**Malek Smida**

**Groupe D**

**Compte Rendu**

Gestion de Banque

Email: msmida96@gmail.com

**Table de contenu:**

Code …….…………………………………………………….……...2

compte.h ……………………………………………….…….3

compte.cpp …………………………………………….……3

compteCourant.h …………………………………………4

compteCourant.cpp ……………………………………..5

compteEpargne.h ……..….……………………………..6

compteEpargne.cpp …………………….………………6

banque.h ………………………..……………………………8

banque.cpp ………..…………..……………………………8

Main.cpp ………………………………………………….…10

Exemple d’execution ..………………………………………..13

**Code :**

**<compte.h>**

#pragma once

class compte

{

protected:

long num;

double sold;

public:

virtual void saisir();

virtual void ajouter(double);

virtual void retirer(double);

virtual void afficher();

compte(long=0,double=0);

virtual ~compte(void);

virtual long getNum(){return num;}

virtual double getSold(){return sold;}

};

**<compte.cpp>**

#include "compte.h"

#include<iostream>

using namespace std;

compte::compte(long n,double s):num(n),sold(s){}

void compte::saisir()

{

long x;double y;

cout<<"donnez le num compte: ";cin>>x;

cout<<"donnez le solde: ";cin>>y;

num=x;

sold=y;

}

compte::~compte(void)

{

}

void compte::ajouter(double x)

{

sold+=x;

}

void compte::retirer(double x)

{

sold-=x;

}

void compte::afficher()

{

cout<<"le numero de compte est: "<<num<<endl;

cout<<"le solde est: "<<sold<<endl;

}

**<compteCourant.h>**

#pragma once

#include "compte.h"

class compteCourant :

public compte

{

public:

void saisir();

void ajouter(double);

void afficher();

void retirer(double);

compteCourant(long=0,double=0);

~compteCourant(void);

long getNum();

double getSold();

};

**<compteCourant.cpp>**

#include "compteCourant.h"

#include<iostream>

using namespace std;

compteCourant::compteCourant(long n,double s):compte(n,s)

{

}

void compteCourant::saisir()

{

while(1)

{

compte::saisir();

if(sold>=(-500))

break;

else

cout<<"solde invalide "<<endl;

}

}

void compteCourant::ajouter(double x)

{

compte::ajouter(x);

}

compteCourant::~compteCourant(void)

{

}

void compteCourant::afficher()

{

cout<<"\*\*\*\*\*\*\*\*\*\*\* affichage de compte Courant \*\*\*\*\*\*\*\*\*\*"<<endl;

compte::afficher();

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

void compteCourant::retirer(double r)

{

if((sold-r)>=(-500))

compte::retirer(r);

else

cout<<"solde insuffisant !"<<endl;

}

long compteCourant::getNum()

{

return compte::getNum();

}

double compteCourant::getSold()

{

return compte::getSold();

}

**<compteEpargne.h>**

#pragma once

#include "compte.h"

class compteEpargne :

public compte

{

float tia;

public:

void saisir();

void ajouter(double);

void afficher();

void retirer(double);

double interetAnnuel();

compteEpargne(long=0,double=0,float=0);

~compteEpargne(void);

long getNum(){return num;}

double getSold(){return sold;}

};

**<compteEpargne.cpp>**

#include "compteEpargne.h"

#include<iostream>

using namespace std;

compteEpargne::compteEpargne(long n,double s,float t):compte(n,s),tia(t)

{

}

void compteEpargne::saisir()

{

while(1)

{

compte::saisir();

if(sold>=5)

break;

else

cout<<"solde invalide "<<endl;

}

cout<<"donnez taux interet annuel: ";cin>>tia;

}

void compteEpargne::ajouter(double x)

{

compte::ajouter(x);

}

compteEpargne::~compteEpargne(void)

{

}

void compteEpargne::afficher()

{

cout<<"\*\*\*\*\*\*\*\*\*\*\* affichage de compte Epargne \*\*\*\*\*\*\*\*\*\*"<<endl;

compte::afficher();

cout<<"taux interet annuel: "<<tia<<endl<<"interet annuel: "<<interetAnnuel()<<endl;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

}

void compteEpargne::retirer(double r)

{

if((sold-r)>=5)

compte::retirer(r);

else

cout<<"solde insuffisant !"<<endl;

}

double compteEpargne::interetAnnuel()

{

return (tia\*sold)/100;

}

**<banque.h>**

#pragma once

#include "compte.h"

#include "compteCourant.h"

#include "compteEpargne.h"

#include<vector>

using namespace std;

class banque

{

vector<compte\*> bq;

public:

void remplir();

void ajouter(long,double);

void supprimer(long);

void afficher(long);

void afficherTous();

void retirer(long);

banque(void);

~banque();

};

**<banque.cpp>**

#include "banque.h"

#include<iostream>

#include<string>

#include<vector>

using namespace std;

banque::banque(void)

{

}

banque::~banque(void)

{

for(int i=0;i<bq.size();i++)

delete (bq[i]);

bq.clear();

}

void banque::ajouter(long n,double x)

{

for(int unsigned i=0;i<bq.size();i++)

if(bq[i]->getNum()==n)

bq[i]->ajouter(x);

}

void banque::supprimer(long n)

{

for(int unsigned i=0;i<bq.size();i++)

if(bq[i]->getNum()==n)

bq.erase(bq.begin()+i);

}

void banque::retirer(long n)

{

double r;

cout<<"donnez le sold a retire: ";cin>>r;

for(int unsigned i=0;i<bq.size();i++)

if(bq[i]->getNum()==n)

bq[i]->retirer(r);

}

void banque::afficher(long n)

{

for(int unsigned i=0;i<bq.size();i++)

if(bq[i]->getNum()==n)

bq[i]->afficher();

}

void banque::afficherTous()

{

for(int unsigned i=0;i<bq.size();i++)

bq[i]->afficher();

}

void banque::remplir()

{

string c;

do

{

cout<<"compte epargne [e] ou compte courant [c]: ";cin>>c;

}while((c!="e")&&(c!="c"));

if(c=="e")

{

compte\* q=new compteEpargne;

q->saisir();

bq.push\_back(q);

}

else

{

if(c=="c")

{

compte\* q=new compteCourant;

q->saisir();

bq.push\_back(q);

}

else

cout<<"caractere invalide: "<<endl;

}

}

**<main.cpp>**

#include<iostream>

#include<vector>

#include<string>

#include"compte.h"

#include "banque.h"

#include "compteCourant.h"

#include "compteEpargne.h"

using namespace std;

void main()

{

cout<<"\t\t\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

cout<<"\t\t\t| |"<<endl;

cout<<"\t\t\t| Gestion de Banque |"<<endl;

cout<<"\t\t\t|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|"<<endl;

int x;

banque b;

while(1)

{

cout<<"\t\t\t ----------------MENU-----------------"<<endl;

cout<<"\t\t\t| 1 > Ajouter un compte |"<<endl;

cout<<"\t\t\t| 2 > Supprimer un compte |"<<endl;

cout<<"\t\t\t| 3 > ajouter argent au compte |"<<endl;

cout<<"\t\t\t| 4 > Retirer argent de compte |"<<endl;

cout<<"\t\t\t| 5 > rechercher un compte |"<<endl;

cout<<"\t\t\t| 6 > Afficher tous les comptes |"<<endl;

cout<<"\t\t\t -------------------------------------"<<endl;

cout<<"\n\t\t\t votre choix: ";cin>>x;

switch(x)

{

case 1:

{

b.remplir();

}break;

case 2:

{

long y;

cout<<"donnez le num de compte: ";cin>>y;

b.supprimer(y);

}break;

case 3:

{

long y;

cout<<"donnez le num de compte: ";cin>>y;

double x;

cout<<"donnez le montant a ajouter au compte: ";cin>>x;

b.ajouter(y,x);

}break;

case 4:

{

long y;

cout<<"donnez le num de compte: ";cin>>y;

b.retirer(y);

}break;

case 5:

{

long y;

cout<<"donnez le num de compte: ";cin>>y;

b.afficher(y);

}break;

case 6:

{

b.afficherTous();

}break;

default: cout<<"\n\t\t\tchoix invalide, try again! "<<endl<<endl<<endl; break;

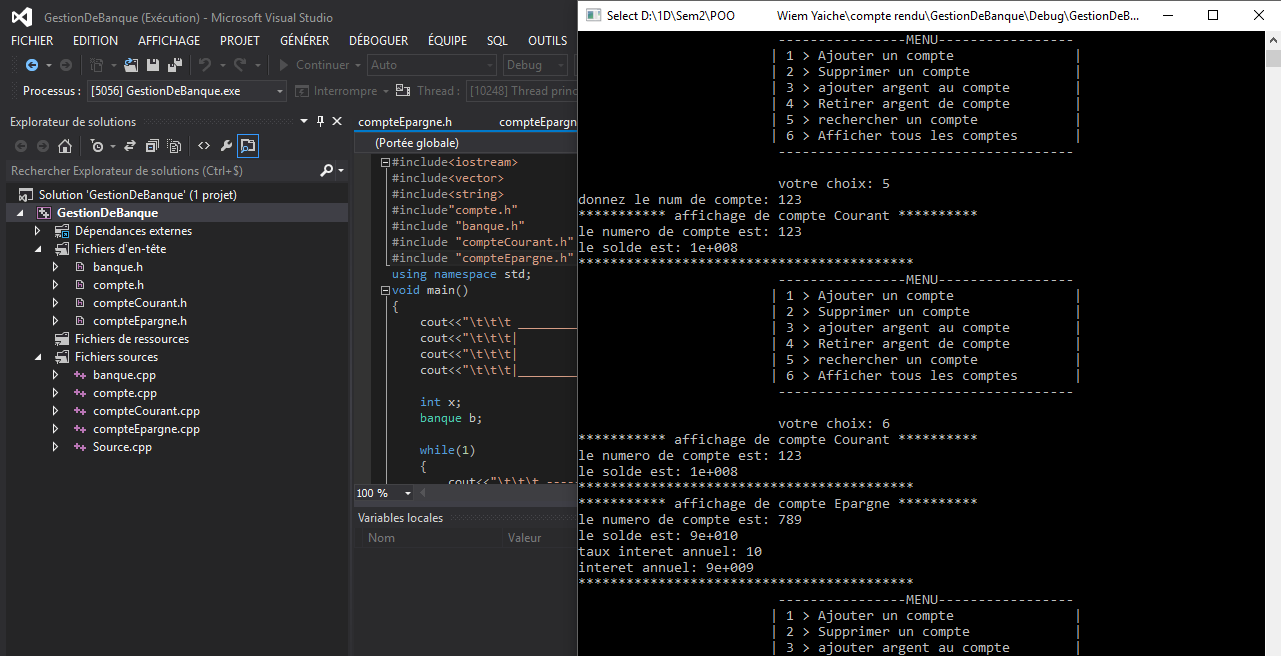
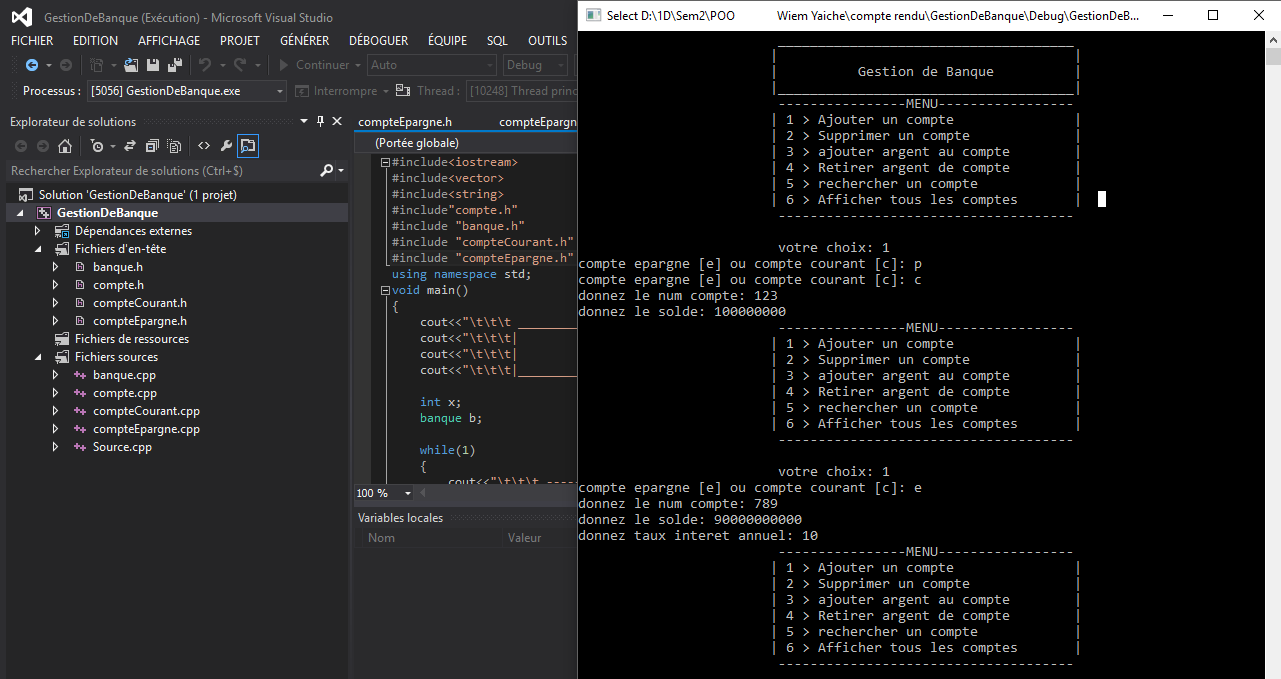
}

}

system("pause");

}

**Exemple d’exécution :**

****

