Họ và tên: Trần Đăng Hiếu

Lớp: 64CNTT3

MSV: 2251061779

**BÀI 20: QUẢN LÝ TÍNH CÔNG BÁN HÀNG CHO SIÊU THỊ**

Khai báo lớp  ***Nhân viên bán hàng (mã NV, Họ tên, Địa chỉ, Số ĐT, Ngày hợp đồng),*** mã NV là một số nguyên có 4 chữ số, tự động tăng***.***

Khai báo lớp  ***Mặt hàng*** gồm các thuộc tính ***(mã hàng, Tên hàng, Nhóm hàng, Giá mua, Giá bán)*** – trong đó Nhóm hàng có thể là: Hàng thời trang, Hàng tiêu dùng, Hàng điện máy, Hàng gia dụng.  Mã hàng là một số nguyên có 4 chữ số, tự động tăng.

Khai báo lớp ***Bảng Danh sách bán hàng*** có quan hệ kết hợp (association) với lớp **Nhân viên bán hàng** và lớp **Mặt hàng** trong đó với mỗi nhân viên, nhập danh sách các mặt hàng và số lượng mà nhân viên đó đã bán ra trong ngày.

Viết chương trình trong ngôn ngữ C++ thực hiện các yêu cầu sau:

1. Nhập danh sách mặt hàng mới. In ra danh sách các mặt hàng đã có
2. Nhập danh sách nhân viên bán hàng.  In ra danh sách nhân viên đã có
3. Lập Bảng danh sách bán hàng cho từng nhân viên và in danh sách ra màn hình *(chú ý: cùng một nhân viên với một mặt hàng thì không thể xuất hiện 2 lần trong bảng này).*
4. Sắp xếp danh sách Bảng danh sách bán hàng

a. Theo tên nhân viên

b. Theo tên mặt hàng

5. Lập bảng tính công cho mỗi nhân viên. Giá trị tiền công được tính là 2% tổng số lợi nhuận thu được.

Source code

Class NhanVienBanHang

1. #include <iostream>
2. #include <string>
3. #include <iomanip>
5. **using** **namespace** std;
7. **class** NhanVienBanHang{
8. **private**:
9. **static** **int** MaNVId;
10. **int** MaNV;
11. string HoTen;
12. string DiaChi;
13. string SoDienThoai;
14. string NgayHopDong;
15. **public**:
16. NhanVienBanHang(){
17. **this** -> MaNV = MaNVId++;
18. }
19. NhanVienBanHang(string HoTen, string DiaChi, string SoDienThoai, string NgayHopDong): HoTen(HoTen), DiaChi(DiaChi), SoDienThoai(SoDienThoai), NgayHopDong(NgayHopDong){
20. **this** -> MaNV = MaNVId++;
21. }
22. string getHoTen(){
23. **return** HoTen;
24. }
25. **void** input();
26. **void** output();
27. };
29. **int** NhanVienBanHang::MaNVId = 1000;
31. **void** NhanVienBanHang::input(){
32. cout << endl << "Nhap ho ten: ";
33. getline(cin, HoTen);
34. cout << "Nhap dia chi: ";
35. getline(cin, DiaChi);
36. cout << "Nhap so dien thoai: ";
37. getline(cin, SoDienThoai);
38. cout << "Nhap ngay hop dong: ";
39. getline(cin, NgayHopDong);
40. }
42. **void** NhanVienBanHang::output(){
43. cout << left << setw(15) << MaNV << setw(30) << HoTen << setw(30) << DiaChi << setw(30) << SoDienThoai << setw(10) << NgayHopDong << endl;
44. }

Class MatHang

1. #include <iostream>
2. #include <string>
3. #include <iomanip>
5. **using** **namespace** std;
7. **class** MatHang{
8. **private**:
9. **static** **int** MaHangID;
10. **int** MaHang;
11. string TenHang;
12. string NhomHang;
13. **double** GiaMua;
14. **double** GiaBan;
15. **public**:
16. MatHang(){
17. **this** -> MaHang = MaHangID++;
18. }
19. MatHang(string TenHang, string NhomHang, **double** GiaMua, **double** GiaBan): TenHang(TenHang), NhomHang(NhomHang), GiaMua(GiaMua), GiaBan(GiaBan){
20. **this** -> MaHang = MaHangID++;
21. }
22. string getTenHang(){
23. **return** TenHang;
24. }
25. **double** getGiaMua(){
26. **return** GiaMua;
27. }
28. **double** getGiaBan(){
29. **return** GiaBan;
30. }
31. **void** Nhap();
32. **void** Xuat();
33. };
35. **int** MatHang::MaHangID = 1000;
37. **void** MatHang::Nhap(){
38. cout << endl << "Nhap ten hang: ";
39. getline(cin, TenHang);
40. cout << "Nhap nhom hang: ";
41. getline(cin, NhomHang);
42. cout << "Nhap gia mua: ";
43. cin >> GiaMua;
44. cout << "Nhap gia ban: ";
45. cin >> GiaBan;
46. cin.ignore();
47. }
49. **void** MatHang::Xuat(){
50. cout << left << setw(15) << MaHang << setw(30) << TenHang << setw(30) << NhomHang << setw(30) << GiaMua << setw(10) << GiaBan << endl;
51. }

Class DanhSachBanHang

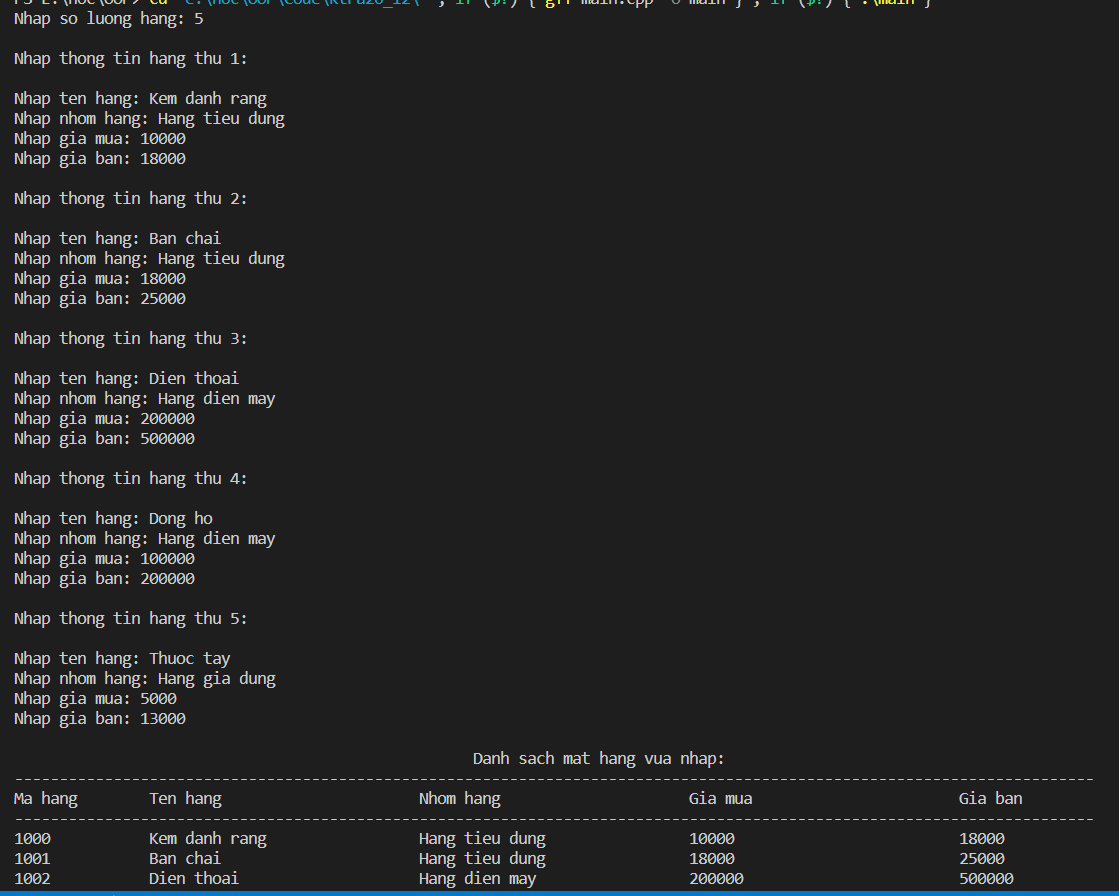
1. #include <iostream>
2. #include <string>
3. #include <iomanip>
4. #include <utility>
5. #include "MatHang.cpp"
6. #include "NhanVienBanHang.cpp"
8. **using** **namespace** std;
10. **class** DanhSachBanHang{
11. **private**:
12. // su dung pair de luu cap NhanVienBanHang va MatHang
13. pair<NhanVienBanHang\*, MatHang\*> NhanVienPairMatHang;
14. //
15. **int** SoLuongBanRa;
16. **public**:
17. DanhSachBanHang(){};
18. DanhSachBanHang(NhanVienBanHang\* nhanvien, MatHang\* mathang, **int** SoLuongBanRa): NhanVienPairMatHang(make\_pair(nhanvien, mathang)), SoLuongBanRa(SoLuongBanRa){};
19. string getNhanVien(){
20. **return** NhanVienPairMatHang.first -> getHoTen();
21. }
22. string getMatHang(){
23. **return** NhanVienPairMatHang.second -> getTenHang();
24. }
25. **int** getSoluong(){
26. **return** SoLuongBanRa;
27. }
28. **double** getLai(){
29. **return** (NhanVienPairMatHang.second -> getGiaBan() - NhanVienPairMatHang.second -> getGiaMua());
30. }
31. **void** Xuat();
32. };
34. **void** DanhSachBanHang::Xuat(){
35. cout << left << setw(30) << NhanVienPairMatHang.first ->getHoTen() << setw(30) << NhanVienPairMatHang.second -> getTenHang() << setw(10) << SoLuongBanRa << endl;
36. }

Main – Xử lý yêu cầu đề bài

1. #include <iostream>
2. #include <string>
3. #include <iomanip>
4. #include <vector>
5. #include <utility>
6. #include <algorithm>
7. #include <math.h>
8. #include "DanhSachBanHang.cpp"
10. **using** **namespace** std;
12. **int** main(){
13. // Khai bao cac vector luu danh sach nhan vien, mat hang, danh sach ban hang
14. vector<NhanVienBanHang\*> ListnhanVien;
15. vector<MatHang\*> ListmatHang;
16. vector<DanhSachBanHang\*> ListdanhSachBanHang;
18. //1. Nhap danh sach hang + in
19. //Nhap
20. **int** soLuongHang;
21. cout << "Nhap so luong hang: ";
22. cin >> soLuongHang;
23. cin.ignore();
24. **for**(**int** i = 0; i < soLuongHang; i++){
25. cout << endl << "Nhap thong tin hang thu " << i + 1 << ": " << endl;
26. MatHang\* temp = **new** MatHang();
27. temp -> Nhap();
28. ListmatHang.push\_back(temp);
29. }
30. cout << endl;
31. //Xuat
32. cout << right << setw(80) << "Danh sach mat hang vua nhap: " << endl;
33. cout << left << setfill('-') << setw(120) << "-" << setfill(' ') << endl;
34. cout << left << setw(15) << "Ma hang" << setw(30) << "Ten hang" << setw(30) << "Nhom hang" << setw(30) << "Gia mua" << setw(10) << "Gia ban" << endl;
35. cout << left << setfill('-') << setw(120) << "-" << setfill(' ') << endl;
36. **for**(**int** i = 0; i < ListmatHang.size(); i++){
37. ListmatHang[i] -> Xuat();
38. }
40. //2. Nhap danh sach nhan vien + in
41. //Nhap
42. **int** soLuongNhanVien;
43. cout << endl << "Nhap so luong nhan vien: ";
44. cin >> soLuongNhanVien;
45. cin.ignore();
46. **for**(**int** i = 0; i < soLuongNhanVien; i++){
47. cout << endl << "Nhap thong tin nhan vien thu " << i + 1 << ": " << endl;
48. NhanVienBanHang\* temp = **new** NhanVienBanHang();
49. temp -> input();
50. ListnhanVien.push\_back(temp);
51. }
52. cout << endl;
53. //Xuat
54. cout << right << setw(80) << "Danh sach nhan vien vua nhap: " << endl;
55. cout << left << setfill('-') << setw(120) << "-" << setfill(' ') << endl;
56. cout << left << setw(15) << "Ma NV" << setw(30) << "Ho ten" << setw(30) << "Dia chi" << setw(30) << "So dien thoai" << setw(10) << "Ngay HD" << endl;
57. cout << left << setfill('-') << setw(120) << "-" << setfill(' ') << endl;
58. **for**(**int** i = 0; i < ListnhanVien.size(); i++){
59. ListnhanVien[i] -> output();
60. }
62. //3.1 Lap bang ban hang
63. cout << endl << right << setw(80) << "Nhap thong tin ban hang(Nhap x hoac de trong de ket thuc): " << endl;
64. **for**(**int** i=0; i < ListnhanVien.size(); i++){
65. **bool** check=**false**;
66. **do**{
67. cout << endl << "Nhap thong tin ban hang cua nhan vien " << ListnhanVien[i]->getHoTen() << ": " << endl;
68. cout << "Nhap ten hang ban ra: ";
69. string tenHang;
70. getline(cin, tenHang);
71. **if**(tenHang == "x" || tenHang == "X" || tenHang == ""){
72. check = **true**;
73. cout << "Da ket thuc nhap thong tin ban hang cua nhan vien " << ListnhanVien[i]->getHoTen() << "! " << endl;
74. **break**;
75. }
76. cout << "Nhap so luong ban ra: ";
77. **int** soLuong;
78. cin >> soLuong;
79. cin.ignore();
80. //su dung pair de luu cap gia tri ten nhan vien va so luong
81. pair<string, **int**> Thongtin = make\_pair(tenHang, soLuong);
82. **for**(**int** j=0; j < ListmatHang.size(); j++){
83. **if**(ListmatHang[j]->getTenHang() == Thongtin.first){
84. DanhSachBanHang\* temp = **new** DanhSachBanHang(ListnhanVien[i], ListmatHang[j], Thongtin.second);
85. ListdanhSachBanHang.push\_back(temp);
86. **break**;
87. }
88. **if**(j == ListmatHang.size() - 1){
89. cout << "Khong tim thay ten hang trong danh sach hang hoa! Vui long Nhap lai" << endl;
90. }
91. }
92. }**while**(check==**false**);
93. }
95. //3.2 In bang ban hang
96. cout << endl << endl << right << setw(50) << "Bang ban hang: " << endl;
97. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
98. cout << left << setw(30) << "Nhan vien" << setw(30) << "Mat hang" << setw(10) << "So luong" << endl;
99. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
100. **for**(**int** i=0; i < ListdanhSachBanHang.size(); i++){
101. ListdanhSachBanHang[i]->Xuat();
102. }
104. //4 Sap xep
106. //4.1 Sap xep theo ten nhan vien
107. **for**(**int** i=0; i < ListdanhSachBanHang.size() - 1; i++){
108. **for**(**int** j=i+1; j < ListdanhSachBanHang.size(); j++){
109. **if**(ListdanhSachBanHang[i]->getNhanVien() > ListdanhSachBanHang[j]->getNhanVien()){
110. swap(ListdanhSachBanHang[i], ListdanhSachBanHang[j]);
111. }
112. }
113. }
114. cout << endl << endl << right << setw(60) << "Bang ban hang sap xep theo ten nhan vien: " << endl;
115. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
116. cout << left << setw(30) << "Nhan vien" << setw(30) << "Mat hang" << setw(10) << "So luong" << endl;
117. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
118. **for**(**int** i=0; i < ListdanhSachBanHang.size(); i++){
119. ListdanhSachBanHang[i]->Xuat();
120. }
122. //4.2 Sap xep theo ten mat hang
123. **for**(**int** i=0; i < ListdanhSachBanHang.size() - 1; i++){
124. **for**(**int** j=i+1; j < ListdanhSachBanHang.size(); j++){
125. **if**(ListdanhSachBanHang[i]->getMatHang() > ListdanhSachBanHang[j]->getMatHang()){
126. swap(ListdanhSachBanHang[i], ListdanhSachBanHang[j]);
127. }
128. }
129. }
130. cout << endl << endl << right << setw(60) << "Bang ban hang sap xep theo ten mat hang: " << endl;
131. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
132. cout << left << setw(30) << "Nhan vien" << setw(30) << "Mat hang" << setw(10) << "So luong" << endl;
133. cout << left << setfill('-') << setw(80) << "-" << setfill(' ') << endl;
134. **for**(**int** i=0; i < ListdanhSachBanHang.size(); i++){
135. ListdanhSachBanHang[i]->Xuat();
136. }
137. cout << endl;
139. //5. lap bang tinh cong cho nhan vien
140. cout << endl << endl << right << setw(50) << "Bang tinh cong cho nhan vien: " << endl;
141. cout << left << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
142. cout << left << setw(30) << "Nhan vien" << setw(30) << "Cong" << endl;
143. cout << left << setfill('-') << setw(60) << "-" << setfill(' ') << endl;
144. **for**(**int** i=0; i < ListnhanVien.size(); i++){
145. **double** Cong = 0;
146. **for**(**int** j=0; j < ListdanhSachBanHang.size(); j++){
147. **if**(ListdanhSachBanHang[j]->getNhanVien() == ListnhanVien[i]->getHoTen()){
148. Cong += ListdanhSachBanHang[j]->getSoluong() \* ListdanhSachBanHang[j]->getLai();
149. }
150. }
151. cout << left << setw(30) << ListnhanVien[i]->getHoTen() << setw(30) << fixed << setprecision(2) << Cong\*2/100 << endl;
152. }
153. }

Ket Qua

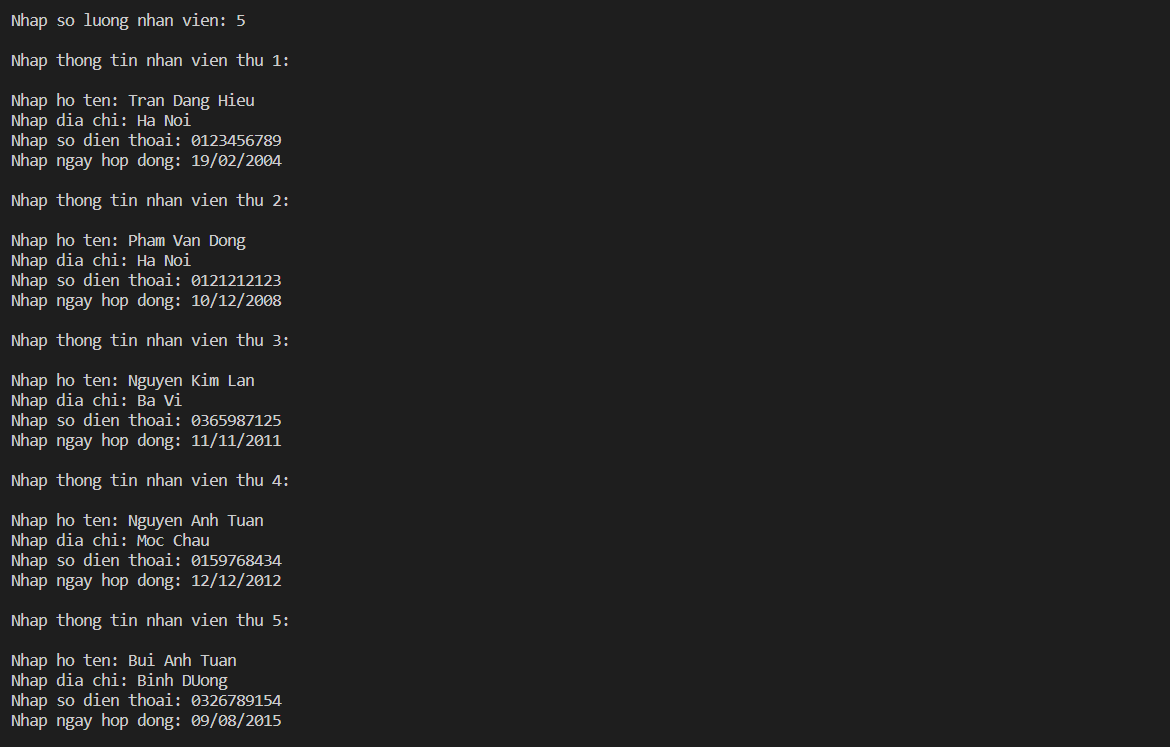
Bài 1: Nhập danh sách mặt hàng mới. In ra danh sách các mặt hàng đã có



A screen shot of a computer

Description automatically generated

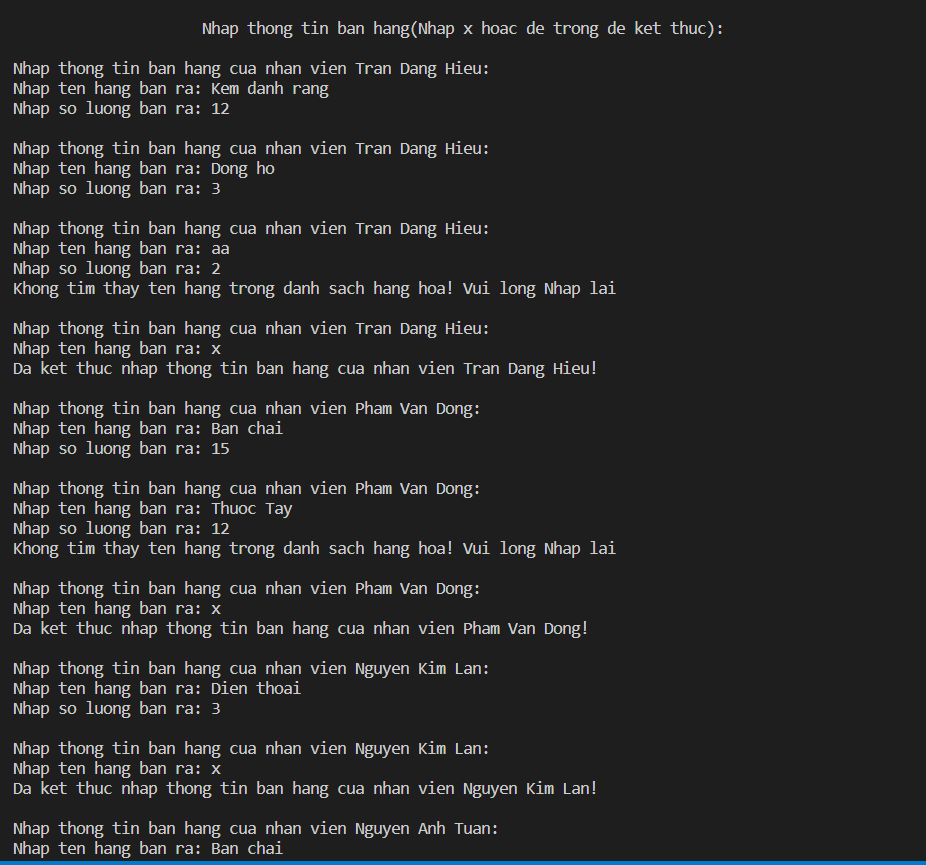
Bài 2: Nhập danh sách nhân viên bán hàng.  In ra danh sách nhân viên đã có



A screen shot of a computer

Description automatically generated

Bài 3: Lập Bảng danh sách bán hàng cho từng nhân viên và in danh sách ra màn hình



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

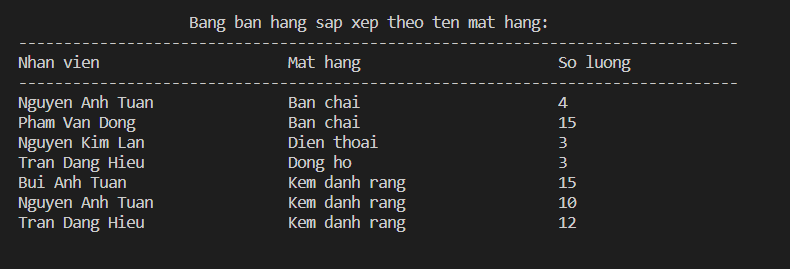
Bài 4: Sắp xếp danh sách Bảng danh sách bán hàng   

* 1. Theo tên nhân viên

A screenshot of a computer

Description automatically generated

b. Theo tên mặt hàng



Bài 5: Lập bảng tính công cho mỗi nhân viên. Giá trị tiền công được tính là 2% tổng số lợi nhuận thu được.

A screenshot of a computer

Description automatically generated