Rappel de programmation objet

François Schwarzentruber
ENS Cachan – Antenne de Bretagne

Programmation objet

- Est-ce une programmation où les données sont centrales ?

ennemi.avancer()

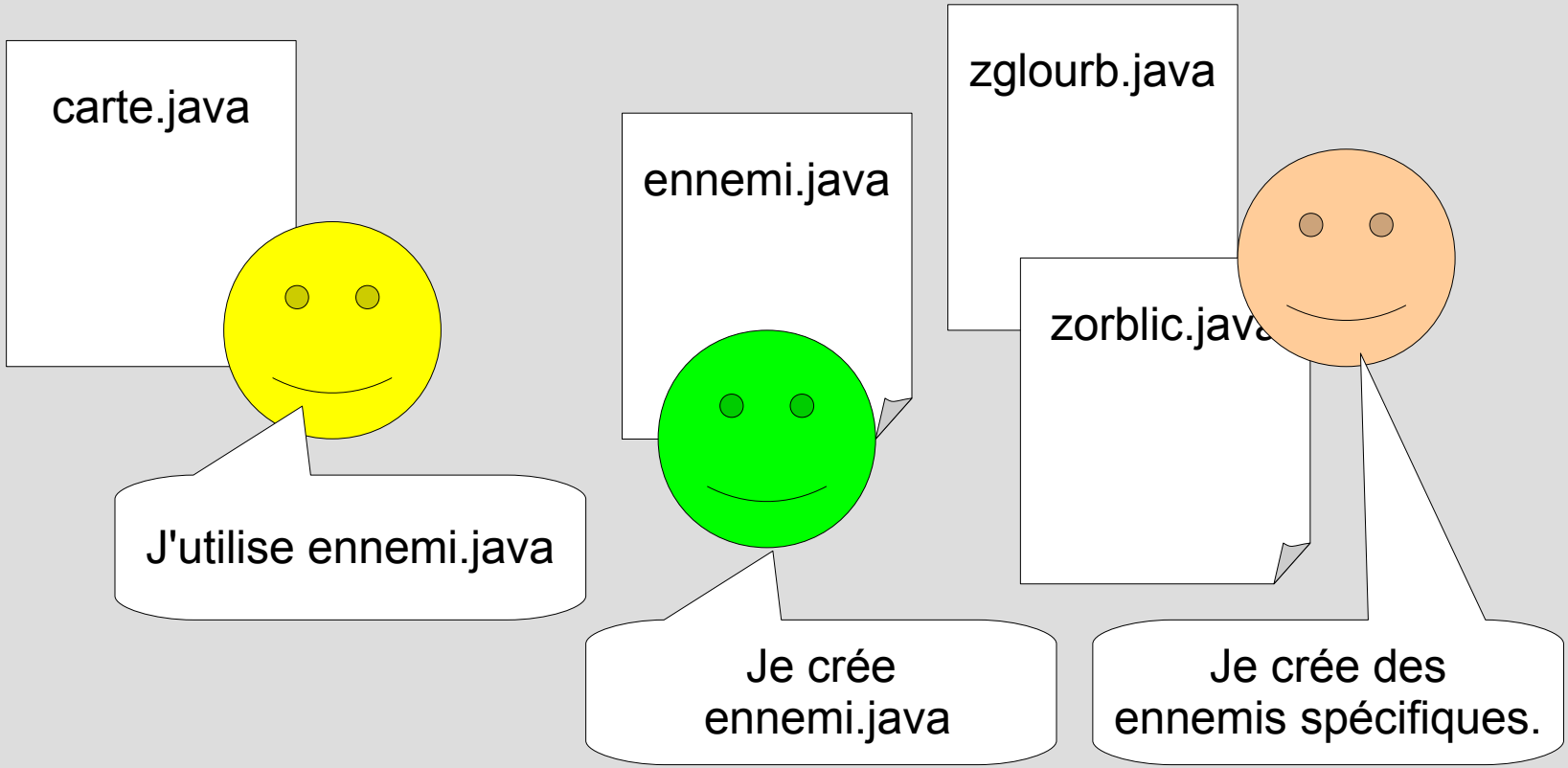
ennemiAvancer(ennemi)

Programmation objet

- Opposée à la programmation impérative et à la programmation fonctionnelle ?

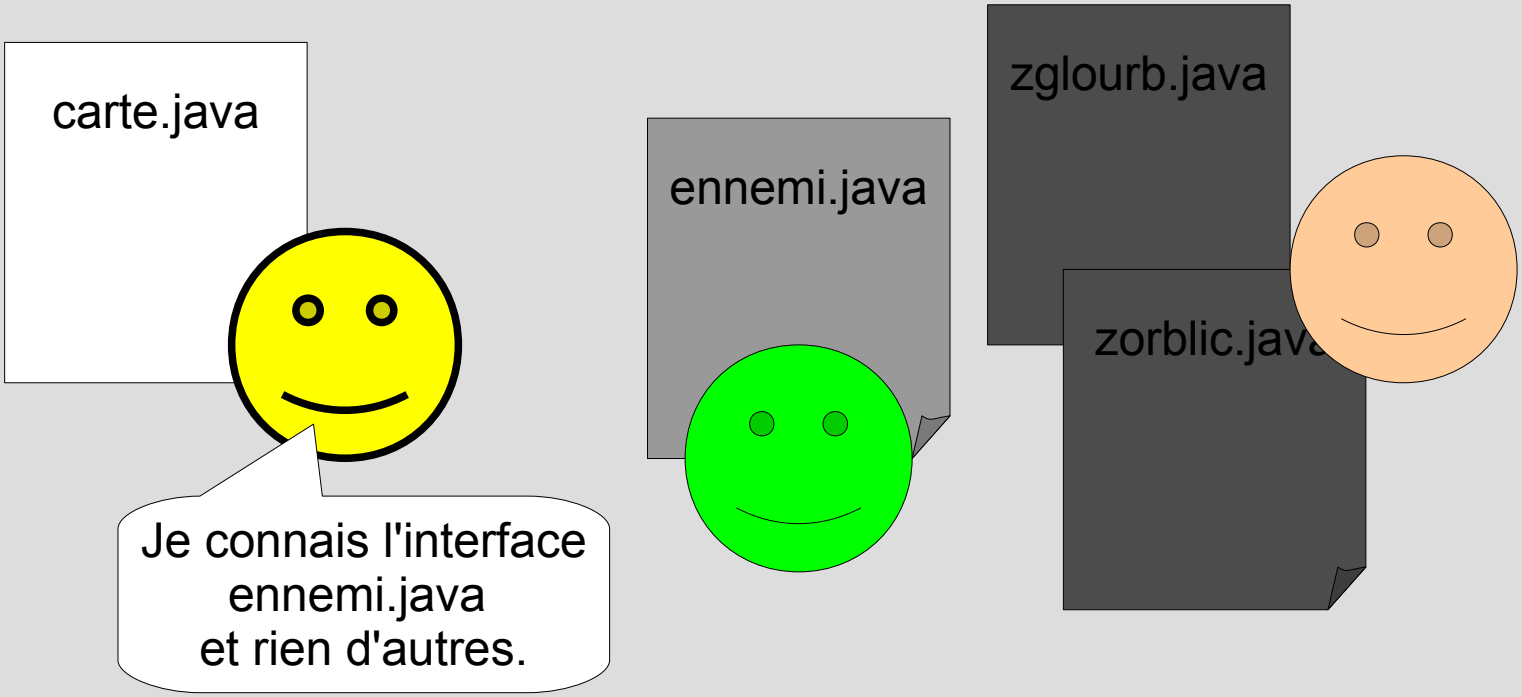
Programmation objet

Un paradigme qui s'intéresse à **QUI** programme et **QUI** utilise.



Programmation objet

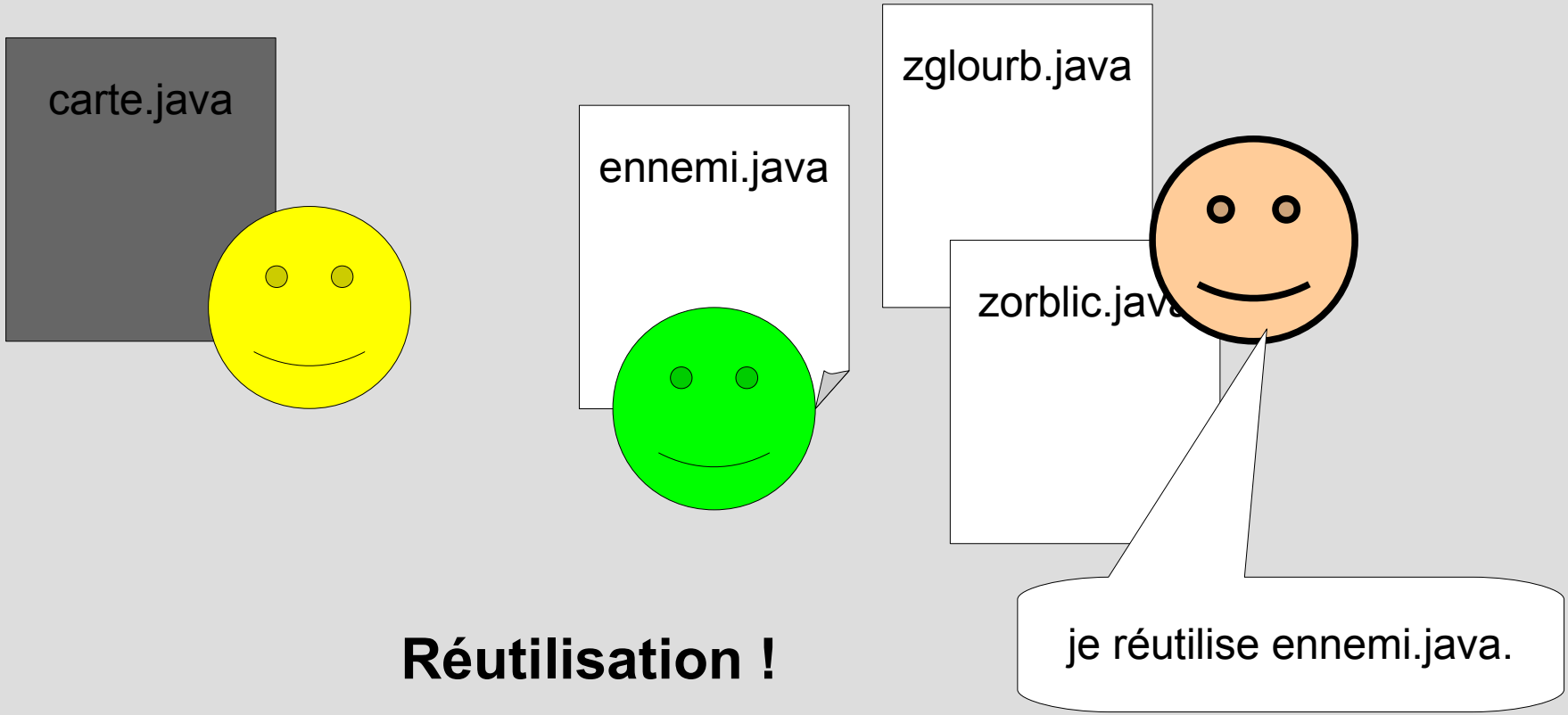
Un paradigme qui s'intéresse à **QUI** programme et **QUI** utilise.



Encapsulation !

Programmation objet

Un paradigme qui s'intéresse à **QUI** programme et **QUI** utilise.



Réutilisation !

Encapsulation

Interface

- Opérations

Implémentations : classes

- Méthodes

Réutilisation

- Classe abstraite / concrète
- Héritage
- Délégation

Classe concrète

- Toutes les opérations sont implémentées

Classe abstraite

- Il existe une opération non implémentée.

Héritage

```
abstract class Chien
{
    Yeux yeux;

}


class Labrador extends Chien
{
    int longueurOreille;
}
```

Couplage fort
Si Chien change,
Labrador change.

Délégation

```
class Chien
{
    Yeux yeux ;
    void marcher()
    {
        ...
    }
}
```

```
class Labrador
{
    final private Chien c;

    Labrador()
    {
        c := new Chien();
    }

    void marcher()
    {
        c.marcher();
    }
}
```

Délégation

```
class Chien
{

}


class ChienSauvage extends Chien
{

}


class ChienAvecColier extends Chien
{

}
```

```
class Labrador
{
    final private Chien c;

    Labrador(Chien c)
    {
        this.c := c;
    }

    void marcher()
    {
        c.marcher();
    }
}
```

```
new Labrador(new ChienSauvage())
new Labrador(new ChienAvecColier())
```