Exercice 1

L'erreur suivante signifie que le moteur d'exécution a refusé l'expression « prix = PrixAbonnement ; » au moment de son exécution. Il faut donc utiliser le mot clé base afin d'avoir plutôt une ligne du genre : prix = base.PrixAbonnement() afin de pouvoir appeler la méthode PrixAbonnement de la classe mère Abonnement et non celle de la classe fille qui génèrera une boucle récursive infinie.

Exercice 2

executeReader() : Génère un objet de type DataReader permettant de lire les données issues d'une requête CommandText contenant une instruction SQL de type SELECT.

executeScalar() : Retourne la première colonne de la première ligne d'un résultat de requête SQL. Sert à récupérer le résultat d'un COUNT par exemple (mais il faut comparer les références de types avec TypeOf).

executeNonQuery() : Retourne le nombre de lignes affectées par une instruction SQL de type Transaction lors d'une requête INSERT, UPDATE ou DELETE notamment.

Exercice 3

Fonctions :

Attributs privés de la classe :

Private string role;

Private int duree;

Private DateTime date;

Private List<Fonction> historique ;

En-tête des méthodes :

Public DateTime GetDateFin() ;

Public DateTime GetDateDebut() ;

Collaborateur :

Attributs privés de la classe :

Private string nom;

Private float salaire;

Private List<Projet> mesProjetsEnCours ;

En-tête des méthodes :

Public float GetSalaire() ;

Projet :

Attributs privés de la classe :

Private string nom;

Private DateTime deadline;

Private List<Tache> listeTaches ;

En-tête des méthodes :

Public float GetTauxAvancement() ;

Public void newOperation();

Projet :

Attributs privés de la classe :

Private string etat;

Private DateTime dateDebut;

En-tête des méthodes :

Public DateTime GetDuree() ;

Exercice 4

Premier WriteLine : Compare les références de a et de b et comme a et b possèdent les mêmes références donc a == b renvoi true. ia et ib ne possèdent pas les mêmes références car ce sont 2 objets contenant 2 entiers n'ayant pas les mêmes valeurs, donc les valeurs de ia et ib ne sont pas identiques donc ia == ib renverra false.

Second WriteLine : Ce coup ci on compare les références de a et b et celles de ia et ib. Comme a et b sont tous les 2 des entiers avec la même valeur, a.Equals(b) renverra true. Comme ia et ib sont de type Object également, ia.Equals(ib) renverra true également.

Mission 2

1°)

Licence Jeune

```
public string GetDescription()
{
   return this.telResp + base.GetDescription();
}
```

Licence Mixte

```
public string GetDescription()
{
   return this.laEntreprise.GetNom() + base.GetDescription();
}
```

2°)

Licence :

```
public bool estActive()
{
 bool active = false;
 if(DateTime.Year == annee)
   active = true;
 return active;
}
```

Club :

```
public List<Licence> getLicencesActives()
```

```
{
 List<Licence> listeLicences = new List<Licence>();
 foreach(Licence l in lesLicences)
 {
  if(l.estActive())
    listeLicences.Add(l);
 }
 return listeLicences;
}
```

LigueRegionale :

```
public Dictionary<string, int> getNbLicencesParCategorie()
{
 Dictionnary<string, int> dicoCateg = new Dictionnary<string,int>();

 List<Licence> licencePoussin = new List<Licence>();
 List<Licence> licenceBenjamin = new List<Licence>();
 List<Licence> licenceJunior = new List<Licence>();
 foreach(Club c in lesClubs)
 {
  foreach(Licence l in c)
  {
    if(l.GetCategorie() == "Poussin")
      licencePoussin.add(l);
    if(l.GetCategorie() == "Benjamin")
      licenceBenjamin.add(l);
    if(l.GetCategorie() == "Junior")
      licenceJunior.add(l);
  }

  dicoCateg.Add("Poussin", licencePoussin.Count);
  dicoCateg.Add("Benjamin", licenceBenjamin.Count);
```

```
   dicoCateg.Add("Junior", licenceJunior.Count);
   L.Clear();
  }
  return dicoCateg;
}
```

Autre Variante :

```
public Dictionary<string, int> getNbLicencesParCategorie()
{
  Dictionnary<string, int> dicoCateg = new Dictionnary<string,int>();

  List<Licence> licencePoussin = new List<Licence>();
  List<Licence> licenceBenjamin = new List<Licence>();
  List<Licence> licenceJunior = new List<Licence>();
  foreach(Club c in lesClubs)
  {
   foreach(Licence l in c)
   {
    switch(l.GetCategorie())
    {
     case "Poussin":
      licencePoussin.add(l);
      break;
     case "Benjamin":
      licenceBenjamin.add(l);
      break;
     case "Junior":
      licenceJunior.add(l);
      break;
    }
   }
```

```
    dicoCateg.Add("Poussin", licencePoussin.Count);
    dicoCateg.Add("Benjamin", licenceBenjamin.Count);
    dicoCateg.Add("Junior", licenceJunior.Count);
  }
  return dicoCateg;
}
```