

PROJET C++:

Gestion Des Banque

Réalisé par:

BAJADDA AHMED
YASSER JERAF
OUSSAMA MAZOUZ

Encadre Par

Nom Classe

PersonnePhysique

Attributs

-nom_m: string
-prenom_m:string

Méthodes

+PersonnePhysique()
+getNom()
+getPrenom()

Hérite de

RetraitInterditException

+RetraitInterditException ()
+affiche_err ()

CompteBancaire

-solde_m: real
-code_m :string
-decouvertAutorise_m:entier
-nombreComptes_m:static int
-nombreComptesDebiteurs_m:static entier

+CompteBancaire ()
+setCode ()
+getCode ()
+getSolde ()
+depot ()
+retrait ()
+SetDecouvertAutorise()
+getDecouvertAutorise ()
+getNombreComptes ()
+getNombreComptesDebiteurs ()
+getDecouvertAutorise ()
+getNombreComptes ()

```

void retrait(double montant, string code)
{
    if (code_m != code)
    {
        cout << "Code incorrect\n" << endl;
        return;
    }

    if (montant > solde_m + decouvertAutorise_m)
    {
        throw RetraitInterditException();
        return;
    }

    solde_m -= montant;

    if (solde_m < 0 )
    {
        nombreComptesDebiteurs_m++;
    }
}

```

```

class RetraitInterditException
{
public:
    RetraitInterditException() {}
    string affiche_err()
    {
        return "Retrait impossible : solde insuffisant.";
    }
};

```

Inside notre main():

```

try
{
    CompteBancaire::List_CompteBancaires[i]->retrait(montant,code);
}
catch(RetraitInterditException er)
{
    cout<<er.affiche_err();
}

```

si le montant à retirer est supérieur au solde disponible, en prenant en compte le découvert autorisé, la méthode retrait() lance une exception de type RetraitInterditException

```

class CompteBancaire:public PersonnePhysique
{
private:
    double solde_m;
    string code_m;
    int decouvertAutorise_m;
    static int nombreComptes_m;
    static int nombreComptesDebiteurs_m;

public:
    static vector<CompteBancaire*> List CompteBancaires;
    CompteBancaire(string nom="inconnu", string prenom="inconnu"):PersonnePhysique(nom, prenom)
    {
        solde_m = 0;
        code_m = "NULL";
        decouvertAutorise_m = 0;
        nombreComptes_m++;
        List_CompteBancaires.push_back(this); // Stocker l'objet créé dans le vecteur 'List_Com
    }
}

```

La variable static vector List_CompteBancaires nous aide pour stocker tous les objets créés de la classe CompteBancaire. Elle permet de les référencer et de les manipuler ultérieurement dans notre main.

Inside main():

```
BIENVENUE AU MENU ():  
  
1 > INSCRIPTION  
2 > SOLDE  
3 > DEPOT  
4 > RETRAIT  
5 > INTERET  
6 > E X I T  
  
|-->Entrer votre choix: _
```

On a utilisé une switch case pour garantir a l'utilisateur multi choix, et un goto (Bl) pour faire un retour au menu

MERCI POUR
VOTRE
ATTENTION