

## Lekcja 18 – podprogramy (funkcje)

**Temat:** wyświetlanie tekstu na ekranie

**Kod programu źródłowego:**

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

void rzut_kosci ()
{
    int k;
    k = rand () % 6;
    cout << k << ' ';
}

int main()
{
    int x, i;
    srand (time(NULL) );

    cout <<"Symulator gry w kosci - z podprogramami" << endl;
    cout << "Ile chcesz kości? ";
    cin >> x;

    for (i = 0; i <x; i++)
    {
        rzut_kosci ();
    }

    return 0;
}
```

**Podprogram:** Podprogram to wydzielona część programu wykonująca jakieś operacje. Podprogramy stosuje się, aby uprościć program główny i zwiększyć czytelność kodu.

W języku C/C++ podprogramy nazywamy funkcjami.

**Budowa funkcji w C++:**

```
typ_zwracanej_wartosci nazwa_funkcji( typ_argumentu_1
nazwa_argumentu_1 /*,...*/ , typ_argumentu_n nazwa_argumentu_n )
{
    return zwracana_wartosc;
}
```

Każda funkcja posiada trzy własności:

1. zwraca dane (lub nie jeśli tego nie chcemy);

2. posiada swoją nazwę;
3. może posiadać dowolną liczbę argumentów wejściowych (lub może nie mieć żadnego, jeśli tego nie chcemy).

**Miejsce funkcji w programie:** Funkcja musi być umieszczona przed funkcją główną.

```
void rzut_kosci ()
{
    int k;
    k = rand () % 6;
    cout << k << ' ';
}

int main()
{
    ...
    return 0;
}
```

**Wywołanie funkcji :** Funkcję wywołujemy używając jej nazwy.

```
void rzut_kosci ()
{
    int k;
    k = rand () % 6;
    cout << k << ' ';
}

int main()
{
    ...
    for (i = 0; i <x; i++)
    {
        rzut_kosci ();
    }

    return 0;
}
```

Daną funkcję możemy wywoływać z poziomu funkcji głównej **main** lub z poziomu innej funkcji.

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

void rzut_kosci ()
{
    int k;
    k = rand () % 6;
    cout << k << ' ';
}

void ile_kosci()
{
    int x, i;
    cout << "Ile chcesz kości? ";
    cin >> x;

    for (i = 0; i <x; i++)
    {
        rzut_kosci ();
    }
}

int main()
{
    srand (time(NULL) );
    cout <<"Symulator gry w kosci - z podprogramami" << endl;
    ile_kosci();

    return 0;
}
```

### Ćwiczenia:

1. Napisz program liczący:
  - a. Pole kwadratu
  - b. Pole prostokąta
  - c. Pole trójkątaKażda z tych figur ma być wyliczona w oddzielnej funkcji.