

Zadanie 1

Sprawdź działanie operatorów `new` i `delete` dla tablicy klas z konstruktorem i destruktorom np.:

```
#include <iostream.h>

class A
{
public:
    A() { cout << "konstruktor A\n";}
    ~A() { cout << "destruktor A\n";}
};

int main()
{
    A *ptr;
    ptr = new A[5];
    delete [] ptr; // co będzie gdy usunie się nawiasy [] ?
    return 0;
}
```

- Dodaj do klasy A atrybut statyczny np. `static int mattr`.
- Sprawdź jakie wartości będą drukowane (dlaczego?) jeśli wykonasz:

```
for ( i=0; i<5; ++i)
    ptr[i].mattr = i+1;

for ( i=0; i<5; ++i)
    cout << " element " << i << " attr = " << ptr[i].mattr << endl;
```

Zadanie 2

Utwórz klasę bazową `Pojazd` opisującą pewien pojazd.

- Atrybut klasy to np. przebieg danego pojazdu (typu `int`) umieszczony w sekcji `private`

- Utwórz:
 - konstruktor domyślny: `Pojazd()` – przebieg zainicjuj zerem
 - konstruktor `Pojazd(const int& n)` – przebieg zainicjuj argumentem `n`
 - destruktor `~Pojazd()`
- W konstruktorach i destruktorach drukuj informację o ich wykonaniu tak aby można było stwierdzić co i kiedy zostało wywołane.
- Utwórz metodę `Przebieg()` zwracającą wartość przebiegu.
- Sprawdź poprawność tej klasy tworząc zmienną typu `Pojazd` i drukując wartość metody `Przebieg()`

Zadanie 3

Utwórz klasę `Autobus` która jest klasą pochodną klasy `Pojazd`.

- Atrybut klasy to np. ilość pasażerów (typu `int`) umieszczony w sekcji `private`
- Utwórz:
 - konstruktor domyślny: `Autobus()` – przebieg zainicjuj zerem
 - konstruktor `Autobus(const int& m)` – ilość pasażerów zainicjuj argumentem `m`
 - destruktor `~Autobus()`
- Podobnie jak dla klasy `Pojazd` w konstruktorach i destruktorach drukuj informację o ich wykonaniu.
- Utwórz metodę `IloscPaszalow()` zwracającą wartość atrybutu przechowującego ilość pasażerów.
- Jeśli utworzysz zmienną typu `Autobus` jakie konstruktory będą wywoływane? W jakiej kolejności będą wołane konstruktory i destruktory?
- Jaki przebieg ma zmienna typu `Autobus`?
- Dodaj nowy konstruktor do klasy `Autobus` który pozwoli również zainicjalizować przebieg pojazdu i sprawdź jego działanie.



- Co się stanie jeśli wykonasz poniższy kod i co zrobić aby uniknąć tego typu niejednoznaczności?

```
Autobus bus;  
bus = 3;
```

Zadanie 4

Zmodyfikuj program tak aby każda klasa była umieszczona w oddzielnym pliku .h i .cpp.