ФГБОУ ВО Хакасский государственный университет им. Н.Ф. Катанова

Инженерно-технологический институт

Кафедра программного обеспечения вычислительной техники и

автоматизированных систем

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

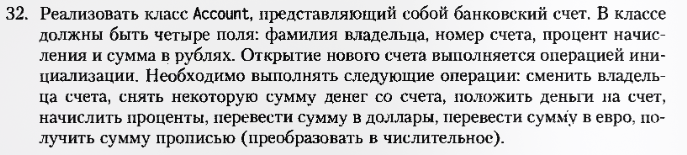
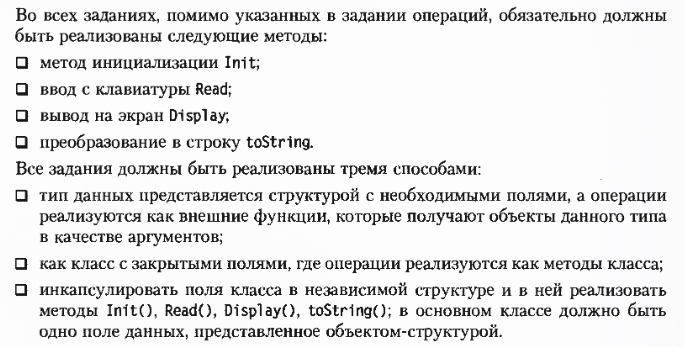
Проверил

доцент кафедры ПОВТиАС Санников Е.В.

Выполнил

студент группы 120-1 Хвостов Д.А.

Абакан 2021



Способ 1

#include <iostream>

#include <cmath>

using namespace std;

struct AccountData{

string last\_name;

string UID;

float percent;

float sum;

};

void Init(AccountData &Acc, string \_last\_name,

string \_UID, float \_percent, float \_sum){

Acc.last\_name = \_last\_name;

Acc.UID = \_UID;

Acc.percent = \_percent;

Acc.sum = \_sum;

}

bool ChangeMaster(AccountData &Acc, string \_last\_name){

if (\_last\_name == ""){

return false;

}

Acc.last\_name = \_last\_name;

return true;

}

bool GetSum(AccountData &Acc, float \_sum){

if (\_sum <= 0 || \_sum > Acc.sum) {

return false;

}

Acc.sum -= \_sum;

return true;

}

bool AddSum(AccountData &Acc, float \_sum){

if (\_sum <= 0){

return false;

}

Acc.sum += \_sum;

return true;

}

void PlusPetcent(AccountData &Acc){

Acc.sum += Acc.sum \* Acc.percent;

}

float ShowInEuro(AccountData &Acc, float Euro){

return Acc.sum / Euro;

}

float ShowInDollar(AccountData &Acc, float Dollar){

return Acc.sum / Dollar;

}

int ShowSumInNumeral(AccountData &Acc){

return Acc.sum;

}

}

void Read(AccountData &Acc){

cout << "Input last\_name: ";

cin >> Acc.last\_name;

cout << "Input UID: ";

cin >> Acc.UID;

cout << "Input percent: ";

cin >> Acc.percent;

cout << "Input sum: ";

cin >> Acc.sum;

}

void Display(AccountData &Acc)

{

cout << "last\_name: " << Acc.last\_name

<< " |UID: " << Acc.UID << " |percent: "

<< Acc.percent << " |sum: " << Acc.sum << endl;

}

string toString()

{

return "<Class Account{last\_name, UID, percent, sum}>";

}

int main()

{

AccountData account;

Read(account);

Display(account);

cout << toString() << endl;

cout << ShowInEuro(account, 65) << endl;

cout << ShowInDollar(account, 70) << endl;

cout << ShowSumInNumeral(account) << endl;

AddSum(account, 90);

Display(account);

GetSum(account, 80);

Display(account);

ChangeMaster(account, "cat");

Display(account);

PlusPetcent(account);

Display(account);

return 0;

}

Способ 2

#include <iostream>

#include <cmath>

using namespace std;

class Account{

private:

string last\_name;

string UID;

float percent;

float sum;

public:

void Init(string \_last\_name, string \_UID, float \_percent, float \_sum) {

last\_name = \_last\_name;

UID = \_UID;

percent = \_percent;

sum = \_sum;

}

bool ChangeMaster(string \_last\_name){

if (\_last\_name == "") {

return false;

}

last\_name = \_last\_name;

return true;

}

bool GetSum(float \_sum) {

if (\_sum <= 0 || \_sum > sum){

return false;

}

sum -= \_sum;

return true;

}

bool AddSum(float \_sum){

if (\_sum <= 0){

return false;

}

sum += \_sum;

return true;

}

void PlusPetcent(){

sum += sum \* percent;

}

float ShowInEuro(float Euro){

return sum / Euro;

}

float ShowInDollar(float Dollar) {

return sum / Dollar;

}

int ShowSumInNumeral() {

return sum;

}

void Read()

{

cout << "Input last\_name: ";

cin >> last\_name;

cout << "Input UID: ";

cin >> UID;

cout << "Input percent: ";

cin >> percent;

cout << "Input sum: ";

cin >> sum;

}

void Display()

{

cout << "last\_name: " << last\_name << " |UID: " << UID << " |percent: " << percent << " |sum: " << sum << endl;

}

string toString()

{

return "<Class Account{last\_name, UID, percent, sum}>";

}

};

int main()

{

Account account;

account.Read();

account.Display();

cout << account.toString() << endl;

cout << account.ShowInEuro(65) << endl;

cout << account.ShowInDollar(70) << endl;

cout << account.ShowSumInNumeral() << endl;

account.AddSum(90);

account.Display();

account.GetSum(80);

account.Display();

account.ChangeMaster("cat");

account.Display();

account.PlusPetcent();

account.Display();

return 0;

}

Способ 3

#include <iostream>

#include <cmath>

using namespace std;

class MetaData{

public:

string last\_name;

string UID;

float percent;

float sum;

void Init(string \_last\_name, string \_UID, float \_percent, float \_sum){

last\_name = \_last\_name;

UID = \_UID;

percent = \_percent;

sum = \_sum;

}

void Read() {

cout << "Input last\_name: ";

cin >> last\_name;

cout << "Input UID: ";

cin >> UID;

cout << "Input percent: ";

cin >> percent;

cout << "Input sum: ";

cin >> sum;

}

void Display(){

cout << "last\_name: " << last\_name <<

" |UID: " << UID << " |percent: " << percent <<

" |sum: " << sum << endl;

}

string toString(){

return "<Class MetaData{last\_name, UID, percent, sum}>";

}

};

class Account{

public:

MetaData meta;

Account(){

meta.Read();

}

Account(string \_last\_name, string \_UID, float \_percent, float \_sum){

meta.Init(\_last\_name, \_UID, \_percent, \_sum);

}

bool ChangeMaster(string \_last\_name){

if (\_last\_name == ""){

return false;

}

meta.last\_name = \_last\_name;

return true;

}

bool GetSum(float \_sum){

if (\_sum <= 0 || \_sum > meta.sum){

return false;

}

meta.sum -= \_sum;

return true;

}

bool AddSum(float \_sum){

if (\_sum <= 0){

return false;

}

meta.sum += \_sum;

return true;

}

void PlusPetcent(){

meta.sum += meta.sum \* meta.percent;

}

float ShowInEuro(float Euro){

return meta.sum/ Euro;

}

float ShowInDollar(float Dollar){

return meta.sum/ Dollar;

}

int ShowSumInNumeral(){

return meta.sum;

}

};

int main(){

Account account;

account.meta.Display();

cout << account.meta.toString() << endl;

cout << account.ShowInEuro(65) << endl;

cout << account.ShowInDollar(70) << endl;

cout << account.ShowSumInNumeral() << endl;

account.AddSum(90);

account.meta.Display();

account.GetSum(80);

account.meta.Display();

account.ChangeMaster("cat");

account.meta.Display();

account.PlusPetcent();

account.meta.Display();

return 0;

}

