

Lab 6 – Zadanie

System obsługi produktów

Opis:

Napisz program w C++, który będzie symulować prosty system obsługi produktów. Program powinien umożliwiać dodawanie, usuwanie i wyświetlanie informacji o produktach. Informacje o produktach powinny być przechowywane w pliku binarnym.

Wymagania:

Stwórz klasę Product reprezentującą produkt. Każdy produkt powinien mieć następujące atrybuty:

- ID produktu (liczba całkowita)
- Nazwa produktu (łańcuch znaków)
- Cena produktu (liczba zmiennoprzecinkowa)

Stwórz klasę ProductManager, która będzie odpowiedzialna za zarządzanie produktami.

Powinna zawierać następujące metody:

- addProduct: dodaje nowy produkt do systemu
- removeProduct: usuwa produkt z systemu na podstawie ID
- displayProducts: wyświetla wszystkie produkty przechowywane w systemie

Obsłuż wyjątki w przypadku niepowodzenia operacji dodawania lub usuwania produktu (np. próba dodania produktu o już istniejącym ID).

Zaimplementuj obsługę zapisu i odczytu danych z pliku binarnego za pomocą strumieni*.

Wykorzystaj RTTI do dynamicznego określenia typu obiektów podczas wyświetlania informacji o produktach.

*Aby zapisać std::string do pliku binarnego należy najpierw zapisać jego długość, a potem surowe dane.

Klasy podziel na pliki – zasada 1 klasa 2 pliki, zzipuj i wstaw na Deltę

Jako pomoc możesz wykorzystać kod z następnej strony pamiętając o zasadzie podziału na pliki!

```

#include <iostream>
#include <fstream>
#include <vector>
#include <stdexcept>
#include <typeinfo>

class Product {
private:
    int id;
    std::string name;
    double price;
public:
    // konstruktor, getter, setter, itp.
};

class ProductManager {
private:
    std::vector<Product> products;
public:
    void addProduct(const Product& product);
    void removeProduct(int id);
    void displayProducts();
    void saveToFile(const std::string& filename);
    void loadFromFile(const std::string& filename);
};

void ProductManager::addProduct(const Product& product) {
    // implementacja dodawania produktu
    // obsługa wyjątków w przypadku konfliktu ID
}

void ProductManager::removeProduct(int id) {
    // implementacja usuwania produktu
    // obsługa wyjątków w przypadku braku produktu o podanym ID
}

void ProductManager::displayProducts() {
    // implementacja wyświetlania produktów
    // wykorzystanie RTTI do określenia typu obiektów
}

void ProductManager::saveToFile(const std::string& filename) {
    // implementacja zapisu do pliku binarnego
}

void ProductManager::loadFromFile(const std::string& filename) {
    // implementacja odczytu z pliku binarnego
}

int main() {
    ProductManager manager;

    // Testowanie funkcji

    return 0;
}

```