Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Петрозаводский государственный университет»

Физико-технический институт

Кафедра информационно-измерительных систем и физической электроники

ПЕКАРНЯ ИНДИВИДУАЛЬНЫЙ ОТЧЁТ

Командный проект по дисциплине «Технология программирования»

Выполнил:

студент группы 21312

А. А. Запорожец

Научный руководитель:

канд. физ.-мат. наук, доцент КИИСиФЭ А. В. Бульба

Петрозаводск 2023

1. **Удалил .pro.user** (ЗАА) - Удалил Baker-street.pro.user, тк вызвал конфликт.
2. **Добавил Profit.h, ProfitInputScreen.h, ProfitList.h** (ЗАА) - В заголовочные файлы были добавлены прототипы класса Profit, ProfitInputScreen и ProfitList.

**Profit.h**

#ifndef PROFIT

#define PROFIT

#include "libs.h"

class Profit

{

private:

string name;

int quantity;

double globalcost;

string date;

public:

Profit(string, int, double,string);

~Profit();

string GetName();

int GetQuantity();

double GetGlobalcost();

string GetDate();

};

#endif // PROFIT

**ProfitInputScreen.h  
  
#ifndef PROFITINPUTSCREEN**

**#define PROFITINPUTSCREEN**

**#include "ProfitList.h"**

**class ProfitInputScreen**

**{**

**private:**

**ProfitList\* ptrProfitList;**

**string ProfitName;**

**int ProfitQuantity;**

**double ProfitGlobalcost;**

**string ProfitDate;**

**public:**

**ProfitInputScreen(ProfitList\* ptrPrfL) : ptrProfitList(ptrPrfL)**

**{**

**/\* тут пусто \*/**

**}**

**void setProfit(); // добавить данные о доходе**

**};**

**#endif // PROFITINPUTSCREEN**

**ProfitList.h**

**#ifndef PROFITLIST**

**#define PROFITLIST**

**#include "Profit.h"**

**class ProfitList {**

**private:**

**// установить указатели на доход**

**list <Profit\*> setPtrsProfits; // указатели на класс дохода**

**list <Profit\*>::iterator iter; //итератор**

**public:**

**~ProfitList(); // деструктор (удаление дохода из списка)**

**void insertProfit(Profit\*); // добавить доход в список**

**int GetQuantityProfit(string);**

**double GetGlobalcostProfit(string);**

**string GetDateProfit(string);**

**void DisplaySummProfits(); // вывод списка доходов**

**void DisplayProf();**

**};**

**#endif // PROFITLIST**

1. **Мелкие правки в созданные файлы** (ЗАА) -Проведены мелкие изменения в уже созданных файлах
2. **Merge branch 'develop-Artem' of https://github.com/GigaBuddy0x0/Baker-street into develop-Artem** (ЗАА) -Залил данные с ветки 'develop-Artem' из удалённого репозитория в ветку 'develop-Artem' своего локального
3. **Merge branch 'develop-Artem' into develop-general** (ЗАА) - Залил все наработки со своей ветки в основную ветку разработки.
4. **Merge branch 'develop-general' into develop-Artem** (ЗАА) – Залил все наработки с 'develop-general' ветки в свою ветку разработки.
5. **Добавлена реализация класса Profit** (ЗАА) – Была написана реализация класса Profit  
     
   **Profit.cpp**

**#include "Profit.h"**

**Profit::~Profit() // деструктор**

**{ }**

**Profit::Profit(string n, int q, double g, string d): name(n), quantity(q), globalcost(g), date(d)**

**{ }**

**string Profit::GetName()**

**{**

**return name;**

**}**

**int Profit::GetQuantity()**

**{**

**return quantity;**

**}**

**double Profit::GetGlobalcost()**

**{**

**return globalcost;**

**}**

**string Profit::GetDate()**

**{**

**return date;**

**}**

1. **Добавлена реализация класса ProfitList** (ЗАА) – Была написана реализация класса ProfitList  
     
   **ProfitList.cpp**

#include "ProfitList.h"

ProfitList::~ProfitList() // деструктор

{ // удалить строки с доходами,

// удалить указатели из списка.

while (!setPtrsProfits.empty())

{

iter = setPtrsProfits.begin();

delete \*iter;

setPtrsProfits.erase(iter);

}

}

void ProfitList::InsertProfit(Profit\* ptrProf)

{

setPtrsProfits.push\_back(ptrProf);

}

void ProfitList::DisplayProf() // отобразить все строки с доходами

{

cout << endl;

cout << left << setw(15) << "Name" << setw(15) << "Quantity"

<< setw(15) << "Globalcost" << setw(15) << "Date";

cout << endl << "-----------------------------------------------------" << endl;

if (setPtrsProfits.empty()) // если список доходов пуст

cout << "\*\*\*There's nothing here\*\*\*\n" << endl; // выводим запись, что он пуст)

else

{

iter = setPtrsProfits.begin();

while (iter != setPtrsProfits.end()) // распечатываем все доходы

{

cout << left << setw(15) << (\*iter)->GetName()

<< setw(15) << (\*iter)->GetQuantity()

<< setw(15) << (\*iter)->GetGlobalcost()

<< setw(15) << (\*iter)->GetDate() << endl;

\*iter++;

}

}

}

float ProfitList::DisplaySummProfits() // сумма всех доходов

{

float sumProfits = 0.0;

iter = setPtrsProfits.begin();

while (iter != setPtrsProfits.end())

{ // плюсуем суммы всех доходов за все время

sumProfits += (\*iter)->GetGlobalcost();

iter++;

}

return sumProfits;

}

1. **Добавлена реализация класса ProfitInputScreen, так же подправлены комментарии в ProfitList** (ЗАА) – Была написана реализация класса ProfitInputScreen, так же исправил комментарии в классе ProfitList

**ProfitInputScreen.cpp**

**#include "ProfitInputScreen.h"**

**void ProfitInputScreen::setProfit() // добавить данные о доходе**

**{**

**cout << "Input the name of profit: " << endl;**

**getaLine(ProfitName);**

**cout << "Input the quantity of profit: " << endl;**

**cin >> ProfitQuantity;**

**cin.ignore(80, '\n');**

**cout << "Input the cost of profit: " << endl;**

**cin >> ProfitGlobalCost;**

**cin.ignore(80, '\n');**

**cout << "Input the date of profit: " << endl;**

**cin >> ProfitDate;**

**cin.ignore(80, '\n');**

**Profit\* ptrProfit = new Profit(ProfitName, ProfitQuantity, ProfitGlobalCost, ProfitDate); // создать доход**

**ptrProfitList->InsertProfit(ptrProfit); // занести в список доходов**

**}**

1. **Merge branch 'develop-Artem' into develop-general** (ЗАА) – Залил все наработки с ветки 'develop-Artem'в основную ветку разработки.