

# Quelques sources de l'exemple final

---

## Les interfaces

---

```
public interface Joueur {
    public Commande getCommande();
    public Statut getStatut();
    public void addNavire(Navire nav);
    public Vector<Navire> getListeNavire();
    public int getId();
}
```

```
public interface JHumain extends Joueur {
    public void interrogationParClavier();
}
```

---

## Les objets “accessoires” (énumérations, exceptions)

---

```
public class OccupException extends Exception {
    public OccupException(String string) {
        super(string) ;
    }
}
```

```
public enum Direction {
    NORD,
    SUD,
    EST,
    OUEST
}
```

---

## Les objets principaux

---

```
public class Equipe implements JHumain, JIA {
    protected Vector<Navire> listeNavire;
    protected Statut myStatut;
    protected Commande myCommande;
    protected Nature myNature ;
    protected int ident ;
    private Random rd;
    private Scanner sc;

    public Equipe(Nature n) {
        this.listeNavire = new Vector<Navire>() ;
        this.myNature = n;
        rd = new Random() ;
        sc = new Scanner(System.in);
    }
}
```

```

public void setStatut(Statut s) {
    myStatut = s ;
}

public Statut getStatut() {
    return myStatut;
}

public Commande getCommande() {
    if (myNature == Nature.HUMAIN)
        interrogationParClavier() ;
    else
        tirageAleatoire() ;
    System.out.println("Commande de l'equipe "+ident+" "+myCommande);
    return myCommande;
}

public void tirageAleatoire() {
    int idNav = rd.nextInt(listeNavire.size()) ;
    idNav = listeNavire.get(idNav).getIdent() ;
    int idAction = rd.nextInt(2) ;
    if ((myStatut == Statut.NEUTRE) &&(idAction == 1))
        idAction ++ ;
    int idDirection = -1 ;
    if (idAction <2)
        idDirection = rd.nextInt(4) ;
    myCommande = new Commande(this,idNav,idAction,idDirection) ;
}

public void interrogationParClavier() {
    String strIdNav = "" ;
    String strAction = "" ;
    String strDirection = "" ;
    System.out.println("Veuillez saisir le numero du navire qui va agir (de 0 a "+
        (listeNavire.size()-1)+") :)");
    strIdNav = sc.nextLine();
    System.out.println("Vous avez saisi : " + strIdNav);
    if (myStatut == Statut.MILITAIRE)
        System.out.println("Veuillez saisir l'action a faire (DEPLACEMENT, TIR :)");
    else
        System.out.println("Veuillez saisir l'action a faire (DEPLACEMENT, PECHE :)");
    strAction = sc.nextLine();
    System.out.println("Vous avez saisi : " + strAction);
    if ((strAction.compareTo("DEPLACEMENT")== 0) ||(strAction.compareTo("TIR")== 0)) {
        System.out.println("Veuillez saisir la direction ds laquelle agir
            (NORD, SUD, EST, OUEST :)");
        strDirection = sc.nextLine();
        System.out.println("Vous avez saisi : " + strDirection);
    }
    myCommande = new Commande(this,strIdNav,strAction,strDirection) ;
}

public String toString() {
    return "Equipe "+ident+" (" + myStatut + "," + myNature + "), avec " + listeNavire;
}

```

```

    public void addNavire(Navire nav) {
        listeNavire.add(nav) ;
    }

    public Vector<Navire> getListeNavire() {
        return listeNavire;
    }

    public int getId() {
        return ident;
    }
}

```

---

```

public class EqBataillon extends Equipe {
    public EqBataillon(int idEq, Nature n) {
        super(n) ;
        myStatut = Statut.MILITAIRE ;
        ident = idEq ;
    }
}

```

---