Chaque année, le Comité d’entreprise (CE) d’une entreprise régionale organise un achat groupé de jouets pour les enfants du personnel. Le catalogue des jouets proposés est disponible sur l’intranet de la société. Vous êtes chargés de la finalisation de cette application au sein de la société ERGOSUM.

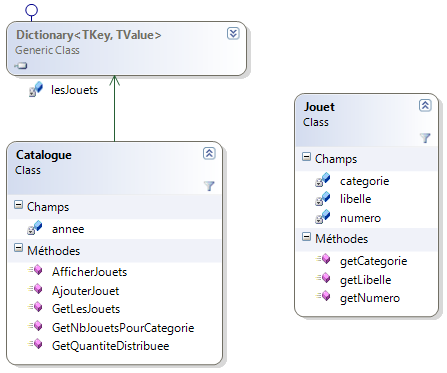
Le diagramme des classes utilisées figure en annexe 1.

Le code partiel des classes utilisées figure en annexe 2.

Annexe 3 : Extrait de la documentation de la classe Dictionary.

|  |  |
| --- | --- |
| **Travail à faire :** | |
| 3.1 | Écrire la méthode *GetLesJouets()* de la classe *Catalogue* (ligne 42). |
| 3.2 | Écrire la méthode *AfficherJouets()* de la classe *Catalogue (*ligne 60*)*. |
| 3.3 | Ecrire la méthode *AjouterJouet* de la classe *Catalogue (*ligne 52*)*. |
| 3.4 | Écrire la méthode GetQ*uantiteDistribuee()* de la classe *Catalogue (*ligne 68*)*. |
| 3.5 | Écrire la méthode *GetNbJouetsPourCategorie()* de la classe *Catalogue (*ligne 78*)*. |

**Annexe 1 : diagramme de classes**



Remarque : le dictionnaire (*Dictionary)* permet d’enregistrer pour chaque jouet sa quantité dans le catalogue.

**Annexe 2 : code partiel des classes utilisées**

class Jouet

{

private int numero;

private string libelle;

private string categorie;

/// <summary>

/// Instancie un objet Jouet

/// </summary>

public Jouet(int numero, string libelle, string categorie)

{

this.numero = numero;

this.libelle = libelle;

this.categorie = categorie;

}

// accesseurs

public int getNumero() { return this.numero; }

public string getLibelle() { return this.libelle; }

public string getCategorie() { return this.categorie; }

}

class Catalogue

{

private int annee;

/// <summary>

/// Contient pour chaque jouet du catalogue :

///- en clé, l’objet de la classe Jouet

/// - en valeur, la quantité de ce jouet distribuée pour ce catalogue

/// </summary>

private Dictionary<Jouet, int> lesJouets;

public Catalogue(int annee)

{

this.annee = annee;

this.lesJouets = new Dictionary<Jouet, int>();

}

/// <summary>

/// Accesseur en lecture pour lesJouets

/// </summary>

public Dictionary<Jouet, int> GetLesJouets()

{

// à compléter

}

/// <summary>

/// Ajoute au dictionnaire un jouet et sa quantité

/// retourne vrai si l'objet n'était pas déjà présent,

/// faux dans le cas contraire

/// </summary>

public bool AjouterJouet(Jouet jouet, int quantite)

{

// à compléter

}

/// <summary>

/// Affiche les libellés des jouets du catalogue.

/// </summary>

public void AfficherJouets()

{

// à compléter

}

/// <summary>

/// Retourne la quantité totale de jouets distribués pour ce catalogue.

/// </summary>

public int GetQuantiteDistribuee()

{

// à compléter

}

/// <summary>

/// Retourne le nombre de jouets de la catégorie passée en paramètre.

/// </summary>

/// <param name="categorie"></param>

/// <returns></returns>

public int GetNbJouetsPourCategorie(string categorie)

{

// à compléter

}

}

class Program

{

static void Main(string[] args)

{

Catalogue cat = new Catalogue(2015);

cat.AjouterJouet(new Jouet(1, "Monopoly", "Jeu de société"), 10);

cat.AjouterJouet(new Jouet(2, "Playstation", "Console"), 20);

cat.AjouterJouet(new Jouet(3, "Risk", "Jeu de société"), 15);

cat.AfficherJouets();

Console.WriteLine(cat.GetQuantiteDistribuee());

Console.WriteLine(cat.GetNbJouetsPourCategorie("Console"));

Console.WriteLine(cat.GetNbJouetsPourCategorie("Jeu de société"));

}

}

Ce programme affichera :

Monopoly

Playstation

Risk

45

20

25

**Annexe 3 : Extrait de la documentation de la classe Dictionary**

**Propriétés**

|  |
| --- |
| Nom |
| [Count](http://msdn.microsoft.com/fr-fr/library/zhcy256f(v=vs.100).aspx) | Obtient le nombre de paires clé/valeur contenues dans **Dictionary<TKey, TValue>**. |
| [Item](http://msdn.microsoft.com/fr-fr/library/9tee9ht2(v=vs.100).aspx) | Obtient ou définit la valeur associée à la clé spécifiée.  **Dictionary<TKey, TValue>.[TKey *key*]**  Exemple d’affichage d’un élément du dictionnaire lesJouets :  *// on suppose que le dictionnaire lesJouets a été instancié et possède des objets*  *Console.WriteLine(lesJouets[clé]) ; // clé est de type Jouet* |
| [Keys](http://msdn.microsoft.com/fr-fr/library/yt2fy5zk(v=vs.100).aspx) | Obtient une collection contenant les clés dans **Dictionary<TKey, TValue>**.  Exemple de parcours d’une collection de clés :  *Dictionary<Jouet, int> lesJouets = new Dictionary<Jouet, int>() ;*  *foreach( Jouet jouet in lesCles.Keys )*  *{ // faire quelque chose }* |
| Values | Obtient une collection contenant les valeurs de **Dictionary<TKey, TValue>.**  Exemple de parcours d’une collection de valeurs :  *Dictionary<Jouet, int> lesJouets = new Dictionary<Jouet, int>() ;*  *foreach( int quantite in lesCles.Values )*  *{ // faire quelque chose }* |

[**Méthodes**](javascript:void(0))

|  |  |
| --- | --- |
| Nom | Description |
| [Add](http://msdn.microsoft.com/fr-fr/library/k7z0zy8k(v=vs.100).aspx) | Ajoute la clé et la valeur spécifiées au dictionnaire.  **Dictionary<TKey, TValue>.Add(TKey *key*,TValue *value*)**  Une exception **ArgumentException** est lancée si la clé existe déjà dans le dictionnaire. |
| [ContainsKey](http://msdn.microsoft.com/fr-fr/library/kw5aaea4(v=vs.100).aspx) | Détermine si **Dictionary<TKey, TValue>** contient la clé spécifique.  Bool b = **Dictionary<TKey, TValue>.ContainsKey(TKey *key*);** |