|  |
| --- |
| Polymorphisme |

class Produit

{

protected string reference;

protected string designation;

protected int prixVente;

public Produit(string reference, string designation, int prixVente)

{

this.reference = reference;

this.designation = designation;

this.prixVente = prixVente;

}

public virtual void afficher()

{

Console.WriteLine("référence : " +reference);

Console.WriteLine("désignation : " +designation);

Console.WriteLine("prix de vente : " +prixVente);

}

public void proc(Produit pdt)

{

this.pdt = pdt;

pdt.afficher();

}

}

class PdtAch : Produit

{

private int numFournisseur;

private int prixAchat;

public PdtAch(string reference, string designation, int prixVente, int numFournisseur, int prixAchat) : base(reference, designation, prixVente)

{

this.numFournisseur = numFournisseur;

this.prixAchat = prixAchat;

}

public override void afficher()

{

base.afficher();

Console.WriteLine("numero fournisseur : " + numFournisseur);

Console.WriteLine("prix d'achat : " + prixAchat);

}

}

class PdtFab : Produit

{

private int numFab;

private int nbHFab;

private int coutMat;

public PdtFab(string reference, string designation, int prixVente, int numFab, int nbHFab, int coutMat) : base(reference, designation, prixVente)

{

this.numFab = numFab;

this.nbHFab = nbHFab;

this.coutMat = coutMat;

}

public override void afficher()

{

base.afficher();

Console.WriteLine("numero fabriqué : " + numFab);

Console.WriteLine("nombre d'heure nécessaire : " + nbHFab);

Console.WriteLine("cout de la matière première : " + coutMat);

}

}

class Program

{

static void Main(string[] args)

{

Produit pdt;

PdtFab pf;

PdtAch pa;

pdt = new Produit("aaa", "télé", 220);

pf = new PdtFab("f01", "réveil matin", 150, 1, 10, 50);

pa = new PdtAch("a01", "montre bracelet", 75, 1, 45);

pdt = pf;

Console.WriteLine("Affichage d'un produit fabriqué : ");

pdt.afficher();

Console.WriteLine("Affichage d'un produit acheté : ");

pdt = pa;

pdt.afficher();

Console.ReadLine();

}

}

# Avec la procédure proc

class Program

{

static void Main(string[] args)

{

Produit pdt;

PdtFab pf;

PdtAch pa;

pdt = new Produit("ref", "", 0);

pf = new PdtFab("f01", "réveil matin", 150, 1, 10, 50);

pa = new PdtAch("a01", "montre bracelet", 75, 1, 45);

Console.WriteLine("Affichage d'un produit fabriqué : ");

pdt.proc(pf);

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ");

Console.WriteLine("Affichage d'un produit acheté : ");

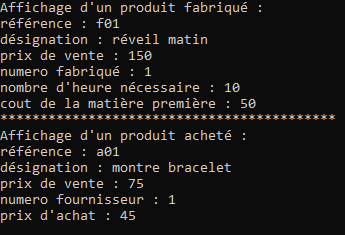
pdt.proc(pa);

Console.ReadLine();

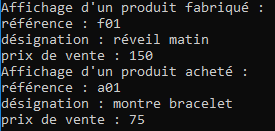
}

}

### L'affichage reste le même



Si on enlève virtual dans produit et override dans ProdFab et ProdAch, on obtient :



# Collection polymorphe

class Produit

{

protected string reference;

protected string designation;

protected int prixVente;

protected Produit pdt;

protected List<Produit> produits = new List<Produit>();

public void addListPdt(Produit pdt)

{

this.produits.Add(pdt);

}

public void afficherListPdt()

{

for (int i = 0; i < produits.Count; i++)

{

Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine(produits.ElementAt(i).GetType());

produits.ElementAt(i).afficher();

}

}

}

class Program

{

static void Main(string[] args)

{

Produit pdt;

PdtFab pf;

PdtAch pa;

pdt = new Produit("ref", "", 0);

pf = new PdtFab("f01", "réveil matin", 150, 1, 10, 50);

pa = new PdtAch("a01", "montre bracelet", 75, 1, 45);

pdt.addListPdt(pf);

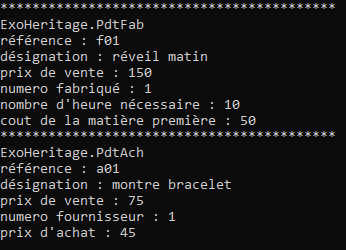
pdt.addListPdt(pa);

pdt.afficherListPdt();

Console.ReadLine();

}

}



class Produit

{

protected string reference;

protected double prixVente;

public Produit(string reference, double prixVente)

{

this.reference = reference;

this.prixVente = prixVente;

}

public virtual double cout()

{

return 0;

}

public double marge()

{

return prixVente - cout();

}

}

class PdtAch : Produit

{

private double prixAchat;

public PdtAch(double prixAchat, string reference, double prixVente) : base(reference, prixVente)

{

this.prixAchat = prixAchat;

}

public override double cout()

{

return prixAchat + 100;

}

}

class PdtFab : Produit

{

private double nbHeures;

private double matiere;

public PdtFab(double nbHeures, double matiere, string reference, double prixVente) : base(reference, prixVente)

{

this.nbHeures = nbHeures;

this.matiere = matiere;

}

public override double cout()

{

return matiere + nbHeures \* 4;

}

}

class Program

{

static void Main(string[] args)

{

Produit pdt = new Produit("p01", 20);

PdtFab pf = new PdtFab(5,2,"f01", 30);

pdt = pf;

Console.WriteLine("Marge (30 - (2 + 5\*4)) : "+pf.marge());

Console.Read();

}

}

