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. Л. Рокотянский

Научный руководитель:

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

Петрозаводск 2023

**1.Update UserInterface.h and AnnualReport.h** (РАЛ)  – Обновлены файлы UserInterface.h и AnnualReport.h  
  
31. **Merge branch 'develop-general' into develop-Sasha** (РАЛ) – Залил все наработки с 'develop-general' ветки в свою  ветку разработки.  
  
**32. Добавил реализацию класса Produce** (РАЛ)  – Была написана реализация класса Produce  
  
**33. Добавил реализацию класа ProduceList** (РАЛ) – Была написана реализация класса ProduceList  
  
34. **Добавил реализацию ProduceInputScreen,изменил вид вывода продукции на экран** (РАЛ)  – Была написана реализация класса ProduceInputScreen, так же внесены изменения в вывод продукции на экран.  
  
46. **Подправлен вывод таблицы расходов** (РАЛ) – Исправлен вывод данных в таблице расходов.  
  
**47. Merge branch 'develop-Sasha' into develop-general** (РАЛ)  – Залил все наработки с ветки 'develop-Sasha'в основную ветку разработки.  
  
**48. Визуальные изменения** (РАЛ)  – Были внесены небольшие визуальные изменения.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Baker-street/UserInterface.h  
  
#ifndef USERINTERFACE

#define USERINTERFACE

//////////////////Класс UserInterface//////////////////////

//Главный класс для запуска приложения:

//этот класс определяет взаимодействие юзера с программой.

#include "ProduceInputScreen.h"

#include "ProfitInputScreen.h"

#include "ExpenseInputScreen.h"

#include "AnnualReport.h"

class UserInterface

{

private:

ProduceList\* ptrProduceList;

ProduceInputScreen\* ptrProduceInputScreen;

ProfitList\* ptrProfitList;

ProfitInputScreen\* ptrProfitInputScreen;

ExpenseList\* ptrExpenseList;

ExpenseInputScreen\* ptrExpenseInputScreen;

AnnualReport\* ptrAnnualReport;

public:

UserInterface();

~UserInterface();

void interact();

};

#endif // USERINTERFACE

Baker-street/AnnualReport.h  
  
#ifndef ANNUALREPORT

#define ANNUALREPORT

//////////////////класс AnnualReport///////////////////////

//Класс годового отчета

#include "ProfitList.h"

#include "ExpenseList.h"

class AnnualReport

{

private:

ProfitList\* ptrPL; // записи доходов

ExpenseList\* ptrEL; // записи расходов

float expenses, profits; // суммы доходов и расходов

public:

AnnualReport(ProfitList\* pPL, ExpenseList\* pEL): ptrPL(pPL), ptrEL(pEL)

{ }

void display(); // отображение годового отчета

};

#endif // ANNUALREPORT  
  
  
  
  
  
  
  
  
  
  
  
Baker-street/ProduceList.cpp  
  
#include "ProduceList.h"

ProduceList::~ProduceList() // деструктор

{ // удалить строки с продукцией,

// удалить указатели из списка.

while (!setPtrsProds.empty())

{

iter = setPtrsProds.begin();

delete \*iter;

setPtrsProds.erase(iter);

}

}

void ProduceList::InsertProduce(Produce\* ptrPr) // добавить продукцию в список

{

setPtrsProds.push\_back(ptrPr); // вставка новой продукции в список

}

//void ProduceList::DeletProduce(Produce\* ptrPr) // удалить продукцию из списка

double ProduceList::GetCostProduce(string name) // получить стоимость товара по его имени

{

int cost;

iter = setPtrsProds.begin();

while (iter != setPtrsProds.end())

{

cost = (\*iter)->GetCost();

if (name == ((\*iter)->GetName())) // сравниваем по именам и

{ // если получившаяся пара совпадает - значит,

//мы нашли запись об этом товаре в списке, в этом случае

return cost; // возвращаем его стоимость

}

iter++;

}

return -1; // если нет - возвращаем -1

}

void ProduceList::DisplayProd() // вывод списка товаров

{

cout << endl;

cout << left << setw(15) << "Id" << setw(15) << "Name" << setw(15) << "Cost";

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

if (setPtrsProds.empty()) // если список товаров пуст

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

else

{

iter = setPtrsProds.begin();

while (iter != setPtrsProds.end()) // распечатываем все товары

{

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

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

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

\*iter++;

}

}

}  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Baker-street/Produce.cpp  
  
#include "Produce.h"

Produce::Produce(string s, int i, double d): name(s), id(i), cost(d)

{ }

Produce::~Produce() // деструктор

{ }

string Produce::GetName()

{

return name;

}

int Produce::GetId()

{

return id;

}

double Produce::GetCost()

{

return cost;

}  
  
  
  
  
  
  
  
Baker-street/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

}