

- I. Przygotuj klasę `Osoba` z polami `char* imie` i `int wiek`. Następnie utwórz dwa obiekty zdefiniowanej klasy, używając konstruktorów wykorzystujących mechanizmy: dynamicznego alokowania pamięci oraz lokalnej definicji.
- II. Rozwiń definicję klasy `Osoba` o konstruktory:
- pusty – wprowadzający wartości pól do dynamicznie alokowanych zasobów,
 - przyjmujący dwa parametry (`char*`, `int`) – inicjujący pola przez listę inicjacyjną.

Oba z konstruktorów powinny wyświetlić na ekranie informacje o utworzeniu obiektu `Osoba`. Zdefiniuj destruktora, wyświetlający informacje o zniszczeniu obiektu i zwalnający dynamicznie zaalokowane zasoby.

W ciele funkcji `main` przedstaw tworzenie obiektów powyższej klasy z użyciem obu typów konstruktorów.

- III. Wykorzystując operator `new`, utwórz obiekt klasy `Osoba` i używając funkcji `show` wyświetl zawarte w nim pola. Utwórz odnośnik `ref` do tego obiektu, a następnie utwórz nowy obiekt klasy `Osoba` dostarczając jako parametr konstruktora odnośnika `ref`. Ponownie wyświetl stany pól wszystkich utworzonych obiektów. Usuń pierwszy z utworzonych obiektów i wyświetl stan pól pozostałego obiektu.
- IV. Zdefiniuj klasę `Rachunek` zawierającą pola `Osoba wlasciciel` (zdefiniowana na poprzednich zajęciach) oraz `double stan_konta`. Następnie utwórz klasę `Bank` przechowującą obiekty klasy `Rachunek` i pozwalającą na wywołanie funkcji:
- `otworzNowyRachunek` - tworzącą nowy rachunek dla wskazanego klienta,
 - `zmienWlasciciela` - modyfikującą Osobę będącą właścicielem rachunku,
 - `usunRachunek` - kasującą rachunek.

W programie należy zadbać aby wszystkie obiekty zawierały zdefiniowane i funkcjonalne:

- konstruktory,
- destruktory.

Przedstaw działanie programu na przykładzie Pana Tomka, który założył trzy rachunki i wpłacił na nie po 100 zł. Następnie jeden rachunek przepisał na małżonkę Małgorzatę, a drugi usunął. Przy każdej z tych operacji bank wyświetlał na konsoli wszystkie posiadane rachunki, wraz z imieniem klienta i stanem konta.