

# C++

## Lab\_4

---

### Tablica obiektów

1. Utwórz nowy projekt. Dodaj plik *node\_coord.h* do projektu, w którym masz umieścić definicję klasy *NODE\_COORD*:

```
class NODE_COORD
{
    double *pcoord; //pcoord[0] - x, pcoord[1] - y
public:
    NODE_COORD() : pcoord(NULL) {cout << "default constructor: NODE_COORD\n";}
    NODE_COORD(double x, double y); //parameterized constructor
    ~NODE_COORD(); //destructor
    void set(double x, double y); //set x, y coordinates
    void disp(); //display x, y coordinates on screen

private:
    void crash(); //memory allocation error handler - call
                //this function if a memory allocation fails
};
```

Dodaj *node\_coord.cpp* i umieść wszystkie implementacje metod klas. Konstruktor klasy wyprowadzają na monitor informacje o działalności konstruktora, a destruktor – o działalności destruktor.

Funkcja *main()*:

```
int main ()
{
    const double math_pi = 3.141592;
    NODE_COORD tb[3] = { NODE_COORD(0, 0), NODE_COORD(1, 0), NODE_COORD(0, 1) };
    for(size_t i=0; i<3; ++i)
    {
        //wyświetl każdy element tablicy na monitorze
    }

    NODE_COORD * ptr = NULL;
    //Używając operatora new, dynamicznie zaalokuj pamięć dla tablicy typu NODE_COORD
    // o 10 elementów i przypisz jej adres do wskaźnika ptr.

    for(size_t i=0; i<10; ++i)
    {
        // obliczyć współrzędne x, y punktów równomiernie rozmieszczonych na okręgu
        // o promieniu r0 ze środkiem w początku układu współrzędnych.
        // Każdy element tablicy reprezentuje punkt leżący na tym okręgu.
    }

    fun(ptr); // utwórz funkcję, która wyświetli współrzędne wszystkich punktów
```

```

//leżących na okręgu

//zwolni pamięć zajmowaną tablicą.

system("pause");
return 0;
}

```

