



Fundusze Europejskie
dla Rozwoju Społecznego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Politechnika Białostocka
Wydział Elektryczny
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki

Instrukcja
do pracowni specjalistycznej z przedmiotu

Informatyka 1

Kod przedmiotu: **EMS1A1004**

(studia stacjonarne)

JĘZYK C - TABLICE DWU- I WIELOWYMIAROWE, OPERACJE NA TABLICACH

Numer ćwiczenia

INF1_E06

Autor:
dr inż. Jarosław Forenc

Białystok 2025

Spis treści

1. Opis stanowiska	3
1.1. Stosowana aparatura	3
1.2. Oprogramowanie	3
2. Wiadomości teoretyczne.....	3
2.1. Deklaracja tablicy dwuwymiarowej	3
2.2. Inicjalizacja elementów tablicy	4
2.3. Podstawowe operacje na macierzach	6
2.4. Tablice wielowymiarowe	12
3. Przebieg ćwiczenia.....	14
4. Literatura.....	16
5. Pytania kontrolne	17
6. Wymagania BHP	17

Materiały przygotowane w ramach projektu „PB 5.0 - dostosowanie oferty dydaktycznej Politechniki Białostockiej do potrzeb nowoczesnej gospodarki oraz zielonej i cyfrowej transformacji” (nr umowy FERS.01.05-IP.08-0327/23-00) w ramach programu Fundusze Europejskie dla Rozwoju Społecznego 2021-2027 współfinansowanego ze środków Europejskiego Funduszu Społecznego Plus.

Publikacja jest udostępniona na licencji Creative Commons - Uznanie autorstwa 4.0 (CC BY 4.0). Pełna treść licencji dostępna na stronie creativecommons.org.

1. Opis stanowiska

1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows 10/11.

1.2. Oprogramowanie

Na komputerach zainstalowany jest edytor kodu źródłowego Visual Studio Code 1.103 (lub nowszy) wraz z odpowiednimi rozszerzeniami (C/C++, Code Runner, Polish Language Pack for Visual Studio Code) oraz MinGW - zestaw kompilatorów różnych języków programowania (m.in. C, C++, Fortran, Java).

2. Wiadomości teoretyczne

2.1. Deklaracja tablicy dwuwymiarowej

Deklaracja tablicy dwuwymiarowej (macierzy) wymaga podania liczby wierszy i liczby kolumn, z których będzie składała się ta tablica.

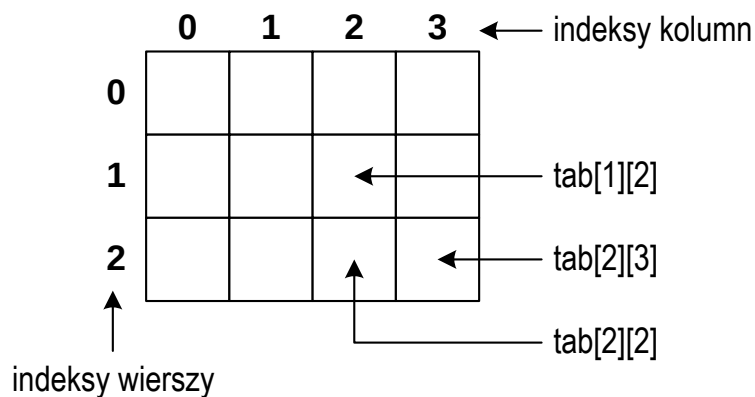
`float tab[1][4];`

- średnik
- liczba kolumn tablicy
- liczba wierszy tablicy
- nazwa tablicy
- typ elementów tablicy

Powyższa deklaracja definiuje tablicę dwunastu elementów typu **float**, składającą się z trzech wierszy i czterech kolumn (Rys. 1). Oba wymiary (liczba wierszy i liczba kolumn) muszą być wartościami całkowitymi, dodatnimi i znanymi na etapie kompilacji programu.

Odwołując się do elementów tablicy dwuwymiarowej należy podać dwa indeksy: numer wiersza i numer kolumny, na przecięciu których znajduje się dany

element (Rys. 1). Jako indeks może występować stała liczbowa, nazwa zmiennej przechowującej liczbę całkowitą lub wyrażenie dające w wyniku liczbę całkowitą.



Rys. 1. Tablica o rozmiarze 3×4

Zapisanie wartości **10** do elementu **tab[1][2]** oraz odczytanie tego elementu i przypisanie jego wartości zmiennej o nazwie **x** ma następującą postać:

```
tab[1][2] = 10;  
x = tab[1][2];
```

Przy odwołaniach do elementów tablicy kompilator nie sprawdza, czy zapis i odczyt dotyczy obszaru pamięci przydzielonego na tablicę (kompilator nie sprawdza poprawności indeksów).

2.2. Inicjalizacja elementów tablicy

Po zadeklarowaniu tablicy, wartości jej elementów są nieokreślone. Inicjalizacja polega na umieszczeniu w deklaracji po znaku równości, ujętej w nawiasy klamrowe, listy wartości kolejnych jej elementów. Poszczególne elementy tablicy oddzielane są od siebie przecinkami. Wiersze tablicy wyróżniane są dodatkowymi nawiasami klamrowymi. Deklaracja:

```
int tab[2][3] = {{3,6,2},{4,1,0}};
```

powoduje wypełnienie tablicy w sposób przedstawiony na Rys. 2.

	0	1	2
0	3	6	2
1	4	1	0

Rys. 2. Tablica z zainicjalizowanymi wartościami

Nawiasy klamrowe wyróżniające elementy kolejnych wierszy tablicy mogą być pominięte.

```
int tab[2][3] = {3,6,2,4,1,0};
```

Powyższa postać inicjalizacji pokazuje sposób w jaki tablice dwuwymiarowe przechowywane są w pamięci komputera (wiersz za wierszem).

Jeśli wartości podanych w trakcie inicjalizacji jest mniej niż wynosi rozmiar tablicy, to pozostałe elementy tablicy wypełniane są zerami (Rys. 3), np.

```
int tab[2][3] = {3,6,2,4};
```

	0	1	2
0	3	6	2
1	4	0	0

↑ ↑
elementy
uzupełnione zerami

Rys. 3. Tablica z elementami domyślnie zainicjalizowanymi wartością 0

Domyślna inicjalizacja pominiętych elementów tablicy pozwala w bardzo prosty sposób wypełnić całą tablicy wartością zero. Należy podać wartość tylko pierwszego elementu tablicy (zero), zaś pozostałym elementom automatycznie zostanie przypisana wartość zero.

```
int tab[2][3] = {0};
```

Jeśli deklaracja tablicy dwuwymiarowej powiązana jest z jej inicjalizacją, to można pominąć pierwszy wymiar tablicy (liczbę wierszy). Wymiar ten zostanie automatycznie określony przez kompilator na podstawie inicjalizacji.

```
int tab[][3] = {{3,6,2},{4,1,0}};
```

2.3. Podstawowe operacje na macierzach

W poniższym programie przedstawiono najczęściej wykonywane operacje na tablicy dwuwymiarowej (macierzy) przechowującej liczby całkowite.

Program wykonujący wybrane operacje na macierzy liczb całkowitych.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define N 3      /* liczba wierszy */
#define M 3      /* liczba kolumn */

int main(void)
{
    int  tab[N][M], i, j, max, suma, suma1, suma2;
    float srednia;

    /* generowanie pseudolosowe elementow macierzy */

    srand((unsigned int) time(NULL));
    for (i=0; i<N; i++)
        for (j=0; j<M; j++)
            tab[i][j] = rand() % 10;

    /* wyswietlenie elementow macierzy */

    for (i=0; i<N; i++)
    {
        for (j=0; j<M; j++)
            printf("%3d", tab[i][j]);
        printf("\n");
    }
    printf("\n");
```

```
/* poszukiwanie elementu o wartosci maksymalnej */
```

```
max = tab[0][0];  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        if (tab[i][j] > max)  
            max = tab[i][j];  
printf("Wartosc max: %d\n\n",max);
```

```
/* suma i srednia arytmetyczna elementow */
```

```
suma = 0;  
for (i=0; i<N; i++)  
    for (j=0; j<M; j++)  
        suma = suma + tab[i][j];  
srednia = (float) suma/(N*M);  
printf("Suma elementow:      %d\n",suma);  
printf("Srednia arytmetyczna: %f\n\n",srednia);
```

```
/* sumy elementow w poszczegolnych wierszach */
```

```
for (i=0; i<N; i++)  
{  
    suma = 0;  
    for (j=0; j<M; j++)  
        suma = suma + tab[i][j];  
    printf("Suma wiersza %d = %d\n",i,suma);  
}  
printf("\n");
```

```
/* sumy elementow w poszczegolnych kolumnach */
```

```
for (j=0; j<M; j++)  
{  
    suma = 0;  
    for (i=0; i<N; i++)  
        suma = suma + tab[i][j];  
    printf("Suma kolumny %d = %d\n",j,suma);  
}  
printf("\n");
```

```

/* sumy elementow nad, na i ponizej przekatnej */

suma = suma1 = suma2 = 0;
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
    {
        if (i < j) suma1+=tab[i][j]; /* nad */
        if (i == j) suma+=tab[i][j]; /* na */
        if (i > j) suma2+=tab[i][j]; /* pod */
    }

printf("Suma nad przekatna: %d\n",suma1);
printf("Suma na przekatnej: %d\n",suma);
printf("Suma pod przekatna: %d\n",suma2);

return 0;
}

```

Przykładowy wynik działania programu:

```

1  7  6
3  5  2
4  8  4

```

Wartosc max: 8

Suma elementow: 40
Srednia arytmetyczna: 4.444445

Suma wiersza 0 = 14
Suma wiersza 1 = 10
Suma wiersza 2 = 16

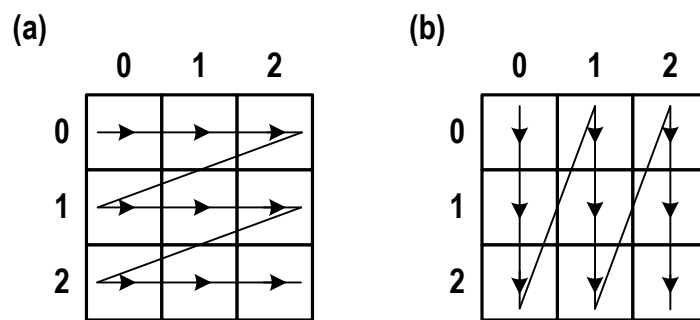
Suma kolumny 0 = 8
Suma kolumny 1 = 20
Suma kolumny 2 = 12

Suma nad przekatna: 15
Suma na przekatnej: 10
Suma pod przekatna: 15

W programie wykonywane są następujące operacje na tablicy:

- zapisanie wygenerowanych pseudolosowo liczb całkowitych z zakresu $\langle 0, 9 \rangle$; zewnętrzna pętla **for** określa indeks wiersza (*i*), zaś pętla wewnętrzna - indeks kolumny (*j*); macierz jest zatem wypełniana liczbami wiersz po wierszu (Rys. 4a):

```
srand((unsigned int) time(NULL));
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
        tab[i][j] = rand() % 10;
```



Rys. 4. Przeglądanie macierzy: (a) wiersz po wierszu, (b) kolumna po kolumnie

Macierz może być także wypełniana kolumnami (Rys. 4b). Wystarczy wtedy zamienić miejscami pętle **for** - pierwsza pętla powinna określać indeks kolumny (*j*), a druga pętla - indeks wiersza (*i*).

- wyświetlenie elementów z podziałem na wiersze i kolumny; po wyświetleniu elementów jednego wiersza w pętli wewnętrznej, następuje przejście do nowego wiersza:

```
for (i=0; i<N; i++)
{
    for (j=0; j<M; j++)
        printf("%3d", tab[i][j]);
    printf("\n");
}
printf("\n");
```

- wyszukanie elementu o największej wartości - zakładamy, że element **tab[0][0]** jest największy (**max = tab[0][0]**); przeglądamy pozostałe elementy macierzy (wierszami); jeśli kolejny z elementów macierzy (**tab[i][j]**) jest większy od dotychczasowego największego (**max**), to element ten staje się największym (**max = tab[i][j]**); po zakończeniu obu pętli wyświetlamy na ekranie wartość elementu największego (**max**):

```
max = tab[0][0];
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
        if (tab[i][j] > max)
            max = tab[i][j];
printf("Wartosc max: %d\n\n",max);
```

- obliczenie sumy i średniej arytmetycznej elementów macierzy - w dwóch pętlach **for** dodajemy kolejne elementy macierzy do zmiennej **suma** (przed pętlą zmienna ta musi być wyzerowana); następnie obliczamy średnią arytmetyczną dzieląc **sumę** przez liczbę elementów (**N*M**); ponieważ **suma** i liczba elementów są typu całkowitego, to w celu uniknięcia dzielenia liczby całkowitej przez liczbę całkowitą, wymuszamy zmianę typu zmiennej **suma** na **float**: (**float**) **suma**:

```
suma = 0;
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
        suma = suma + tab[i][j];
srednia = (float) suma/(N*M);
printf("Suma elementow: %d\n",suma);
printf("Srednia arytmetyczna: %f\n\n",srednia);
```

- obliczenie sum elementów w poszczególnych wierszach - pętla zewnętrzna określa numer wiersza (**i**); w pętli tej wykonywane są trzy operacje: wyzerowanie zmiennej **suma**, sumowanie elementów **i**-tego wiersza w wewnętrznej pętli **for**, wyświetlenie sumy **i**-tego wiersza:

```

for (i=0; i<N; i++)
{
    suma = 0;
    for (j=0; j<M; j++)
        suma = suma + tab[i][j];
    printf("Suma wiersza %d = %d\n",i,suma);
}
printf("\n");

```

- obliczenie sum elementów w poszczególnych kolumnach - wykonywane jest w podobny sposób jak obliczanie sum wierszy; najważniejsza zmiana polega na zamianie miejscami pętli zewnętrznej i wewnętrznej; pętla zewnętrzna określa numer kolumny (**j**); w pętli tej wykonywane są trzy operacje: wyzerowanie zmiennej **suma**, sumowanie elementów **j**-tej kolumny w wewnętrznej pętli **for**, wyświetlenie sumy **j**-tej kolumny:

```

for (j=0; j<M; j++)
{
    suma = 0;
    for (i=0; i<N; i++)
        suma = suma + tab[i][j];
    printf("Suma kolumny %d = %d\n",j,suma);
}
printf("\n");

```

- obliczenie sumy elementów nad, na i poniżej głównej przekątnej macierzy - do przechowywania poszczególnych sum stosowane są trzy zmienne (**suma**, **suma1**, **suma2**); w dwóch pętlach **for** przeglądane są wierszami wszystkie elementy macierzy; jeśli indeks wiersza (**i**) elementu macierzy jest mniejszy od indeksu kolumny (**j**), to element ten znajduje się nad główną przekątną macierzy; jeśli indeks wiersza (**i**) elementu macierzy jest większy od indeksu kolumny (**j**), to element ten znajduje się poniżej głównej przekątnej macierzy; jeśli indeksy są sobie równe (**i == j**), to element znajduje się na głównej przekątnej macierzy:

```

suma = suma1 = suma2 = 0;
for (i=0; i<N; i++)
    for (j=0; j<M; j++)
    {
        if (i < j) suma1+=tab[i][j]; /* nad */
        if (i == j) suma+=tab[i][j]; /* na */
        if (i > j) suma2+=tab[i][j]; /* pod */
    }

printf("Suma nad przekatna: %d\n", suma1);
printf("Suma na przekatnej: %d\n", suma);
printf("Suma pod przekatna: %d\n", suma2);

```

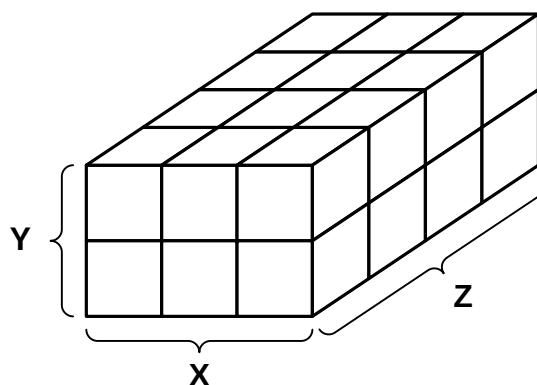
2.4. Tablice wielowymiarowe

Ogólna postać deklaracji tablicy wielowymiarowej jest następująca:

```
typ nazwa[wymiar_1][wymiar_2]...[wymiar_N]
```

Sposób inicjalizacji i odwoływania się do elementów tablic wielowymiarowych jest analogiczny jak w przypadku tablic dwuwymiarowych.

W poniższym przykładzie pokazano deklarację i inicjalizację tablicy mającej trzy wymiary ($X \times Y \times Z \rightarrow 3 \times 2 \times 4$). Interpretację graficzną tablicy przedstawia Rys. 5.



Rys. 5. Tablica trójwymiarowa

Deklaracja, inicjalizacja i wyświetlenie elementów tablicy trójwymiarowej.

```
#include <stdio.h>
#define X 3
#define Y 2
#define Z 4

int main(void)
{
    int tab[Z][Y][X] = {{{9,5,7},{5,9,6}},
                        {{0,1,3},{7,4,3}},
                        {{8,5,9},{1,3,5}},
                        {{6,0,1},{8,2,5}}};

    int x, y, z;

    for(z=0; z<Z; z++)
    {
        for(y=0; y<Y; y++)
        {
            for(x=0; x<X; x++)
                printf("%3d",tab[z][y][x]);
            printf("\n");
        }
        printf("\n");
    }

    return 0;
}
```

Wynik działania programu:

```
9  5  7
5  9  6
```

```
0  1  3
7  4  3
```

```
8  5  9
1  3  5
```

```
6  0  1
8  2  5
```

3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

1. Zadeklaruj **N×M** - elementową tablicę liczb całkowitych typu **int** (**N**, **M** - stałe zadeklarowane dyrektywą preprocesora **#define**). Wykonaj następujące operacje:
 - a) zapisz do tablicy (wierszami) kolejne liczby całkowite **1, 2, 3, ...**; wyświetl elementy tablicy z podziałem na wiersze i kolumny;
 - b) zapisz do tablicy (kolumnami) kolejne liczby całkowite **1, 2, 3, ...**; wyświetl elementy tablicy z podziałem na wiersze i kolumny;
 - c) zapisz do tablicy wygenerowane pseudolosowo liczby całkowite z zakresu **<-9, 9>**; wyświetl elementy tablicy z podziałem na wiersze i kolumny;
 - d) oblicz i wyświetl liczbę elementów tablicy: dodatnich, ujemnych, równych zero;
 - e) oblicz i wyświetl liczbę elementów tablicy o parzystych i nieparzystych wartościach;
 - f) zastąp wszystkie ujemne elementy tablicy wartością zero; wyświetl elementy tablicy z podziałem na wiersze i kolumny;
 - g) oblicz i wyświetl średnią arytmetyczną tylko tych elementów tablicy, które są większe od zera.
2. Tablica **P** przechowuje wyniki **100** pomiarów wartości chwilowych napięcia (pierwszy wiersz tablicy) i prądu (drugi wiersz tablicy) na pewnym dwójniku RLC. Napisz program, który:
 - a) zapisze do pierwszego wiersza tablicy wartości chwilowe napięcia zgodnie ze wzorem:

$$10.0 * \sin((i + 10.0) / 15.0) \quad i = 0..99 \quad (1)$$

- b) zapisze do drugiego wiersza tablicy wartości chwilowe prądu zgodnie ze wzorem:

$$5.0 * \sin(i/15.0) \qquad i = 0..99 \qquad (2)$$

- c) zapisze do trzeciego wiersza tablicy wartości chwilowe mocy ($p = u \cdot i$);
- d) wypisze zawartość tablicy - wartości chwilowe napięcia, prądu i mocy;
- e) obliczy i wypisze wartości średnie napięcia, prądu i mocy;
- f) wypisze numer pomiaru, w którym wartość napięcia była największa oraz numer pomiaru, w którym wartość prądu była największa;
- g) wypisze liczbę pomiarów, dla których moc chwilowa była dodatnia i liczbę pomiarów, dla których moc chwilowa była ujemna.
3. Tablica **REK** przechowuje informacje dotyczące rekrutacji na studia wyższe. W tablicy znajduje się **N** wierszy i **4** kolumny. Każdy wiersz zawiera informacje o jednym kandydacie. W pierwszych trzech kolumnach umieszczone są oceny kandydatów. Na studia przyjmowane są osoby, których średnia arytmetyczna trzech ocen jest większa lub równa średniej arytmetycznej wszystkich ocen wszystkich kandydatów. Napisz program, który:
- a) zapisze do tablicy wygenerowane losowo oceny wszystkich kandydatów (dopuszczalne oceny to: 2, 3, 4, 5, 6);
- b) obliczy i zapisze w czwartej kolumnie średnie arytmetyczne ocen poszczególnych kandydatów;
- c) wyświetli zawartość całej tablicy;
- d) obliczy średnią arytmetyczną wszystkich ocen kandydatów;
- e) wyświetli numery kandydatów przyjętych na studia (numer kandydata to numer wiersza w tablicy **REK**) oraz średnie arytmetyczne ich ocen.
4. Napisz program, który dla **N×M** - elementowej tablicy liczb całkowitych wygeneruje pseudolosowo elementy tablicy z zakresu $\langle 0, 10 \rangle$. Następnie wyświetli zawartość tablicy z podziałem na wiersze i kolumny oraz obliczy ile razy każda liczba występuje w tablicy, wyświetlając tylko te liczby, których ilość jest różna od zera.

5. Napisz program wykonujący operację mnożenia macierzy $N \times M$ - elementowej przez macierz $M \times K$ - elementową. Elementy macierzy wygeneruj pseudolosowo. Wyświetl elementy wszystkich macierzy z podziałem na wiersze i kolumny.
6. Napisz program, który dla $N \times M$ - elementowej tablicy liczb całkowitych wygeneruje pseudolosowo liczby z zakresu $\langle 0, 10 \rangle$, a następnie odnajdzie taki element tablicy, którego suma sąsiadów jest największa. Jako sąsiadów należy rozumieć elementy znajdujące się po lewej stronie, po prawej stronie, powyżej i poniżej danego elementu. Wyświetl zawartość tablicy z podziałem na wiersze i kolumny oraz indeksy odnalezionego elementu.
7. Napisz program, który dla $N \times M \times K$ - elementowej tablicy liczb całkowitych:
 - a) wygeneruje pseudolosowo liczby z zakresu $\langle 0, 99 \rangle$;
 - b) wyświetli na ekranie elementy tablicy;
 - c) znajdzie w tablicy element o największej wartości i poda indeksy tego elementu;
 - d) znajdzie w tablicy element o najmniejszej wartości i poda indeksy tego elementu.

4. Literatura

- [1] Prata S.: Język C. Szkoła programowania. Wydanie VI. Helion, Gliwice, 2016.
- [2] Kernighan B.W., Ritchie D.M.: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
- [3] Deitel P.J., Deitel H.: Język C. Solidna wiedza w praktyce. Wydanie VIII. Helion, Gliwice, 2020.
- [4] Kochan S.G.: Język C. Kompendium wiedzy. Wydanie IV. Helion, Gliwice, 2015.

- [5] King K.N.: Język C. Nowoczesne programowanie. Wydanie II. Helion, Gliwice, 2011.
- [6] Stańczyk J.: Nowoczesny C: przegląd C23 z przykładami. Helion, Gliwice, 2023.
- [7] <http://www.cplusplus.com/reference/clibrary> - C library - C++ Reference
- [8] <https://cpp0x.pl/dokumentacja/standard-C/1> - Standard C
- [9] <https://code.visualstudio.com/> - Visual Studio Code

5. Pytania kontrolne

1. Omów sposób deklarowania tablic dwuwymiarowych (macierzy) w języku C oraz odwoływania się do elementów tych tablic.
2. Opisz inicjalizację elementów tablicy dwuwymiarowej.
3. Opisz deklarację, inicjalizację i sposób odwoływania się do elementów tablic wielowymiarowych.

6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciw pożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.

- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.