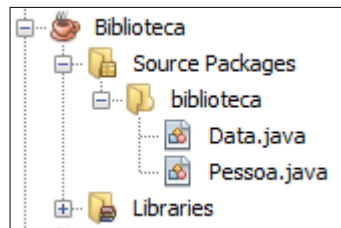


**Objetivos Específicos:** Reutilização de código por agregação e composição de classes. Referências partilhadas. Objetos: mutáveis e imutáveis. Clonagem de objetos. Construtor de cópia. Bibliotecas de classes no ambiente de desenvolvimento Netbeans. *Packages*.

### Exercício

1. Criar uma **biblioteca de classes** no Netbeans, chamada **Biblioteca**, como é ilustrado na figura seguinte. As classes *Data* e *Pessoa* estão disponíveis no moodle.



2. Criar um novo **projeto Netbeans** chamado **Automovel2**.
3. Criar uma **package** chamada *testes* e adicionar uma classe principal para realizar testes do código desenvolvido. Esta classe deve ser designada **TesteAutomovel**.
4. Adicionar a **Biblioteca** ao projeto.
5. Na classe **TesteAutomovel**, criar e visualizar uma **instância** para representar a **pessoa** chamada **Jorge**, cujo número **ID Civil** do cartão do cidadão é **1234567**.
6. Criar a seguinte versão simplificada da classe **Automovel**.

```
public class Automovel {  
  
    private String marca;  
  
    private static final String MARCA_POR_OMISSAO = "sem marca";  
  
    public Automovel(String marca) {  
        this.marca = marca;  
    }  
  
    public Automovel() {  
        this(Automovel.MARCA_POR_OMISSAO);  
    }  
  
    public String getMarca() { return this.marca; }  
  
    public void setMarca(String marca) { this.marca = marca; }  
  
    public String toString() {  
        return "Automóvel: Marca=" + this.marca;  
    }  
}
```

7. Alterar a classe **Automovel** para representar também os **proprietários** dos automóveis, através da classe **Pessoa** armazenada na **Biblioteca**. Comece por esboçar um **diagrama de classes** para ilustrar o **relacionamento** entre as classes *Automovel* e *Pessoa*.
8. Criar e visualizar uma instância para representar o automóvel do **Jorge** que tem a marca **KIA**.
9. Atualize o nome do **Jorge** para **Jorge Paixão** na instância de **Pessoa** e programe nova **visualização** das duas instâncias criadas. Verifique que a instância automóvel também foi atualizada.

10. Alterar novamente a classe **Automovel** para representar também as **matrículas** dos automóveis, usando a classe abaixo apresentada (versão simplificada). Comece por atualizar o **diagrama de classes** para ilustrar o **relacionamento** entre as classes *Automovel*, *Matricula* e *Data*.

```
public class Matricula {
    private String matricula;
    private Data dataRegisto;    // Classe Data da Biblioteca

    private static String MATRICULA_POR_OMISSAO = "sem matrícula";
    private static Data DATAREGISTO_POR_OMISSAO = new Data();

    public Matricula(String matricula, Data dataRegisto) {
        this.matricula = matricula;
        this.dataRegisto = dataRegisto;
    }

    public Matricula() {
        this(Matricula.MATRICULA_POR_OMISSAO, Matricula.DATAREGISTO_POR_OMISSAO);
    }

    public Matricula(Matricula outraMatricula){
        this(outraMatricula.matricula, outraMatricula.dataRegisto);
    }

    public String getMatricula() { return this.matricula; }
    public Data getDataRegisto() { return this.dataRegisto; }

    public void setMatricula(String matricula) { this.matricula = matricula; }
    public void setDataRegisto(Data dataRegisto) { this.dataRegisto = dataRegisto; }

    @Override
    public String toString() {
        return this.matricula + " registada em " +
            this.dataRegisto.toAnoMesDiaString();
    }
}
```

11. Testar as alterações efetuadas na classe **Automovel** da seguinte forma:
- Atualizar e visualizar o automóvel do **Jorge Paixão** com a matrícula **11-11-AA**, registada em **2012/2/2**. Para isso, **comece** por criar uma instância de **Data** e uma instância de **Matricula**.
  - Atualizar a instância de **Matricula** com a matrícula **22-22-BB** e programar nova visualização dessa instância **matricula** e da instância **automóvel**.
  - Atualizar a instância de **Data** com a data de **2013/3/3** e programar nova visualização das instâncias **matricula** e **automóvel**.
  - Corrigir o problema.
12. O problema identificado pode verificar-se no atributo **marca** da classe **Automóvel**? Justifique.
13. Alterar a classe **Automovel** de modo a permitir a **clonagem** das suas instâncias através de um **construtor de cópia**.
14. Testar as alterações efetuadas.