

Lekcja 13. (p)

Temat: **Skracamy i wyłączamy, czyli dodawania ułamków ciąg dalszy.**

Cele lekcji:

Wykazanie na przykładzie podawania wyniku dodawania ułamków, jak ważne jest wyświetlanie danych w optymalnej postaci.
Opracowanie rozwiązania dotyczącego skracania ułamków i wyłączania całości.

Uczeń:

- używa NWD w algorytmie i programie do skracania ułamków
- świadomie stosuje zmienne pomocnicze
- organizuje sposób wyświetlania wyników
- objaśnia i stosuje algorytm wyłączania części całkowitej z ułamka
- układa program upraszczający ułamek i wyłączający z niego całości
- dyskutuje strukturę ułożonego programu i decyduje użycia zmiennych pomocniczych

Przebieg lekcji:

1. Kontynuacja programowania z poprzedniej lekcji.
2. Wyłączanie całości, czyli wynik w najprostszej postaci. (str. 95)
3. Testowanie programu końcowego.
4. Wysłanie działającego programu do oceny – tylko plik źródłowy. (.cpp)

Zadanie domowe:

Powtórz zakres ćwiczenia z lekcji, przetestuj program.

Ćwiczenie z podręcznika – str. 96

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

```
int NWD(int a, int b)           //Funkcja obliczająca NWD
{
    int i;
    while(b!=0)
    {
        i = b;
        b = a%b;
        a = i;
    }
    return (a);
}
```

```
int NWW(int a, int b)         //Funkcja obliczająca NWW
{
    return a/NWD(a,b)*b;
}
```

```
int main()                   //Główna funkcja programu
{
    int c, i, l1, l2, m1, m2, licznik, mianownik;
    cout<<"Podaj licznik pierwszego ułamka ";
    cin >> l1;
    cout << "Podaj mianownik pierwszego ułamka ";
```

```

cin >> m1;

cout<<"Podaj licznik drugiego ułamka ";
cin >> l2;
cout << "Podaj mianownik drugiego ułamka ";
cin >> m2;
mianownik = NWW(m1, m2);
licznik = mianownik/m1*l1 + mianownik/m2*l2;
i=NWD(licznik,mianownik);
licznik = licznik / i;
mianownik = mianownik / i;
if (licznik/mianownik !=0)
{
c=licznik/mianownik;
licznik=licznik-c*mianownik;
mianownik = mianownik / NWD(i,mianownik);

cout<<l1<<"/"<<m1<<" + "<<l2<<"/"<<m2<<" = "<<c<<" i "<<licznik<<"/"<<mianownik;
}
else
cout<<l1<<"/"<<m1<<" + "<<l2<<"/"<<m2<<" = "<<licznik<<"/"<<mianownik;
return 0;
}

```

Rozwiązanie rozszerzające do lekcji 13-tej.

Sprawdzenie programu dodającego ułamki zwykłe z wykorzystaniem NWD i NWW.

```

#include<iostream>
#include<cstdlib>
using namespace std;
int A,a1,a2,B,b1,b2;
int c1,c2,d1,d2,m,mianownik;
int X,x1,x2,W,w1,w2;
int a,b;
int NWD(int a, int b)
{
while(a!=b)
if(a>b)
a-=b; //lub a = a - b;
else
b-=a; //lub b = b-a
return a; // lub b - obie zmienne przechowują wynik NWD(a,b)
}

int NWW(int a, int b)
{
int pom;
pom=(a*b)/NWD(a,b);
return pom;
}

int main()
{

```

```

cout <<"podaj część całkowitą liczby 1 (A)"; cin >> A;
cout <<"podaj licznik liczby 1 (a1)"; cin >> a1;
cout <<"podaj mianownik liczby 1 (b1)"; cin>> a2;

cout <<"podaj część całkowitą liczby 2 (B)"; cin >> B;
cout <<"podaj licznik liczby 2 (b1)"; cin >> b1;
cout <<"podaj mianownik liczby 2 (b2)"; cin >> b2;
cout <<endl<<"Wprowadziłeś ułamki"<<endl<<endl;

cout <<"  "<<a1<<"      "<<b1<<endl;
cout <<A<<" -----"<<" + "<<B<<" -----"<<endl;
cout <<"  "<<a2<<"      "<<b2<<endl;

cout <<"postac ogolna zapisu ułamków"<< endl;
c1=A*a2+a1;
c2=a2;
d1=B*b2+b1;
d2=b2;
cout <<c1<<"      "<<d1<<endl;
cout <<"-----"<<" + -----"<<endl;
cout <<c2<<"      "<<d2<<endl;
m=NWD(c2,d2);
cout << "wspólny dzielnik wynosi = "<<m<< endl;
mianownik=NWW(c2,d2);
cout << "wspólny mianownik = "<<mianownik<< endl;
cout << "postac ułamka po wyznaczeniu wspólnego mianownika"<<endl;
x1= mianownik/c2*c1+mianownik/d2*d1;
x2=mianownik;
cout <<x1<<endl;
cout <<"-----"<<endl;
cout <<x2<<endl;
if (x1>x2)
{ W=x1/x2; w1=x1-W*x2; w2=x2;
cout <<"  "<<w1<<endl;
cout <<W<<" -----"<<endl;
cout <<"  "<<w2<<endl;
m=NWD(w1,w2);
cout <<"wspólny dzielnik dla licznika i mianownika = "<< m<<endl;

cout << "ułamek doprowadzony do najprostrzej postaci"<<endl;
w1=w1/m ;w2=w2/m;
cout <<"  "<<w1<<endl;
cout <<W<<" -----"<<endl;
cout <<"  "<<w2<<endl;
}
return 0;
}

```