSquare(): side(1) {}
TSquare::TSquare(double side): side(side) {}

char *TSquare::getName() const {return "Kwadrat";}

void TSquare::setSide(double side) {TSquare::side = side;}
double TSquare::getSide() const {return side;}

double TSquare::area() const {return side * side;}
double TSquare::perimeter() const {return 4 * side;}
```

Wprowadź i uruchom program wykorzystujący funkcję **calcAndInfo()** pozwalającą na obliczanie sumarycznego pola powierzchni i obwodu dwóch kwadratów. Wykorzystaj tablicę wskaźników na obiekty typu **TFigure**.

```
void calcAndInfo(TFigure *fig[], int figNo)
```

```
{
    double totalArea = 0;
    double totalPerimeter = 0;

    for(int i = 0; i < figNo; i++)
    {
        totalArea += fig[i]->area();
        totalPerimeter += fig[i]->perimeter();
    }

    cout << "Sumaryczna powierzchnia: " << totalArea << endl;
    cout << "Sumaryczny obwód: " << totalPerimeter << endl;
}

int main()
{
    const int figNo = 2;
    TFigure *fig[figNo];

    fig[0] = new TSquare(50);
    fig[1] = new TSquare(70);

    calcAndInfo(fig, figNo);

    for(int i = 0; i < figNo; delete fig[i++]);

    return EXIT_SUCCESS;
}
```

2. Zadanie

Opracuj dodatkowe klasy **TRectangle** i **TTriangle** do obsługi prostokąta i trójkąta prostokątnego. Zmodyfikuj program tak, aby używał wszystkich typów opisujących figury geometryczne.