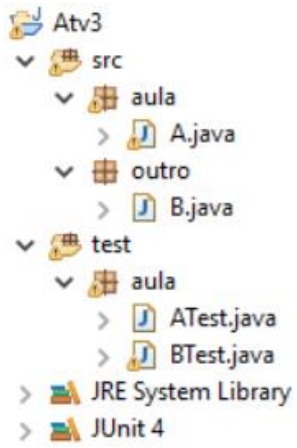
Instruções para a entrega: fazer os exercícios e mostrar para o professor na aula do dia **15/mar**. Os testes deverão ser executados no computador. A entrega pode ser em dupla. Alunos ausentes não terão a nota considerada.

Observações:

- Crie um projeto e adicione a biblioteca JUnit 4;
- Considere que as classes sejam colocadas no projeto assim como representado na figura a seguir;
- Todos os testes devem dar resultado "verde", isto é, deve ser aprovado no teste.




```java
package aula;

public class A {

  public A() {}

  public A(Object obj) throws Exception {
    if( !(obj instanceof Number) ) {
      throw new Exception(
                    "Precisa ser um número");
    }
  }

  public void f(int nro) {
    while( nro-- > 0);
  }

  public void g(int nro) throws Exception{
    if( nro > 10){
      throw new Exception(
                     "Valor maior que 10");
    }
    while( nro-- > 0);
  }

  boolean isImpar(int nro) {
    return nro%2 == 1 ? true : false;
  }

  protected int maior(int x, int y) {
    return x < y? y : x;
  }

  private int dobro(int nro) {
    return nro * 2;
  }

  public static int diff(int x, int y) {
    return x - y;
  }

  public int mod(int x, int y) {
    return x % y;
  }

  public class C{
    public int soma(int x, int y) {
      return x + y;
    }
  }
}
```

```java
package outro;

public class B extends aula.A{
  boolean isPar(int nro) {
    return nro%2 == 1 ? true : false;
  }

  protected int menor(int x, int y) {
    return x < y? y : x;
  }
}
```

Exercício 1 – No projeto de um sistema os métodos precisam ser codificados de modo a facilitarem o seu teste. Por que o método ***f***, da classe A, **não** está adequado para um teste unitário usando uma ferramenta automatizada de testes, assim como o JUnit?

Exercício 2 – Programar testes para o método ***g***, da classe ***A***, para as chamadas com os parâmetros 5 e 15. Observação: esses testes deverão ser codificados na classe ***ATest***.

Exercício 3 – Os métodos ***isImpar*** e ***isPar***, respectivamente, das classes ***A*** e ***B***, possuem visibilidade package, isto é, somente estão disponíveis dentro do próprio pacote.

Considerando que os métodos de testes ***isImparTest*()** e ***isParTest*()** estão nas classes ***ATest*** e ***BTest***, respectivamente. Explique o motivo de **não** ser possível testar o método ***isParTest*()** desta forma como o projeto se encontra, enquanto que o teste do método ***isImparTest*()** não possui problema algum para ser testado.

```
@Test
public void isImparTest() {
   assertTrue( a.isImpar(1) );
}
@Test
public void isParTest() {
   assertTrue( b.isPar(0) );
}
```

Exercício 4 – Os métodos ***maior*** e ***menor***, respectivamente, das classes ***A*** e ***B***, possuem visibilidade protected, isto é, somente estão disponíveis dentro do próprio pacote e pela herança.

Considerando que os métodos de testes ***maiorTest*()** e ***menorTest*()** estão nas classes ***ATest*** e ***BTest***, respectivamente. Explique o motivo de **não** ser possível testar o método ***menorTest*()** desta forma como o projeto se encontra, enquanto que o teste do método ***maiorTest*()** não possui problema algum para ser testado.

```
@Test
public void maiorTest() {
   assertEquals( 3, a.maior(2,3) );
}
@Test
public void menorTest() {
   assertEquals( 2, b.menor(2,3) );
}
```

Exercício 5 – O método ***dobro***, da classe ***A***, possui acesso private, isto é, ele só pode ser acessado de dentro do próprio objeto/classe.

Programar um teste para o método ***dobro***. Lembre-se que você não pode alterar a classe ***A***, então este teste não pode ser colocado dentre do classe ***A***.

Observação: algumas pessoas afirmam que um método privado não deve ser testado, pois o seu objetivo é fazer uma funcionalidade interna da classe que, por sua vez, será consumida por um método público do objeto. Desta forma, cabe apenas o teste dos métodos públicos que invocam esse método privado.

Exercício 6 – O resultado esperado de um construtor é uma instância da classe (objeto).

Programar testes para as seguintes chamadas do construtor:

```
new A(12.5)
new A("12.5")
```

Exercício 7 – Um membro estático é aquele que pertence à classe, isto é, para chamar esse membro não é necessário ter uma instância da classe.

Programar um teste para o método ***diff***, da classe ***A***.

Exercício 8 – É aconselhável usar classe interna (classe aninhada) somente quando a classe externa é a única que utiliza os objetos do tipo de dado interno.

Programar um teste para o método ***soma***, da classe interna ***C***.

Exercício 9 – Constitui boa prática colocar somente **1** asserção por método de teste.

Explique o motivo do texto “Terceiro” não ser impresso no console.

```
@Test
public void modTest() {
   System.out.println("Primeiro");
   assertEquals("11%2 == 1", 1, a.mod(11,2) );
   System.out.println("Segundo");
   assertEquals("11%4 == 2", 2, a.mod(11,4) );
   System.out.println("Terceiro");
   assertEquals("11%3 == 2", 2, a.mod(11,3) );
}
```