

## 4. Sérialisation d'objets

## **Objectif:**

Transfert d'un objet et de son contexte (+ ses liens avec d'autres objets) entre applications

Ex:

- *échange d'objets entre une application serveur et une application client;*
- *transfert d'objets via internet, intranet;*
- *écriture/lecture d'objets dans des fichiers;*

## **Contrainte:**

On doit pouvoir récupérer l'objet et le réutiliser en tant qu'objet

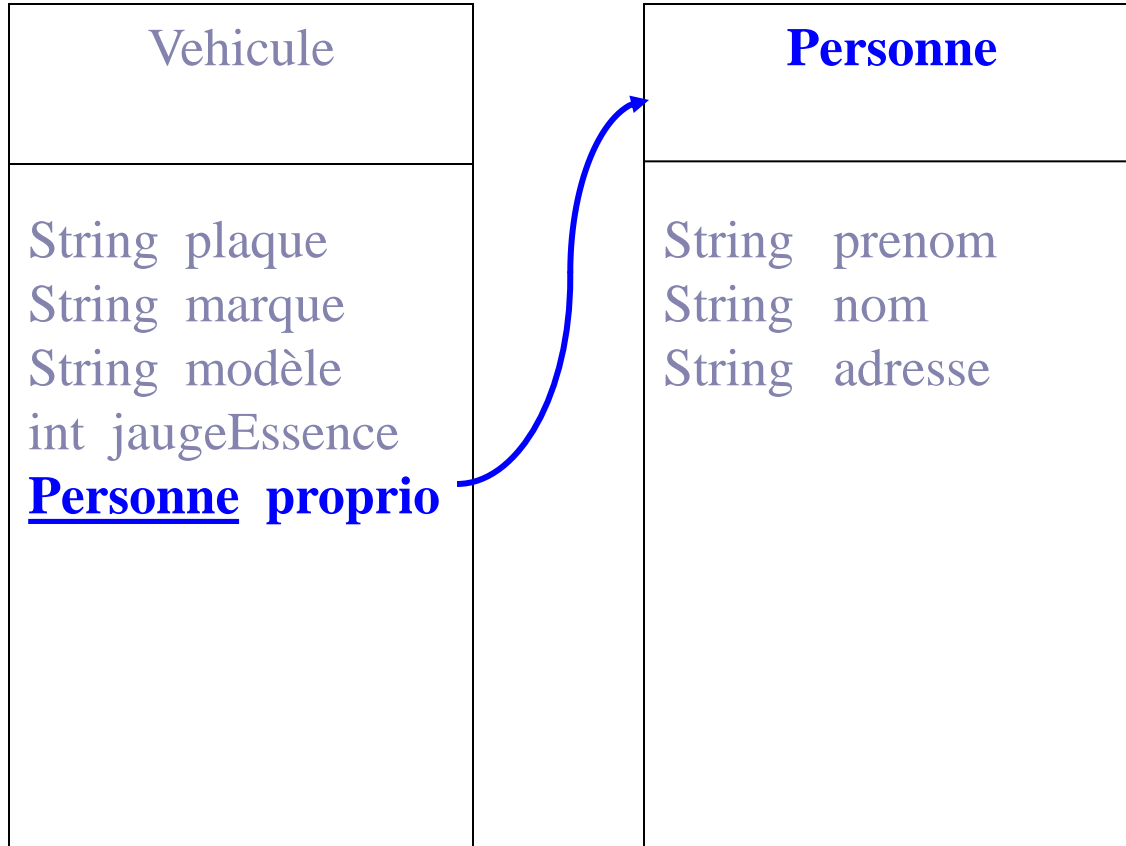
Ex: on doit pouvoir appeler des méthodes sur cet objet

### Contexte d'un objet?

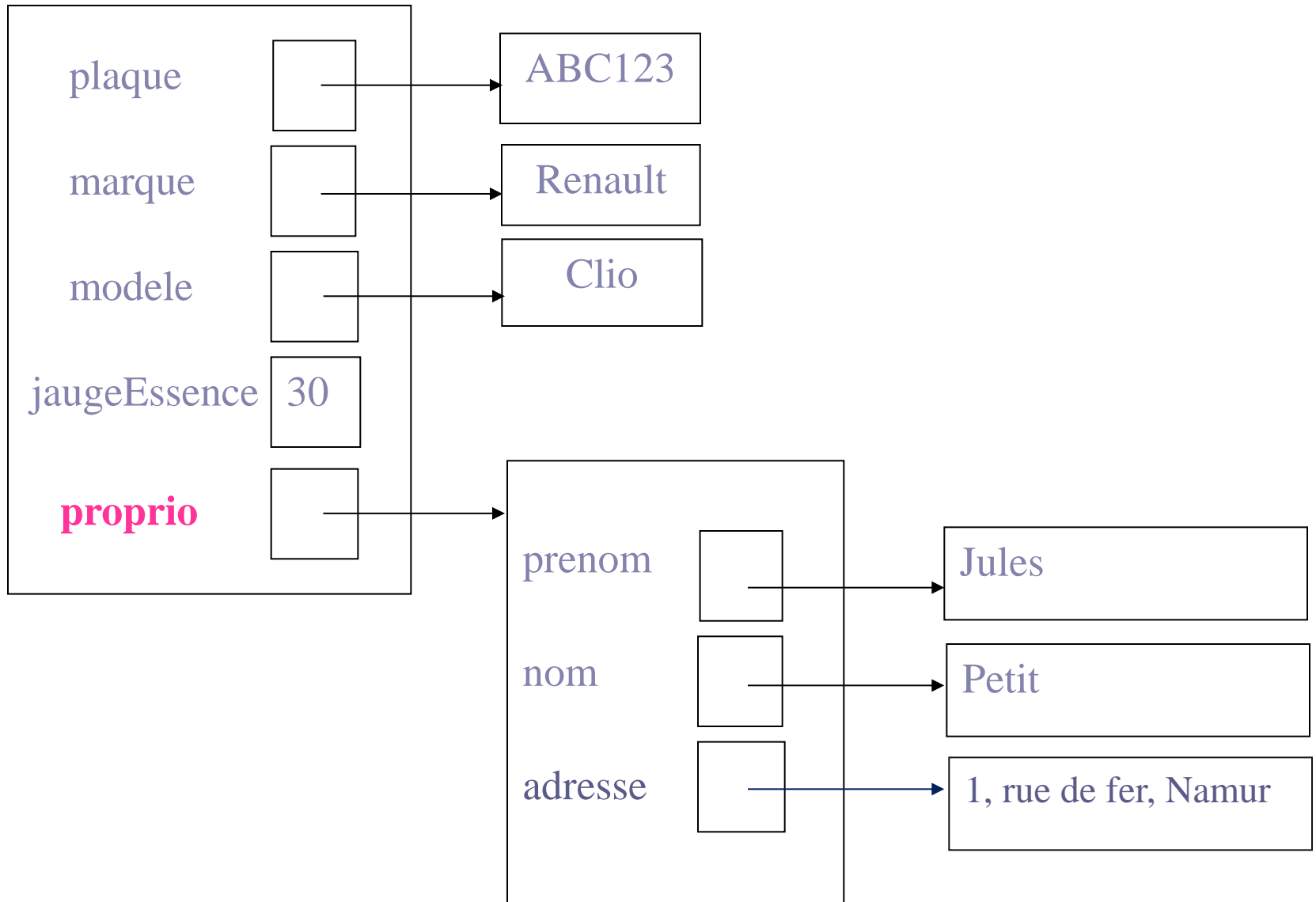
Les valeurs de ses variables d'instances

- si de type primitif : la valeur
- si de type référence: l'objet relié

Exemple :



## Objet *veh*



*Dans l'exemple précédent, le contexte de tout objet véhicule est en fait constitué de 8 objets reliés: 1 objet Vehicule + 1 objet Personne + 6 objets String.*

La sérialisation d'un objet (que l'on veut transférer) permet de

- transférer tout son contexte (tous les objets reliés)
- pouvoir réutiliser l'objet en tant que tel (appeler des méthodes, ...)

Syntaxe: **implements Serializable**

*sur toutes les classes reliées!*

```
import java.io.*;
```

```
public class Personne implements Serializable
```

```
{ private String prenom, nom, adresse;
```

```
    public Personne(String p, String n, String l)
```

```
    {...}
```

```
    public String toString( )
```

```
    {...}
```

```
}
```

```
import java.io.*;
```

```
public class Vehicule implements Serializable
```

```
{ private String plaque, marque, modele;
```

```
    private int jaugeEssence;
```

```
    private Personne proprio;
```

```
    public Vehicule(...) { ... }
```

```
    public String toString( )
```

```
    { return "... " + proprio; }
```

```
}
```



## Exemple :

