

## Atividade Parcial 1 – Exercícios

**Disciplina:** Orientação a Objetos  
**Data Disponibilização:** 31/03/2015

**Turma:** 4º, 5º e 6º Períodos  
**Professor:** José Alexandre Macedo

**Data da Entrega:** 21/04/2015

- **Exercícios**

### Exercício 1 – A classe Circle (círculo)

Implemente a classe Circle de acordo com o diagrama abaixo:

Circle
-radius:double = 1.0 -color:String = "red"
+Circle() +Circle(radius:double) +getRadius():double +getArea():double

- As variáveis devem ser privadas.
- No construtor sem argumentos inicialize o raio e a cor com os valores padrão.
- Implemente a operação de cálculo da área de um círculo no método getArea().
- Crie uma classe TestCircle para construir 2 objetos da classe Circle. O primeiro usando o construtor sem argumento e o segundo usando o construtor com argumento (passe um raio igual a 5).
- Imprima o raio e a área de cada círculo.
- Método toString(): Sobreescreva o método toString() na classe Circle para ele retornar o seguinte valor “"Circulo: raio=" + radius + " cor=" + color”
- Teste o método sobrescrito na classe de testes, em seguida remova o toString() deixando apenas o nome do objeto e veja o que acontece.

### Exercício 2 – As classes Author e Book

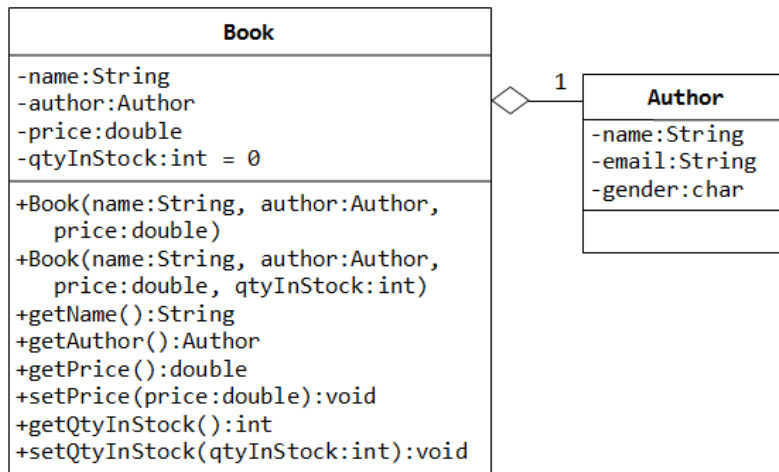
Implemente a classe Author de acordo com o diagrama abaixo:

Author
-name:String -email:String -gender:char
+Author(name:String, email:String, gender:char) +getName():String +getEmail():String +setEmail(email:String):void +getGender():char +toString():String

- Variáveis name, email e gender privadas
- O toString() deve retornar algo do tipo “name (gender) contato: email”

- Escreva a classe TestAuthor para criar dois autores e testar seus métodos públicos

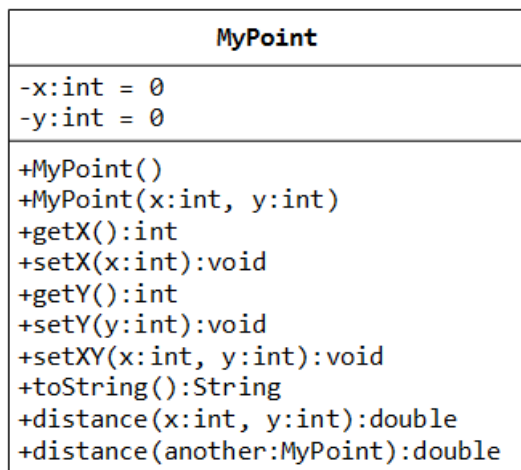
Crie a classe Book conforme o diagrama abaixo:



- A classe Book tem uma variável que é do tipo Author, essa variável se chama author.
- Todos os atributos devem ser privados
- Sobrescreva o toString() para retornar “book-name por author-name (gender) contato: email
- Escreva a classe TestBook para criar objetos da classe Book e testar seus métodos
- Imprima o autor (author) e o email de um objeto Book.
- Crie na classe Book o método getAuthorName

### Exercício 3 – A classe MyPoint

A classe MyPoint modela um ponto 2D com coordenadas x e y. Implemente a classe MyPoint de acordo com o diagrama abaixo:



- Todos os atributos devem ser privados
- Sobrescreva o toString() para retornar “(x, y)”
- O método **distance** retorna a distância do próprio ponto com relação ao x e y passados como argumento.
  - Dica:

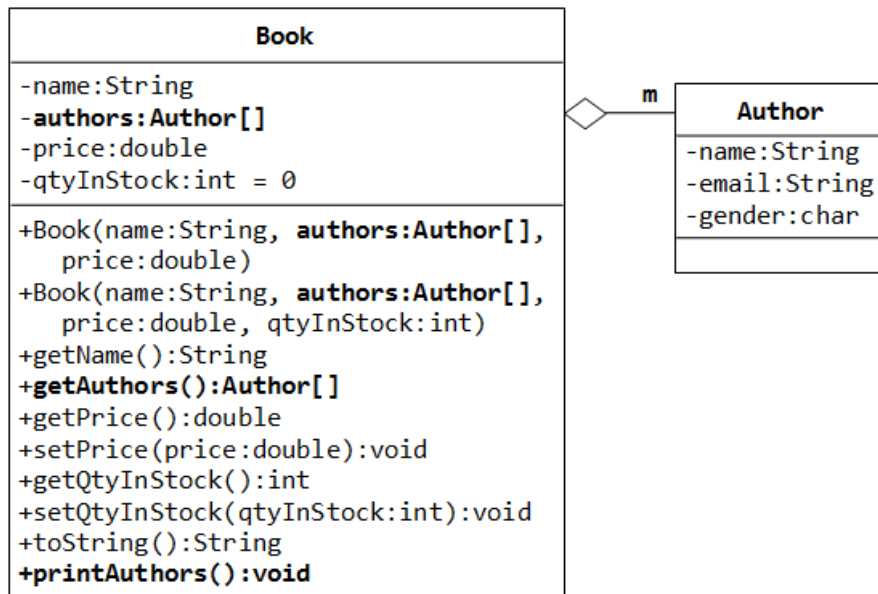
```

int diferencaX = this.x - x;
int diferencaY = .....
return Math.sqrt(diferencaX * diferencaX + diferencaY* diferencaY);

```

- O outro método **distance** retorna a distância do próprio ponto com relação a outro objeto da classe MyPoint.
- Crie a classe TestMyPoint e crie objetos para testar seus métodos
  - Escreva na classe TestMyPoint um vetor que aloca 10 pontos (objetos da classe MyPoint), inicialize os 10 pontos

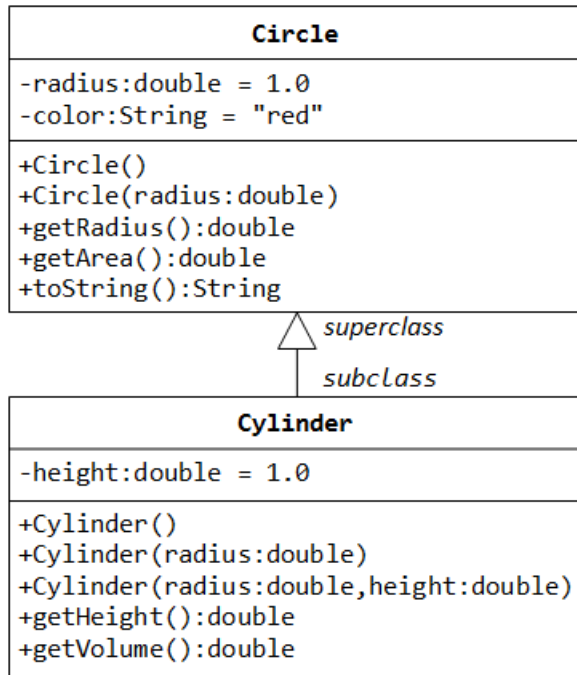
#### Exercício 4 – Alteração classe Book



- Altere a classe book e substitua o Author por um vetor de Author
- Crie um Book com 4 autores na classe teste

#### Exercício 5 – Classes Circle e Cylinder

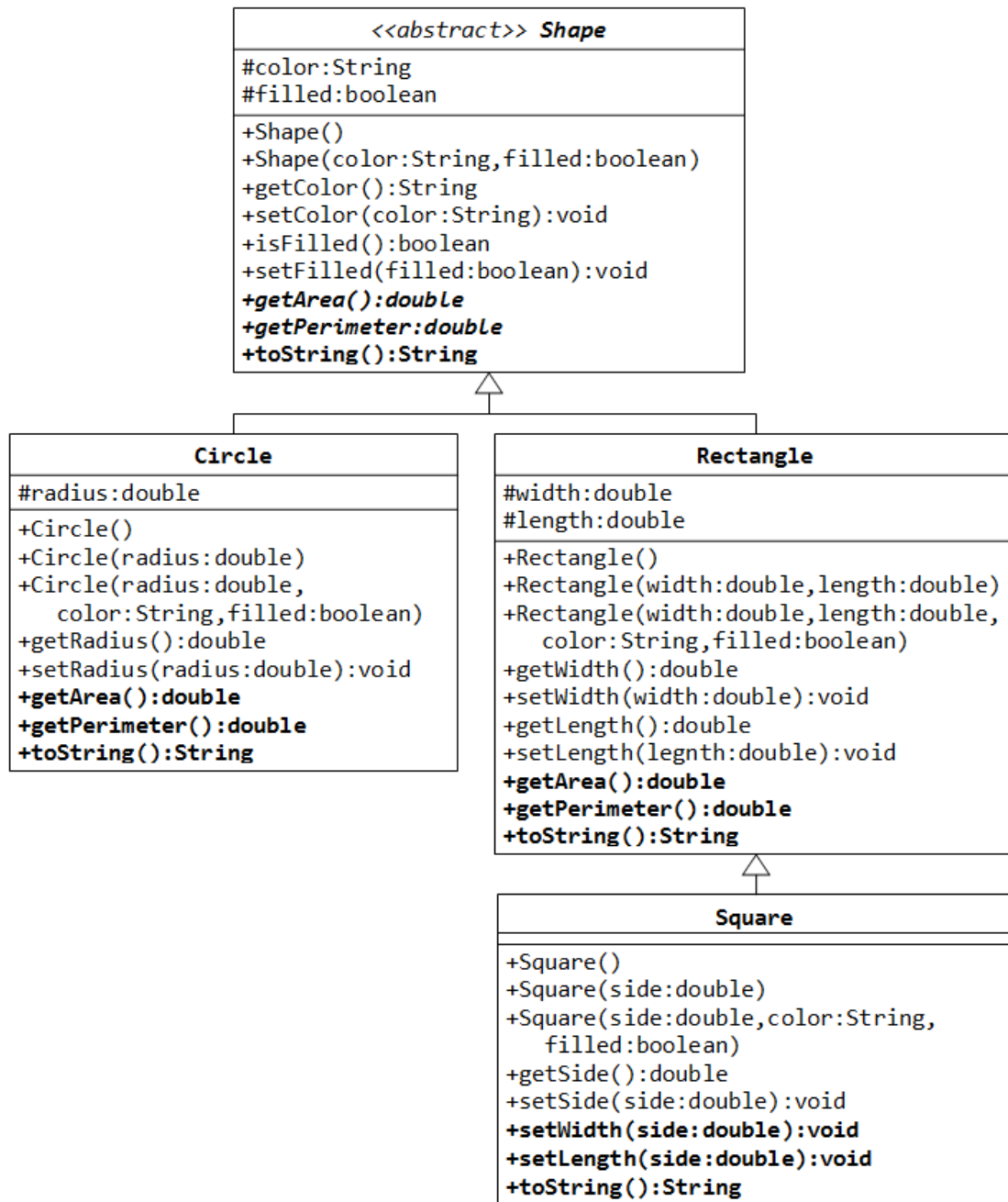
A classe Cylinder é derivada da superclasse Circle. Utilize herança para implementar a classe Cylinder de acordo com o diagrama abaixo.



- Após implementar a classe Cylinder, escreva a classe TestCylinder para criar objetos da classe Cylinder.
- Crie 3 cilindros na classe de teste e imprima para cada um seu raio, altura, área e volume.
- Sobreescreva o toString para exibir a seguinte mensagem: "Cylinder: subclass of " + super.toString() + " height=" + height;

## Exercício 6 – Herança

Implemente o diagrama abaixo.



- **Entregas**
  - Os códigos deverão ser enviados ao professor no dia 21/04 para o email (jamacedo@inf.ufes.br) com assunto “Atividade Parcial 1 – Orientação a Objetos”.
- **Grupos**

Os grupos devem ser de no máximo 3 alunos. A lista com os integrantes de cada grupo será definida no dia 31/03.

Boa Sorte!