

# C++ - Kantor wymiany walut

© Copyright by 3bird Projects 2024, <http://edukacja.3bird.pl>

## Ogólne

Program wylicza koszty nabycia waluty w oparciu o jej aktualny kurs.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii, twarde odstępki, odmienne cudzysłowy, odmienne apostrofy, odmienne myślniki. To godziny dodatkowej pracy na wykrywanie błędów i ich poprawianie. Kod należy przepisać ze zrozumieniem.

## Kod - wersja podstawowa

```
#include <iostream>
#include <limits> // Potrzebne do numeric_limits

using namespace std;
// Kursy walut należy systematycznie uaktualniać:
float kursEuro = 4.31;
float kursDolara = 3.95;
int wyborUzytkownika;
float iloscWaluty = 0;

int main() {
    cout << "==== KANTOR WYMIANY WALUT ====" << endl;
    cout << "Opcje wymiany:" << endl;
    cout << "1. PLN --> EUR" << endl;
    cout << "2. PLN --> USD" << endl;
    cout << "Podaj numer operacji (1-2): ";
    cin >> wyborUzytkownika;

    switch(wyborUzytkownika) {
    case 1:
        cout << "Wpisz, ile euro zamierzasz kupić: ";
        cin >> iloscWaluty;
        cout << "\nKoszt zakupu euro wynosi: " << (iloscWaluty * kursEuro) << " zł." << endl;
        break;
    case 2:
        cout << "Wpisz, ile dolarów zamierzasz kupić: ";
        cin >> iloscWaluty;
        cout << "\nKoszt zakupu dolarów: " << (iloscWaluty * kursDolara) << " zł." << endl;
        break;
    default:
        cout << "\nPodałś nieprawidłowy numer operacji.\n" << endl;
        break;
    }
    cout << "\nNaciśnij ENTER, aby zakończyć..." << endl;
```

```

cin.ignore(256, '\n');
cin.get();
return 0;
}

```

## Kod - wersja rozbudowana (*Windows*)

```

#include <iostream>
#include <limits> // Potrzebne do numeric_limits
#include <iomanip> // Potrzebne do wypisania kwoty do dwóch miejsc po przecinku
#include <algorithm> // Potrzebne do find()
#include <windows.h> // Potrzebne do kolorowania napisów; działa tylko w Windows.

using namespace std;

// Kursy walut należy systematycznie uaktualniać:
float kursEuro = 4.31;
float kursDolara = 3.95;
int wyborUzytkownika;
float iloscWaluty = 0;
string tablicaOdpowiedziTak[] = {"Tak", "tak", "T", "t", "Yes", "yes", "y", "Y", "taa", "yee", "ta", "Ta"};
string odpowiedz;

int main() {
    // Jeśli Twój Windows obsługuje kolorowe znaki ANSI w Wierszu poleceń, możesz poniższy
    // blok kodu pominąć:
    // ===== KOLOROWE ZNAKI ANSI =====
    // Na początek naprawiamy Windowsa, aby kolorował czcionki w konsoli.
    // Tworzymy tzw. uchwyt do tego, co będzie pojawiać się na konsoli (do bufora konsoli):
    HANDLE konsola = GetStdHandle(STD_OUTPUT_HANDLE);
    // Aktywujemy virtualny terminal i to, co będzie się na nim pojawiać:
    #ifndef ENABLE_VIRTUAL_TERMINAL_PROCESSING // Jeśli nie jest zdefiniowany, to:
    #define ENABLE_VIRTUAL_TERMINAL_PROCESSING 0x0004
    #endif

    // Aby powyższe działało, musimy aktywować i to, co poniżej.
    // Jeśli nie jest zdefiniowany, to:
    #ifndef ENABLE_PROCESSED_OUTPUT
    #define ENABLE_PROCESSED_OUTPUT 0x0001
    #endif

    // Wartość trybu (intup lub output). Słowo "dw" to skrót od "Display Window",
    // jest to jednak nazwa zmiennej, i może być inna:
    DWORD dwMode = 0;
    dwMode |= ENABLE_PROCESSED_OUTPUT | ENABLE_VIRTUAL_TERMINAL_PROCESSING;
    SetConsoleMode(konsola, dwMode);
    // ===== KONIEC BLOKU =====

```

**start:**

```
cout << "\n\033[1;34;40m=== KANTOR WYMIANY WALUT ===\033[0m\n" << endl;
```

```
while(true) {
```

```
    cout << "\033[1;30;40mOpcje wymiany:" << endl;
```

```
    cout << "1. PLN --> EUR" << endl;
```

```
    cout << "2. PLN --> USD" << endl;
```

```
    cout << "3. EUR --> PLN" << endl;
```

```
    cout << "4. EUR --> USD" << endl;
```

```
    cout << "5. USD --> PLN" << endl;
```

```
    cout << "6. USD --> EUR\033[0m" << endl;
```

```
    cout << "Podaj numer operacji (1-6): ";
```

```
    // Poniżej sprawdzamy nie tylko, czy udało się przypisać liczbę do zmiennej,
```

```
    // ale czy użytkownik nie wpisał czasem liczby zmiennoprzecinkowej typu 1.0:
```

```
    if ((cin >> wyborUzytkownika) && (cin.peek() == '\n')) {
```

```
        cout << "\033[1;30;40mNumer operacji wybranej przez użytkownika: " <<
```

```
        wyborUzytkownika << ".\033[0m\n" << endl;
```

```
        break;
```

```
    }
```

```
    else {
```

```
        cin.clear();
```

```
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
        cout << "\n\033[1;31;40mPodana wartość nie jest liczbą całkowitą. Spróbuj jeszcze
```

```
        raz.\033[0m\n" << endl;
```

```
    }
```

```
}
```

```
switch(wyborUzytkownika) {
```

```
case 1:
```

```
    while(true) {
```

```
        cout << "Wpisz, ile euro zamierzasz kupić: ";
```

```
        if (cin >> iloscWaluty) {
```

```
            break;
```

```
        }
```

```
        else {
```

```
            cin.clear();
```

```
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
```

```
            cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
```

```
            raz.\033[0m\n" << endl;
```

```
        }
```

```
    }
```

```
    cout << "\nKoszt zakupu takiej ilości euro wynosi: \033[1;32;40m" << fixed <<
```

```
    setprecision(2) << (iloscWaluty * kursEuro) << "zł\033[0m." << endl;
```

```
    break;
```

```
case 2:
```

```
    while(true) {
```

```
        cout << "Wpisz, ile dolarów zamierzasz kupić: ";
```

```
        if (cin >> iloscWaluty) {
```

```
            break;
```

```

    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości dolarów wynosi: \033[1;32;40m" << fixed <<
setprecision(2) << (iloscWaluty * kursDolara) << "zł\033[0m." << endl;
break;

```

case 3:

```

while(true) {
    cout << "Wpisz, ile złotych zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości złotych wynosi: \033[1;32;40m" << fixed <<
setprecision(2) << (iloscWaluty / kursEuro);
// Znak € funkcjonuje w kodowaniu UTF-8, dlatego na chwilę zmieniamy kodowanie:
SetConsoleOutputCP(CP_UTF8);
cout << u8"€\033[0m." << endl;
SetConsoleOutputCP(852); // Wracamy do kodowania CP852
break;

```

case 4:

```

while(true) {
    cout << "Wpisz, ile dolarów zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości dolarów wynosi: \033[1;32;40m" << fixed <<
setprecision(2) << ((iloscWaluty * kursDolara) / kursEuro);

```

// Znak € funkcjonuje w kodowaniu UTF-8, dlatego na chwilę zmieniamy kodowanie:

```
SetConsoleOutputCP(CP_UTF8);  
cout << u8"€\033[0m." << endl;  
SetConsoleOutputCP(852); // Wracamy do kodowania CP852  
break;
```

case 5:

```
while(true) {  
    cout << "Wpisz, ile złotych zamierzasz kupić: ";  
    if (cin >> iloscWaluty) {  
        break;  
    }  
    else {  
        cin.clear();  
        cin.ignore(numeric_limits<streamsize>::max(), '\n');  
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze raz.\033[0m\n" << endl;  
    }  
}  
cout << "\nKoszt zakupu takiej ilości złotych wynosi: \033[1;32;40m" << "$" << fixed  
<< setprecision(2) << (iloscWaluty / kursDolara) << "\033[0m." << endl;  
break;
```

case 6:

```
while(true) {  
    cout << "Wpisz, ile euro zamierzasz kupić: ";  
    if (cin >> iloscWaluty) {  
        break;  
    }  
    else {  
        cin.clear();  
        cin.ignore(numeric_limits<streamsize>::max(), '\n');  
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze raz.\033[0m\n" << endl;  
    }  
}  
cout << "\nKoszt zakupu takiej ilości euro wynosi: \033[1;32;40m" << "$" << fixed <<  
setprecision(2) << ((iloscWaluty * kursEuro) / kursDolara) << "\033[0m." << endl;  
break;
```

default:

```
cout << "\n\033[1;31;40mPodałś nieprawidłowy numer operacji.\033[0m\n" << endl;  
break;  
}  
  
cout << "\nCzy chcesz ponownie zakupić walutę? (tak / nie): ";  
cin >> odpowiedz;
```

```

bool jestObecne = find(begin(tablicaOdpowiedziTak), end(tablicaOdpowiedziTak),
odpowiedz)!= end(tablicaOdpowiedziTak);
if (jestObecne) {
    // Generalnie, raczej użylibyśmy tutaj pętli while()...
    // ale aby nie zagnieżdżać kolejnych pętli while() w sobie (bardzo brzydkie!),
    // oraz aby pokazać, że istnieje nadal archaiczne "goto" pochodzące jeszcze z batcha i
    // Pascala:
    goto start;
}
else {
    cout << "\n\033[1;30;40mNaciśnij ENTER, aby zakończyć...\033[0m" << endl;
}

cin.ignore(256,'\n');
cin.get();
return 0;
}

```

## Kod - wersja rozbudowana (*Linux*)

```

#include <iostream>
#include <limits> // Potrzebne do numeric_limits
#include <iomanip> // Potrzebne do wypisania kwoty do dwóch miejsc po przecinku
#include <algorithm> // Potrzebne do find()

using namespace std;
// Kursy walut należy systematycznie uaktualniać:
float kursEuro = 4.31;
float kursDolara = 3.95;
int wyborUzytkownika;
float iloscWaluty = 0;
string tablicaOdpowiedziTak[] = {"Tak", "tak", "T", "t", "Yes", "yes", "y", "Y", "taa", "yee", "ta",
"Ta"};
string odpowiedz;

int main() {
    start:
    cout << "\n\033[1;34;40m=== KANTOR WYMIANY WALUT ===\033[0m\n" << endl;
    while(true) {
        cout << "\033[1;30;40mOpcje wymiany:" << endl;
        cout << "1. PLN --> EUR" << endl;
        cout << "2. PLN --> USD" << endl;
        cout << "3. EUR --> PLN" << endl;
        cout << "4. EUR --> USD" << endl;
        cout << "5. USD --> PLN" << endl;
        cout << "6. USD --> EUR\033[0m" << endl;
        cout << "Podaj numer operacji (1-6): ";
        // Poniżej sprawdzamy nie tylko, czy udało się przypisać liczbę do zmiennej,

```

```

// ale czy użytkownik nie wpisał czasem liczby zmiennoprzecinkowej typu 1.0:
if ((cin >> wyborUzytkownika) && (cin.peek() == '\n')) {
    cout << "\033[1;30;40mNumer operacji wybranej przez użytkownika: " <<
    wyborUzytkownika << ".\033[0m\n" << endl;
    break;
}
else {
    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << "\n\033[1;31;40mPodana wartość nie jest liczbą całkowitą. Spróbuj jeszcze
    raz.\033[0m\n" << endl;
}
}
switch(wyborUzytkownika) {
case 1:
    while(true) {
        cout << "Wpisz, ile euro zamierzasz kupić: ";
        if (cin >> iloscWaluty) {
            break;
        }
        else {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
            raz.\033[0m\n" << endl;
        }
    }
    cout << "\nKoszt zakupu takiej ilości euro wynosi: \033[1;32;40m" << fixed <<
    setprecision(2) << (iloscWaluty * kursEuro) << "zł\033[0m." << endl;
    break;
case 2:
    while(true) {
        cout << "Wpisz, ile dolarów zamierzasz kupić: ";
        if (cin >> iloscWaluty) {
            break;
        }
        else {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
            raz.\033[0m\n" << endl;
        }
    }
    cout << "\nKoszt zakupu takiej ilości dolarów wynosi: \033[1;32;40m" << fixed <<
    setprecision(2) << (iloscWaluty * kursDolara) << "zł\033[0m." << endl;
    break;
}

```

case 3:

```
while(true) {
    cout << "Wpisz, ile złotych zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości złotych wynosi: \033[1;32;40m" << fixed <<
setprecision(2) << (iloscWaluty / kursEuro) << "€\033[0m." << endl;
break;
```

case 4:

```
while(true) {
    cout << "Wpisz, ile dolarów zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości dolarów wynosi: \033[1;32;40m" << fixed <<
setprecision(2) << ((iloscWaluty * kursDolara) / kursEuro) << "€\033[0m." << endl;
break;
```

case 5:

```
while(true) {
    cout << "Wpisz, ile złotych zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości złotych wynosi: \033[1;32;40m" << "$" << fixed
```



```
<< setprecision(2) << (iloscWaluty / kursDolara) << "\033[0m." << endl;
break;
```

case 6:

```
while(true) {
    cout << "Wpisz, ile euro zamierzasz kupić: ";
    if (cin >> iloscWaluty) {
        break;
    }
    else {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "\n\033[1;31;40mPodana wartość nie jest poprawną liczbą. Spróbuj jeszcze
        raz.\033[0m\n" << endl;
    }
}
cout << "\nKoszt zakupu takiej ilości euro wynosi: \033[1;32;40m" << "$" << fixed <<
setprecision(2) << ((iloscWaluty * kursEuro) / kursDolara) << "\033[0m." << endl;
break;
```

default:

```
cout << "\n\033[1;31;40mPodałś nieprawidłowy numer operacji.\033[0m\n" << endl;
break;
}
```

```
cout << "\nCzy chcesz ponownie zakupić walutę? (tak / nie): ";
cin >> odpowiedz;
```

```
bool jestObecne = find(begin(tablicaOdpowiedziTak), end(tablicaOdpowiedziTak),
odpowiedz) != end(tablicaOdpowiedziTak);
```

```
if (jestObecne) {
    // Generalnie, raczej użylibyśmy tutaj pętli while()...
    // ale aby nie zagnieżdżać kolejnych pętli while() w sobie (bardzo brzydkie!),
    // oraz aby pokazać, że istnieje nadal archaiczne "goto" pochodzące jeszcze z batcha i
    // Pascala:
    goto start;
}
else {
    cout << "\n\033[1;30;40mNaciśnij ENTER, aby zakończyć...\033[0m" << endl;
}
}
```

```
cin.ignore(256, '\n');
cin.get();
return 0;
```

```
}
```