МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ

**ЗВІТ**

лабораторної роботи №1

з дисципліни «Об’єктно орієнтоване програмування»

на тему «Структури, файли»

**ВИКОНАВ:**

Студент 1-го курсу

Групи ФЕП-14

Іванський Остап Юрійович

**ПЕРЕВІРИВ:**

Асистент кафедри РКТ

Левуш Павло Назарович

Львів 2025

**МЕТА**: Повторити матеріал, засвоєний у курсі Програмування мовою С.

**ПОСИЛАННЯ**: <https://github.com/Ostap-Ivanskyi/OOP/tree/main/Lab2>

**ЗАВДАННЯ:**

Розв’язати наступну задачу, написавши виконуваний код мовою С++ у середовищі MS VS. Якщо з об’єктивних причин немає можливості використовувати MS VS, можна застосувати будь-яке інше середовище (тоді п.2 пропускається), інші завдання залишаються.

Як усім відомо, у нашій країні виведені з обігу монети до 10 коп. Але ціни залишились у старому форматі. У зв’язку з цим виникає така задача:

Кожна ціна складається з двох чисел, гривні та копійки. Для такої структури визначити операції додавання та множення на додатне число. Також створити метод, який буде заокруглювати кількість копійок за правилами Нацбанку.

Вхідними параметрами програми будуть ціни на товари та їх кількість, наприклад:

*Пакет молока 900г, 19 грн 89 коп, 3 шт;*

*Батон Нарізний, 13 грн 29 коп, 1 шт*

Для спрощення у текстовому файлі можуть знаходитись по 3 числа у стрічці без назви товару.

На виході має бути загальна сума чеку  та сума до оплати із заокругленням.

**ХІД РОБОТИ**

Створюю файл заголовок Price.h в якому об’являю структуру з полями та функціями додавання, множення, округлення та виводу.

Price.h

struct Price {  
 int hrn;  
 short int kop;  
};  
  
Price createPrice(int hrn, short kop);  
void normalizePrice(Price &price);  
Price addPrices (const Price& a,const Price& b);  
Price multiplyPrice (const Price& price,int m);  
  
void round(Price& price);  
void print(const Price& price);

У файлі Price.cpp описую реалізацію функцій.

Функція plus() дає змогу додавати дві структури та перевіряє чи копійки не перевищили 100.

Функція mult() дає змогу множити структуру на число та перевіряє чи копійки не перевищили 100.

Функція round() заокруглює копійки до десяток.

Функція print() виводить значення структури.

Price.cpp

Price createPrice(int hrn, short kop) {  
 Price price;  
 price.hrn = hrn;  
 price.kop = kop;  
 normalizePrice(price);  
 return price;  
}  
void normalizePrice(Price& price) {  
 if (price.kop >= 100) {  
 price.hrn += price.kop / 100;  
 price.kop %= 100;  
 }  
}  
  
Price addPrices(const Price& a, const Price& b) {  
 return createPrice(a.hrn + b.hrn, a.kop + b.kop);  
}  
  
Price multiplyPrice(const Price& price, int quantity) {  
 return createPrice(price.hrn \* quantity, price.kop \* quantity);  
}  
  
void round( Price& price) {  
 int rem = price.kop % 10;  
 if (rem >= 5) {  
 price.kop = price.kop + 10 - rem;  
 } else {  
 price.kop -= rem;  
 }  
 normalizePrice(price);  
}  
  
void print(const Price &price) {  
 std::cout << price.hrn << " hrn " << price.kop << " kop"<<"\n";  
}

У функції main відкриваю та зчитую інформацію з файлу. За допомогою функцій з структур рахую загальну вартість та округлюємо копійки.

main.cpp}

int main() {  
 FILE \*file = fopen("input.txt", "r");  
 if (!file) {  
 std::cerr << "Не вдалося відкрити файл!" << std::endl;  
 return 1;  
 }  
  
 Price total={0,0};  
 int hrn, quantity;  
 short int kop;  
  
 std::cout << "FILE" << std::endl;  
 while (fscanf(file, "%\*[^,], %d %\*s %d %\*[^,], %d;\n", &hrn, &kop, &quantity) == 3) {  
 std::cout << hrn << " hrn " << kop << " kop " << quantity << std::endl;  
 Price itemPrice = createPrice(hrn, kop);  
 Price totalItemPrice = multiplyPrice(itemPrice, quantity);  
 total = addPrices(total, totalItemPrice);  
 }  
  
 fclose(file);  
  
 std::cout << "Total: ";  
 print(total);  
  
 Price roundedTotal = total;  
 round(roundedTotal);  
  
 std::cout << "Rounded: ";  
 print(roundedTotal);  
  
 return 0;  
}

**ВИСНОВОК**

Під час виконання лабораторної роботи попрактикувався у роботі зі структурами і файлами. Повторив навчальний матеріал з попередніх курсів по роботі зі структурами.