

# Język C++ zajęcia nr 10

## I – Przykład zastosowania programowania obiektowego – operacje na ułamkach.

Napisz program **CPP10-1**

W pierwszym wierszu w komentarzu wpisz imię i nazwisko.

Zdefiniuj klasę **Ulamek** posiadającą następujące elementy publiczne:

1. Dane składowe: licznik, mianownik (typu int)
2. Funkcje składowe:

```
Ulamek(int liczn, int mianow)
```

*KONSTRUKTOR DWUARGUMENTOWY - tworzy obiekt i dokonuje inicjalizacji pól obiektu (licznika i mianownika) wartościami argumentów liczn i mianow*

```
Ulamek()
```

*KONSTRUKTOR DOMNIEMANY - przypisuje ułamkowi wartości licznika i mianownika podane z klawiatury*

```
double wartosc()
```

*zwraca wartość ułamka w postaci liczby rzeczywistej dziesiętnej*

```
void drukuj()
```

*drukuję ułamek w nowej linii w postaci: licznik / mianownik = wartość (np. 3/5 = 0.6) (wykorzystaj wynik zwracany przez funkcję wartosc)*

```
void dodaj(Ulamek u)
```

```
void dodaj(int liczba)
```

*funkcje (przeciążone) dodają do bieżącego ułamka wartość określoną za pomocą parametru; wynik umieszczają w bieżącym ułamku.*

Pamiętaj, aby sprawdzać czy mianownik jest różny od zera.

Napisz funkcję `main` tworzącą dwa ułamki i realizującą scenariusz przykładowych działań na ułamkach z wykorzystaniem wszystkich powyższych funkcji.

### Przykładowy scenariusz zapisany w funkcji main:

```
int main()  
{  
    Ulamek a(3,5);  
    Ulamek b;  
    a.drukuj();  
    b.drukuj();  
    a.dodaj(b);  
    a.drukuj();  
    a.dodaj(3);  
    a.drukuj();  
}
```

Program **CPP10-1** należy przynieść na kolejne zajęcia w wersji elektronicznej (pendrive) – program ten będzie dalej rozwijany!

## II – Gra w monety (program **CPP10-2**)

Gracz ma w portfelu  $n$  monet złotówkowych. Gra polega na wykonywaniu rzutów kolejnymi monetami wyjmowanymi z portfela. Moneta spada na kwadratową planszę o boku 15 x 15 cm w ten sposób, że środek monety zajmuje losowo wybrany punkt tego kwadratu (moneta może częściowo wystawać poza kwadrat), po czym moneta pozostaje na planszy. Jeżeli moneta zakryje (częściowo) inną monetę leżącą na planszy, gracz zabiera pulę z planszy i gra się kończy sukcesem. Jeżeli graczowi skończą się monety w portfelu i żadna rzucona moneta nie zakryje innej leżącej na planszy – gra się kończy porażką. Program drukuje liczbę rzutów wykonanych w trakcie rozgrywki i informację o końcowym rezultacie (sukces, porażka). Przyjmujemy, że złotówka ma średnicę 22 mm.

W celu napisania programu:

1. Zdefiniuj klasę *moneta*, zawierającą:
  - dane składowe  $x, y$  – określające współrzędne monety na planszy,
  - funkcję składową *void rzut()*, wywoływaną w momencie rzutu daną monetą, wyznaczającą losowe położenie środka monety na planszy,
  - funkcję składową *int sprawdz(int numerdrugiejmonety)*, która sprawdza, czy dana (rzucona) moneta zakrywa drugą monetę (wcześniej rzuconą) o podanym numerze, jeśli zakrywa – funkcja zwraca 1, jeśli nie – zwraca 0.
2. Napisz definicje funkcji składowych *rzut* i *sprawdz*.
3. Zadeklaruj globalny wskaźnik do obiektu klasy *moneta* – będzie to nazwa tablicy obiektów klasy *moneta*: *moneta \*tab*;

4. W funkcji *main* między innymi:

- utwórz *n*-elementową tablicę monet (tablicę obiektów),
- zaprojektuj pętlę realizującą rozgrywkę.

Możesz wykorzystać poniższą propozycję fragmentu funkcji *main*:

```
int main()
{
    int n, sukces=0;
    srand(time(0));
    cout<<"Podaj liczbe monet: ";
    cin>>n;

    // tu utwórz n-elementową tablicę obiektów klasy moneta

    for (int i=0; i<n; i++) {

        // tu umieść ciało pętli realizującej rozgrywkę

    };
    if (sukces) cout<<"SUKCES! wykonano rzutów "<<i+1;
        else cout<<"Sorry, porazka!";
} //koniec main
```