

// PRZYKŁAD p_20.cpp

//prosta-tablicowa implementacja "stosu" liczb rzeczywistych double

//bez wyróżnienia osobnych funkcji WSTAW i POBIERZ

```
//##### PRZYKŁADOWY PROGRAM #####  
  
#include <iostream.h>  
#include <conio.h> // clrscr(), getch(), ungetch()  
#include <ctype.h> // tolower()  
  
void main()  
{  
    #define ROZMIAR_STOSU 10  
    double tablica_stosu[ROZMIAR_STOSU];  
    int ilosc_zapisanych=0;  
  
    #define ESC 27  
    char klawisz;  
    do  
    {  
        clrscr();  
        if(ilosc_zapisanych==0)  
            cout<<"Stos = [ ] jest pusty";  
        else  
        {  
            cout<<"Stos = [ ";  
            for(int i=0; i<ilosc_zapisanych; i++)  
                cout<<tablica_stosu[i] << " ";  
            cout<<"]";  
            if( ilosc_zapisanych==ROZMIAR_STOSU )  
                cout<<" stos jest zapelniony";  
        }  
  
        cout<<"\n\nOPERACJE NA STOSIE\nw - Wstaw\np - Pobierz\nESC - Zakoncz";  
        cout<<"\n\nPODAJ LITERE OPCJI: ";  
        klawisz = tolower(getch());  
        switch(klawisz)  
        {  
            case 'w': // wczytania liczby z klawiatury  
                cout<<"\n\nPodaj liczbe do umieszczenia na stosie = ";  
                double pomocnicza;  
                cin>>pomocnicza;  
  
                //operacja wstawienia nowej liczby na stos  
                if(ilosc_zapisanych>=ROZMIAR_STOSU)  
                {  
                    cout<<"\n\nBrak miejsca. Nacisnij klawisz";  
                    getch();  
                }  
                else  
                    tablica_stosu[ilosc_zapisanych++]=pomocnicza;  
  
                break;
```

```

case 'p': //operacja pobrania liczby ze stosu i wyswietlenia
    if(ilosc_zapisanych==0)
    {
        cout<<"\n\nStos jest pusty. Nacisnij klawisz";
        getch();
    }
    else
    {
        cout<<"\n\nLiczba ze stosu = ";
        cout<<tablica_stosu[ilosc_zapisanych---1];
        cout<<"\nNacisnij dowolny klawisz";
        getch();
    }

    break;
case ESC: break;
default: cout<<"\n\nNiewlasciwa litera. Spróbuj jeszcze raz";
        klawisz=getch(); ungetch(klawisz);
        break;
    }
}
while(klawisz!=ESC);

cout<<"\n\nKoniec programu. Nacisnij klawisz.";
getch();
}

```

// PRZYKŁAD p_21.cpp

// tablicowa implementacja "stosu"

// z interfejsem wykorzystującym funkcje:

// **PUSTY, PELNY, WSTAW, POBIERZ, WYSWIETL**

// Tu powinien być umieszczony nieformalny (słowny) opis interfejsu
// do wykorzystywanego modelu danych

// . . .

// . . . - znaczenie, parametry i wyniki funkcji

// . . .

//### FORMALNY OPIS INTERFEJSU i PRZYJĘTEJ REPREZENTACJI DANYCH ###

int PUSTY(double tab[], int rozmiar, int zapisane);

int PELNY(double tab[], int rozmiar, int zapisane);

void WYSWIETL(double tab[], int rozmiar, int zapisane);

**int WSTAW(double tab[], int rozmiar, int &zapisane,
double nowa_liczba);**

**int POBIERZ(double tab[], int rozmiar, int &zapisane,
double &pobrana_liczba);**

#define ROZMIAR_STOSU 10

double tablica_stosu[ROZMIAR_STOSU];

int ilosc_zapisanych=0;

//##### PRZYKŁADOWY PROGRAM #####

#include <iostream.h>

#include <conio.h> // clrscr(), getch(), ungetch()

#include <ctype.h> // tolower()

void main()

{

#define ESC 27

char klawisz;

do

{

clrscr();

WYSWIETL(tablica_stosu,ROZMIAR_STOSU,ilosc_zapisanych);

cout<<"\n\nOPERACJE NA STOSIE\nw - Wstaw\np - Pobierz\nESC - Zakoncz";

cout<<"\n\nPODAJ LITERE OPCJI: ";

klawisz = tolower(getch());

switch(klawisz)

{

case 'w': { // wczytania liczby z klawiatury

cout<<"\n\nPodaj liczbe do umieszczenia na stosie = ";

double pomocnicza;

cin>>pomocnicza;

**WSTAW(tablica_stosu,ROZMIAR_STOSU,ilosc_zapisanych,
pomocnicza);**

break;}

```

    case 'p': { //operacja pobrania liczby ze stosu i wyswietlenia
        double pomocnicza;
        if( POBIERZ(tablica_stosu,ROZMIAR_STOSU,
            ilosc_zapisanych,pomocnicza) )
            {
                cout<<"\n\nLiczba ze stosu = "<<pomocnicza;
                cout<<"\nNacisnij dowolny klawisz";
                getch();
            }
            break;}
        case ESC: break;
    default: cout<<"\7\7\7\n\nNiewlasciwa litera. Sprobuj jeszcze raz";
        klawisz=getch(); ungetch(klawisz);
        break;
    }
}
while(klawisz!=ESC);

cout<<"\n\nKoniec programu. Nacisnij klawisz.";
getch();
} //-----

//##### PRZYJETA IMPLEMENTACJA INTERFEJSU dla STOSU #####

int PUSTY(double /*tab*/[], int /*rozmiar*/, int zapisane)
{
    return zapisane==0;
} //-----

int PELNY(double /*tab*/[], int rozmiar, int zapisane)
{
    return zapisane>=rozmiar;
} //-----

void WYSWIETL(double tab[], int rozmiar, int zapisane)
{
    if( zapisane==0 )
        cout<<"Stos = [ ] jest pusty";
    else
        {
            cout<<"Stos = [ ";
            for(int i=0; i<zapisane; i++)
                cout<<tab[i] << " ";
            cout<<"]";
            if( zapisane==rozmiar )
                cout<<" stos jest zapelniony";
        }
} //-----

```

```

int WSTAW(double tab[], int rozmiar, int &zapisane,
          double nowa_liczba)
{
    if( PELNY(tab,rozmiar,zapisane) )
        {
            cout<<"\7\n\nNiestety, brak miejsca. Nacisnij klawisz";
            getch();
            return 0;
        }
    else
        tab[zapisane++]=nowa_liczba;
    return 1;
}//-----

int POBIERZ(double tab[], int rozmiar, int& zapisane,
            double &pobrana_liczba)
{
    if( PUSTY(tab,rozmiar,zapisane) )
        {
            cout<<"\7\n\nNiestety, stos jest pusty. Nacisnij klawisz";
            getch();
            return 0;
        }
    else
        {
            pobrana_liczba = tab[zapisane-- -1];
            return 1;
        }
}//-----

```

```

// PRZYKŁAD p_22.cpp
// "Strukturalna" implementacja operacji na stosie
// z interfejsem wykorzystującym operacje (funkcje):
//     PUSTY, PELNY, WSTAW, POBIERZ, WYSWIETL

// Tu powinien być umieszczony NIEFORMALNY (słowny) OPIS interfejsu
// do wykorzystywanego modelu danych
// . . .
// . . .     -   znaczenie, parametry i wyniki funkcji
// . . .

//### FORMALNY OPIS INTERFEJSU i PRZYJĘTEJ REPREZENTACJI DANYCH ###

#define ROZMIAR_STOSU  10
struct STOS
{
    double  tablica_stosu[ROZMIAR_STOSU];
    int     ilosc_zapisanych;
};

void INICJUU(  STOS &s );
int  PUSTY(   STOS  s );
int  PELNY(   STOS  s );
void WYSWIETL( STOS  s );
int  WSTAW(   STOS &s, double  nowa_liczba);
int  POBIERZ( STOS &s, double &pobrana_liczba);

//##### PRZYKŁADOWY PROGRAM #####

#include <iostream.h>
#include <conio.h>      // clrscr(), getch(), ungetch()
#include <ctype.h>     // tolower()

void main()
{
    //utworzenie zmiennej modelującej stos
    STOS moj_stos;

    INICJUU( moj_stos );

#define ESC 27
    char klawisz;
    do
    {
        clrscr();

        WYSWIETL( moj_stos );

        cout<<"\n\nOPERACJE NA STOSIE\nw - Wstaw\np - Pobierz\nESC - Zakoncz";
        cout<<"\n\nPODAJ LITERE OPCJI: ";
        klawisz = tolower(getch());
    }
}

```

```

switch(klawisz)
{
    case 'w': { // wczytania liczby z klawiatury
        cout<<"\n\nPodaj liczbe do umieszczenia na stosie = ";
        double pomocnicza;
        cin>>pomocnicza;

        WSTAW( moj_stos, pomocnicza );

        break;}
    case 'p': { //operacja pobrania liczby ze stosu i wyswietlenia
        double pomocnicza;

        if( POBIERZ( moj_stos,pomocnicza ) )
            {
                cout<<"\n\nLiczba ze stosu = "<<pomocnicza;
                cout<<"\nNacisnij dowolny klawisz";
                getch();
            }
        break;}
    case ESC: break;
    default: cout<<"\7\7\7\n\nNiewlasciwa litera. Spróbuj jeszcze raz";
        klawisz=getch(); ungetch(klawisz);
        break;
    }
}
while(klawisz!=ESC);

cout<<"\n\nKoniec programu. Nacisnij klawisz.";
getch();
} //-----

```

//##### PRZYJETA IMPLEMENTACJA INTERFEJSU dla STOSU #####

```

void INICJUI( STOS &s )
{ s.ilosc_zapisanych=0; }

```

```

int PUSTY( STOS s )
{ return s.ilosc_zapisanych==0; }

```

```

int PELNY( STOS s )
{ return s.ilosc_zapisanych >= ROZMIAR_STOSU; }

```

```

void WYSWIETL( STOS s )
{
    if( PUSTY( s ) )
        cout<<"Stos = [ ] jest pusty";
    else
    {
        cout<<"Stos = [ ";
        for(int i=0; i<s.ilosc_zapisanych; i++)
            cout << s.tablica_stosu[i] << " ";
        cout<<"]";
        if( PELNY( s ) )
            cout<<" stos jest zapelniony";
    }
}

}

int WSTAW( STOS &s, double nowa_liczba)
{
    if( PELNY( s ) )
    {
        cout<<"\n\nNiestety, brak miejsca. Nacisnij klawisz";
        getch();
        return 0;
    }
    else
        s.tablica_stosu[s.ilosc_zapisanych++]=nowa_liczba;
    return 1;
}

int POBIERZ( STOS &s, double &pobrana_liczba )
{
    if( PUSTY( s ) )
    {
        cout<<"\n\nNiestety, stos jest pusty. Nacisnij klawisz";
        getch();
        return 0;
    }
    else
        pobrana_liczba = s.tablica_stosu[s.ilosc_zapisanych-- -1];
    return 1;
}

```



```

// PRZYKLAD p_24.cpp
// "Uogólniona - szablonowa" implementacja operacji na stosie
// z interfejsem wykorzystującym operacje (metody):
// PUSTY, PELNY, WSTAW, POBIERZ, WYSWIETL

// Tu powinien być umieszczony NIEFORMALNY (słowny) OPIS interfejsu
// do wykorzystywanego modelu danych
// . . .
// . . . - znaczenie, parametry i wyniki funkcji
// . . .

//### FORMALNY OPIS INTERFEJSU i PRZYJĘTEJ REPREZENTACJI DANYCH ###

template <class ELEMENT>
struct STOS
{
    //reprezentacja danych
    #define ROZMIAR_STOSU 10
    ELEMENT tablica_stosu[ROZMIAR_STOSU];
    int ilosc_zapisanych;

    //metody (funkcje własne) implementujące interfejs
    void INICJUZ(void);
    int PUSTY(void);
    int PELNY(void);
    void WYSWIETL(void);
    int WSTAW(ELEMENT nowy_element);
    int POBIERZ(ELEMENT &pobrany_element);
};

//##### PRZYKŁADOWY PROGRAM #####
#include <iostream.h>
#include <conio.h>

//utworzenie obiektu (zmiennej) modelującej stos liczb "double"
STOS<double> moj_stos;

//utworzenie obiektu (zmiennej) modelującej stos struktur "Student"
struct Student
{
    char nazwisko[30];
    int wiek;
    char plec;
};

STOS<Student> moj_stos_struktur_Student;

ostream& operator<<(ostream& o, Student& s){
    o << "\nNazwisko: " << s.nazwisko;
    o << "\nWiek: " << s.wiek << "\nPlec: " << s.plec;
    return o;
}

```

```

//##### PRZYJETA IMPLEMENTACJA INTERFEJSU dla STOSU #####

template <class ELEMENT>
void STOS<class ELEMENT>::INICJUIJ(void)
{ ilosc_zapisanych=0; }

template <class ELEMENT>
int STOS<class ELEMENT>::PUSTY(void)
{ return ilosc_zapisanych==0; }

template <class ELEMENT>
int STOS<class ELEMENT>::PELNY(void)
{ return ilosc_zapisanych >= ROZMIAR_STOSU; }

template <class ELEMENT>
void STOS<class ELEMENT>::WYSWIETL(void)
{
    if( PUSTY() )
        cout<<"Stos = [ ] jest pusty";
    else
    {
        cout<<"Stos = [ ";
        for(int i=0; i<ilosc_zapisanych; i++)
            cout<< tablica_stosu[i] << " "; // UWAGA !!! tu moze wystapic
                                           //      problem z wydrukiem

        cout<<"]";
        if( PELNY() )
            cout<<" stos jest zapelniony";
    }
}

//-----

template <class ELEMENT>
int STOS<class ELEMENT>::WSTAW(ELEMENT nowy_element)
{
    if( PELNY() )
    {
        cout<<"\7\n\nNiestety, brak miejsca. Nacisnij klawisz";
        getch();
        return 0;
    }
    else
        tablica_stosu[ilosc_zapisanych++]=nowy_element;
    return 1;
}

//-----

```

```
template <class ELEMENT>
int STOS<class ELEMENT>::POBIERZ(ELEMENT &pobrany_element)
{
    if( PUSTY() )
    {
        cout<<"\n\nNiestety, stos jest pusty. Nacisnij klawisz";
        getch();
        return 0;
    }
    else
        pobrany_element = tablica_stosu[ilosc_zapisanych-- -1];
    return 1;
}//-----
```