Lista de exercícios de POO em Python e Java – Com solução

Esta foi retirada do site: https://wiki.python.org.br/ListaDeExercicios e implementada em Python e Java

---

1. **Classe Bola:** Crie uma classe que modele uma bola:
    a. Atributos: Cor, circunferência, material
    b. Métodos: trocaCor e mostraCor
2. **Classe Quadrado:** Crie uma classe que modele um quadrado:
    a. Atributos: Tamanho do lado
    b. Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área;
3. **Classe Retangulo:** Crie uma classe que modele um retangulo:
    a. Atributos: LadoA, LadoB (ou Comprimento e Largura, ou Base e Altura, a escolher)
    b. Métodos: Mudar valor dos lados, Retornar valor dos lados, calcular Área e calcular Perímetro;
    c. Crie um programa que utilize esta classe. Ele deve pedir ao usuário que informe as medidades de um local. Depois, deve criar um objeto com as medidas e calcular a quantidade de pisos e de rodapés necessárias para o local.
4. **Classe Pessoa:** Crie uma classe que modele uma pessoa:
    a. Atributos: nome, idade, peso e altura
    b. Métodos: Envelhercer, engordar, emagrecer, crescer. Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.
5. **Classe Conta Corrente:** Crie uma classe para implementar uma conta corrente. A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo. Os métodos são os seguintes: alterarNome, depósito e saque; No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.
6. **Classe TV:** Faça um programa que simule um televisor criando-o como um objeto. O usuário deve ser capaz de informar o número do canal e aumentar ou diminuir o volume. Certifique-se de que o número do canal e o nível do volume permanecem dentro de faixas válidas.
7. **Classe Bichinho Virtual:**Crie uma classe que modele um Tamagushi (Bichinho Eletrônico):
    a. Atributos: Nome, Fome, Saúde e Idade b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade Obs: Existe mais uma informação que devemos levar em consideração, o Humor do nosso tamagushi, este humor é uma combinação entre os atributos Fome e Saúde, ou seja, um campo calculado, então não devemos criar um atributo para armazenar esta informação por que ela pode ser calculada a qualquer momento.
8. **Classe Macaco:** Desenvolva uma classe Macaco,que possua os atributos nome e bucho (estomago) e pelo menos os métodos comer(), verBucho() e digerir(). Faça um programa ou teste interativamente, criando pelo menos dois macacos, alimentando-os com pelo menos 3 alimentos diferentes e verificando o conteúdo do estomago a cada refeição. Experimente fazer com que um macaco coma o outro. É possível criar um macaco canibal?
9. **Classe Ponto e Retangulo:** Faça um programa completo utilizando funções e classes que:
    a. Possua uma classe chamada Ponto, com os atributos x e y.
    b. Possua uma classe chamada Retangulo, com os atributos largura e altura.
    c. Possua uma função para imprimir os valores da classe Ponto
    d. Possua uma função para encontrar o centro de um Retângulo.
    e. Você deve criar alguns objetos da classe Retangulo.
    f. Cada objeto deve ter um vértice de partida, por exemplo, o vértice inferior esquerdo do retângulo, que deve ser um objeto da classe Ponto.
    g. A função para encontrar o centro do retângulo deve retornar o valor para um objeto do tipo ponto que indique os valores de x e y para o centro do objeto.
    h. O valor do centro do objeto deve ser mostrado na tela

i. Crie um menu para alterar os valores do retângulo e imprimir o centro deste retângulo.

10. **Classe Bomba de Combustível:** Faça um programa completo utilizando classes e métodos que:

    a. Possua uma classe chamada bombaCombustível, com no mínimo esses atributos:
        i. tipoCombustivel.
        ii. valorLitro
        iii. quantidadeCombustivel
    b. Possua no mínimo esses métodos:
        i. abastecerPorValor( ) – método onde é informado o valor a ser abastecido e mostra a quantidade de litros que foi colocada no veículo
        ii. abastecerPorLitro( ) – método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.
        iii. alterarValor( ) – altera o valor do litro do combustível.
        iv. alterarCombustivel( ) – altera o tipo do combustível.
        v. alterarQuantidadeCombustivel( ) – altera a quantidade de combustível restante na bomba.

    OBS: Sempre que acontecer um abastecimento é necessário atualizar a quantidade de combustível total na bomba.

11. **Classe carro:** Implemente uma classe chamada Carro com as seguintes propriedades:

    a. Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque.
    b. O consumo é especificado no construtor e o nível de combustível inicial é 0.
    c. Forneça um método andar( ) que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.
    d. Forneça um método obterGasolina( ), que retorna o nível atual de combustível.
    e. Forneça um método adicionarGasolina( ), para abastecer o tanque. Exemplo de uso:

```
f. meuFusca = Carro(15);            # 15 quilômetros por litro de combustível.
g. meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.
h. meuFusca.andar(100);             # anda 100 quilômetros.
   meuFusca.obterGasolina()         # Imprime o combustível que resta no tanque.
```

12. **Classe Conta de Investimento:** Faça uma classe contaInvestimento que seja semelhante a classe contaBancaria, com a diferença de que se adicione um atributo taxaJuros. Forneça um construtor que configure tanto o saldo inicial como a taxa de juros. Forneça um método adicioneJuros (sem parâmetro explícito) que adicione juros à conta. Escreva um programa que construa uma poupança com um saldo inicial de R$1000,00 e uma taxa de juros de 10%. Depois aplique o método adicioneJuros() cinco vezes e imprime o saldo resultante.

13. **Classe Funcionário:** Implemente a classe Funcionário. Um empregado tem um nome (um string) e um salário(um double). Escreva um construtor com dois parâmetros (nome e salário) e métodos para devolver nome e salário. Escreva um pequeno programa que teste sua classe.

14. Aprimore a classe do exercício anterior para adicionar o método aumentarSalario (porcentualDeAumento) que aumente o salário do funcionário em uma certa porcentagem.
    - Exemplo de uso:

```
o   harry=funcionário("Harry",25000)
    harry.aumentarSalario(10)
```

15. **Classe Bichinho Virtual++:** Melhore o programa do bichinho virtual, permitindo que o usuário especifique quanto de comida ele fornece ao bichinho e por quanto tempo ele brinca com o bichinho. Faça com que estes valores afetem quão rapidamente os níveis de fome e tédio caem.

16. Crie uma "porta escondida" no programa do programa do bichinho virtual que mostre os valores exatos dos atributos do objeto. Consiga isto mostrando o objeto quando uma opção secreta, não listada no menu, for informada na escolha do usuário. Dica: acrescente um método especial str() à classe Bichinho.

17. Crie uma Fazenda de Bichinhos instanciando vários objetos bichinho e mantendo o controle deles através de uma lista. Imite o funcionamento do programa básico, mas ao invés de exigis que o usuário tome conta de um único bichinho, exija que ele tome conta da fazenda inteira. Cada opção do menu deveria permitir que o usuário executasse uma ação para todos os bichinhos (alimentar todos os bichinhos, brincar com todos os bichinhos, ou ouvir a todos os bichinhos). Para tornar o programa mais interessante, dê para cada bichinho um nivel inicial aleatório de fome e tédio.

Solução em Python e Java

1. **Classe Bola:** Crie uma classe que modele uma bola:

   a. Atributos: Cor, circunferência, material
   b. Métodos: trocaCor e mostraCor

Python

```python
class Bola():
    def __init__(self, cor, circunferencia, material):
        self.cor = cor
        self.circunferencia = circunferencia
        self.material = material

    def trocaCor(self, cor):
        self.cor = cor

    def mostraCor(self):
        return self.cor

b = Bola("azul", 10, "couro")
print(b.mostraCor())
b.trocaCor("vermelha")
print(b.mostraCor())
```

Java

```java
public class Bola
{
	private String cor;
	private int circunferencia;
	private String material;

	public Bola(String cor, int circunferencia, String material)
	{
		super();
		this.cor = cor;
		this.circunferencia = circunferencia;
		this.material = material;
	}

	public void trocaCor(String cor) {
		this.cor = cor;
	}

	public String mostraCor() {
		return this.cor;
	}


	public static void main(String[] args)
	{
		Bola b = new Bola("azul", 10, "couro");
		System.out.println(b.mostraCor());
		b.trocaCor("vermelha");
		System.out.println(b.mostraCor());
	}
}
```

2. **Classe Quadrado:** Crie uma classe que modele um quadrado:
   a. Atributos: Tamanho do lado
   b. Métodos: Mudar valor do Lado, Retornar valor do Lado e calcular Área

Python

```python
class Quadrado():
    def __init__(self, lado):
        self.setLado(lado)

    def setLado(self, lado):
        self.lado = lado

    def getLado(self):
        return self.lado

    def area(self):
        return self.lado * self.lado

q = Quadrado(5)
print(q.area())
```

Java

```java
public class Quadrado
{
	private float lado;

	public Quadrado(float lado) {
		setLado(lado);
	}

	public void setLado(float lado) {
		this.lado = lado;
	}

	public float getLado() {
		return lado;
	}

	public float area() {
		return this.lado * this.lado;
	}

	public static void main(String[] args)
	{
		Quadrado q = new Quadrado(5);
		System.out.println(q.area());


	}

}
```

3. **Classe Retangulo:** Crie uma classe que modele um retangulo:

   a. Atributos: LadoA, LadoB (ou Comprimento e Largura, ou Base e Altura, a escolher)
   b. Métodos: Mudar valor dos lados, Retornar valor dos lados, calcular Área e calcular Perímetro;
   c. Crie um programa que utilize esta classe. Ele deve pedir ao usuário que informe as medidades de um local. Depois, deve criar um objeto com as medidas e calcular a quantidade de pisos e de rodapés necessárias para o local.

Python

```python
class Retangulo():
    def __init__(self, comprimento, largura):
        self.setComprimento(comprimento)
        self.setLargura(largura)

    def setComprimento(self, comprimento):
        self.comprimento = comprimento

    def setLargura(self, largura):
        self.largura = largura

    def getComprimento(self):
        return self.comprimento

    def getLargura(self):
        return self.largura

    def area(self):
        return self.comprimento * self.largura

    def perimetro(self):
        return (2 * self.comprimento) + (2 * self.largura)

comp = float(input('Informe o valor do comprimento: '))
larg = float(input('Informe o valor da largura: '))

r = Retangulo(comp,larg)
print("A area é: ", r.area())
print("O perimetro é: ", r.perimetro())
```

Java

```java
import java.util.Scanner;

public class Retangulo {
        private float comprimento;
        private float largura;

        public Retangulo(float comprimento, float largura)     {

                this.comprimento = comprimento;
                this.largura = largura;
        }
        public void setComprimento(float comprimento) {
                this.comprimento = comprimento;
        }
        public void setLargura(float largura) {
                this.largura = largura;
        }
        public float getComprimento() {
                return comprimento;
        }
        public float getLargura() {
                return largura;
        }
        public float area() {
                return getComprimento() * getLargura();
        }

        public float perimetro(){
                return (2 * getComprimento()) + (2 * getLargura());
        }

        public static void main(String[] args){
                float comp, larg;

                Scanner teclado = new Scanner(System.in);
                System.out.println("Informe o valor do comprimento: ");
                comp = teclado.nextFloat();

                System.out.println("Informe o valor da largura: ");
                larg = teclado.nextFloat();

                Retangulo r = new Retangulo(comp,larg);

                System.out.println("A área é: " + r.area());
                System.out.println("O perimetro é: " + r.perimetro());
        }
}
```

4. **Classe Pessoa:** Crie uma classe que modele uma pessoa:

   a. Atributos: nome, idade, peso e altura
   b. Métodos: Envelhercer, engordar, emagrecer, crescer. Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.

Python

```python
class Pessoa():
    def __init__(self, nome, idade, peso, altura):
        self.nome = nome
        self.idade = idade
        self.peso = peso
        self.altura = altura

    def envelhecer(self, anos):
        self.idade += anos
        if (self.idade < 21):
            self.crescer(0.5)

    def engordar(self, peso):
        self.peso += peso

    def emagrecer(self, peso):
        self.peso -= peso

    def crescer(self, altura):
        self.altura += altura

a = Pessoa("Amanda", 18, 75, 180)
print(vars(a))
a.engordar(5)
print(vars(a))
a.emagrecer(10)
print(vars(a))
a.crescer(3)
print(vars(a))
a.envelhecer(1)
print(vars(a))
```

Java

```java
public class Pessoa {
	private String nome;
	private int idade;
	private float peso;
	private float altura;

	public Pessoa(String nome, int idade, float peso, float altura) {
		this.nome = nome;
		this.idade = idade;
		this.peso = peso;
		this.altura = altura;
	}

	public void envelhecer(int anos) {
		this.idade += anos;
		if (this.idade < 21)
			crescer(0.5f);
	}

	public void engordar(float peso) {
		this.peso += peso;
	}

	public void emagrecer(float peso) {
		this.peso -= peso;
	}

	public void crescer(float altura) {
		this.altura += altura;
	}

	@Override
	public String toString() {
		return "Pessoa [nome=" + nome + ", idade=" + idade + ", peso=" + peso + ", altura=" +
altura + "]";
	}

	public static void main(String[] args) {
		Pessoa a = new Pessoa("Amanda", 18, 75, 180);
		System.out.println(a.toString());
		a.engordar(5);
		System.out.println(a.toString());
		a.emagrecer(10);
		System.out.println(a.toString());
		a.crescer(3);
		System.out.println(a.toString());
		a.envelhecer(1);
		System.out.println(a.toString());
	}
}
```

5. Classe Conta Corrente: Crie uma classe para implementar uma conta corrente.
A classe deve possuir os seguintes atributos: número da conta, nome do correntista e saldo.
Os métodos são os seguintes: alterarNome, depósito e saque;
No construtor, saldo é opcional, com valor default zero e os demais atributos são obrigatórios.

Python

```python
class Conta():
    def __init__(self, numero, nome, saldo=0):
        self.numero = numero
        self.nome = nome
        self.saldo = saldo

    def setNome(self, nome):
        self.nome = nome

    def deposito(self, valor):
        self.saldo += valor

    def saque(self, valor):
        if (self.saldo >= valor):
            self.saldo -= valor

j = Conta(123, "José", 100.0)
print(vars(j))
j.setNome("Pedro")
j.deposito(50)
print(vars(j))
j.saque(110)
print(vars(j))
```

Java

```java
public class Conta
{
	private int numero;
	private String nome;
	private float saldo;

	public Conta(int numero, String nome, float saldo)
	{
		this.numero = numero;
		this.nome = nome;
		this.saldo = saldo;
	}

	public Conta(int numero, String nome)
	{
		this.numero = numero;
		this.nome = nome;
		this.saldo = 0.0f;
	}

	public void setNome(String nome) {
		this.nome = nome;
	}

	public void deposito(float valor) {
		this.saldo += valor;
	}

	public void saque(float valor) {
		if (this.saldo >= valor) {
			this.saldo -= valor;
		}
	}


	@Override
	public String toString() {
		return "Conta [numero=" + numero + ", nome=" + nome + ", saldo=" + saldo + "]";
	}

	public static void main(String[] args) {
		Conta j = new Conta(123, "José", 100.0f);
		System.out.println(j.toString());
		j.setNome("Pedro");
		j.deposito(50);
		System.out.println(j.toString());
		j.saque(110);
		System.out.println(j.toString());
	}
}
```

6. Classe TV: Faça um programa que simule um televisor criando-o como um objeto.
O usuário deve ser capaz de informar o número do canal e aumentar ou diminuir o volume.
Certifique-se de que o número do canal e o nível do volume permanecem dentro de faixas válidas.

Python

```python
class Tv():
    def __init__(self):
        self.setCanal(0)
        self.volume = 0

    def setCanal(self, canal):
        if (canal > 0) and (canal <= 100):
            self.canal = canal

    def aumentarVolume(self):
        if (self.volume < 100):
            self.volume += 1

    def diminuirVolume(self):
        if (self.volume > 0):
            self.volume -= 1

tv = Tv()
print(vars(tv))
tv.setCanal(56)
print(vars(tv))
tv.aumentarVolume()
print(vars(tv))
tv.diminuirVolume()
print(vars(tv))
```

Java

```java
public class Tv
{
	private int volume;
	private int canal;

	public Tv() {
		this.volume = 0;
		setCanal(56);
	}

	@Override
	public String toString() {
		return "Tv [volume=" + volume + ", canal=" + canal + "]";
	}

	public void setCanal(int canal) {
		if ((canal > 0 ) && (canal <= 100))
		this.canal = canal;

	}

	public void aumentarVolume() {
		if (volume < 100)
		this.volume++;
	}

	public void diminuirVolume() {
		if (volume > 0)
			this.volume--;
	}

	public static void main(String[] args) {
		Tv tv = new Tv();
		System.out.println(tv.toString());
		tv.setCanal(56);
		System.out.println(tv.toString());
		tv.aumentarVolume();
		System.out.println(tv.toString());
		tv.diminuirVolume();
		System.out.println(tv.toString());
	}
}
```

7. Classe Bichinho Virtual: Crie uma classe que modele um Tamagushi (Bichinho Eletrônico):

a. Atributos: Nome, Fome, Saúde e Idade
b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade
c. Obs: Existe mais uma informação que devemos levar em consideração, o Humor do nosso tamagushi, este humor é uma combinação entre os atributos Fome e Saúde, ou seja, um campo calculado, então não devemos criar um atributo para armazenar esta informação por que ela pode ser calculada a qualquer momento.

Python

```python
class Bichinho():
    def __init__(self, nome, fome, saude, idade):
        self.setNome(nome)
        self.setFome(fome)
        self.setSaude(saude)
        self.setIdade(idade)

    def setNome(self, nome):
        self.nome = nome

    def setFome(self, fome):
        self.fome = fome

    def setSaude(self, saude):
        self.saude = saude

    def setIdade(self, idade):
        self.idade = idade

    def getNome(self):
        return self.nome

    def getFome(self):
        return self.fome

    def getSaude(self):
        return self.saude

    def getIdade(self, idade):
        return self.idade

    def humor(self):
        return self.getFome() * self.getSaude()

b = Bichinho("Tamagoshi", 5,5,5)
print(b.humor())
```

Java

```java
public class Bichinho {
	private String nome;
	private int fome;
	private int saude;
	private int idade;
	public Bichinho(String nome, int fome, int saude, int idade) {
		this.nome = nome;
		this.fome = fome;
		this.saude = saude;
		this.idade = idade;
	}
	public String getNome() {
		return nome;
	}
	public void setNome(String nome) {
		this.nome = nome;
	}
	public int getFome() {
		return fome;
	}
	public void setFome(int fome) {
		this.fome = fome;
	}
	public int getSaude() {
		return saude;
	}
	public void setSaude(int saude) {
		this.saude = saude;
	}
	public int getIdade() {
		return idade;
	}
	public void setIdade(int idade) {
		this.idade = idade;
	}
	public float humor() {
		return getFome() * getSaude();
	}
	public static void main(String[] args) {
		Bichinho b = new Bichinho("Tamagoshi", 5, 5, 5);
		System.out.println(b.humor());
	}
}
```

8. Classe Macaco: Desenvolva uma classe Macaco, que possua os atributos nome e bucho (estomago) e pelo menos os métodos comer(), verBucho() e digerir().
Faça um programa ou teste interativamente, criando pelo menos dois macacos, alimentando-os com pelo menos 3 alimentos diferentes e verificando o conteúdo do estomago a cada refeição.
Experimente fazer com que um macaco coma o outro. É possível criar um macaco canibal?

Python

```python
class Macaco():
    def __init__(self, nome):
        self.nome = nome
        self.bucho = []

    def comer(self, comida):
        self.bucho.append(comida)

    def verBucho(self):
        print ("Bucho: " , self.bucho)

    def digerir(self):
        if (len(self.bucho) > 0):
            self.bucho.pop(0)

m1 = Macaco("Macaco 1")
m2 = Macaco("Macaco 2")

m1.comer("Maçã")
m1.verBucho()

m1.comer("Banana")
m1.verBucho()

m1.comer("Abacaxi")
m1.verBucho()
m1.digerir()
m1.verBucho()

m2.comer("Maca")
m2.comer("Banana")
m2.comer(m1)
m2.verBucho()
```

Java

```java
import java.util.ArrayList;

public class Macaco
{
	private String nome;
	private ArrayList<Object> bucho = new ArrayList();

	public Macaco(String nome) {
		this.nome = nome;
	}
	public void comer(Object ob) {
		bucho.add(ob);
	}
	public void verBucho() {
		for(Object i : bucho)
		 System.out.println("Bucho: " + i.toString());

	}
	public void digerir() {
		if (!bucho.isEmpty())
			bucho.remove(bucho.size()-1);
	}
	public static void main(String[] args) {
		Macaco m1 = new Macaco("Macaco 1");
		Macaco m2 = new Macaco("Macaco 1");
		m1.comer("Maçã");
		m1.verBucho();
		m1.comer("Banana");
		m1.verBucho();
		m1.comer("Abacaxi");
		m1.verBucho();
		m1.digerir();
		m1.verBucho();
		m2.comer("Maça");
		m2.comer("Banana");
		m2.comer(m1);
		m2.verBucho();
	}
}
```

9. Classe Ponto e Retangulo: Faça um programa completo utilizando funções e classes que:

a. Possua uma classe chamada Ponto, com os atributos x e y.
b. Possua uma classe chamada Retangulo, com os atributos largura e altura.
c. Possua uma função para imprimir os valores da classe Ponto
d. Possua uma função para encontrar o centro de um Retângulo.
e. Você deve criar alguns objetos da classe Retangulo.
f. Cada objeto deve ter um vértice de partida, por exemplo, o vértice inferior esquerdo do retângulo, que deve ser um objeto da classe Ponto.
g. A função para encontrar o centro do retângulo deve retornar o valor para um objeto do tipo ponto que indique os valores de x e y para o centro do objeto.
h. O valor do centro do objeto deve ser mostrado na tela
i. Crie um menu para alterar os valores do retângulo e imprimir o centro deste retângulo.

Python

```
class Ponto:
   def __init__(self, x, y):
      self.x = x
      self.y = y

class Retangulo:
   def __init__(self, canto1, canto2):
      self.canto1 = canto1
      self.canto2 = canto2
   def centro(self):
      x_centro = (self.canto1.x + self.canto2.x) / 2.0
      y_centro = (self.canto1.y + self.canto2.y) / 2.0
      return "X=" + str(x_centro) + "Y=" + str(y_centro)

x1 = float(input("Entre a coordenada x do canto inferior esquerdo: "))
y1 = float(input("Entre a coordenada y do canto inferior esquerdo: "))
canto1 = Ponto(x1, y1)
x2 = float(input("Entre a coordenada x do canto superior direito: "))
y2 = float(input("Entre a coordenada y do canto superior direito: "))
canto2 = Ponto(x2, y2)
ret = Retangulo(canto1, canto2)
print ("Ponto central e %s" % ret.centro())
```

Java

```
class Ponto{
		private float x;
		private float y;

		public Ponto(float x, float y) {
			this.x = x;
			this.y = y;
		}

		public float getX()
		{
			return x;
		}

		public float getY()
		{
			return y;
		}
}

public class Retangulo1
{
		Ponto c1, c2;
		float x_centro, y_centro;

		public Retangulo1(Ponto canto1, Ponto canto2) {
			c1 = canto1;
			c2 = canto2;
		}

		public String centro() {
			x_centro = (c1.getX() + c2.getX())/2;
			y_centro = (c1.getY() + c2.getY())/2;
			return "X = " + x_centro + "Y =" + y_centro;
		}


		@Override
		public String toString() {
			return "Retangulo1 [c1=" + c1 + ", c2=" + c2 + ", x_centro=" + x_centro + ", y_centro=" + y_centro + "]";
		}

		public static void main(String[] args)
		{
			Ponto p1 = new Ponto(10,20);
			Ponto p2 = new Ponto(20,50);
			Retangulo1 r = new Retangulo1(p1,p2);
			System.out.println(r.centro());
		}
}
```

10. Classe Bomba de Combustível: Faça um programa completo utilizando classes e métodos que:

a. Possua uma classe chamada bombaCombustível, com no mínimo esses atributos:
   i. tipoCombustivel.
   ii. valorLitro
   iii. quantidadeCombustivel
b. Possua no mínimo esses métodos:
   i. abastecerPorValor( ) – método onde é informado o valor a ser abastecido e mostra a quantidade de litros que foi colocada no veículo
   ii. abastecerPorLitro( ) – método onde é informado a quantidade em litros de combustível e mostra o valor a ser pago pelo cliente.
   iii. alterarValor( ) – altera o valor do litro do combustível.
   iv. alterarCombustivel( ) – altera o tipo do combustível.
   v. alterarQuantidadeCombustivel( ) – altera a quantidade de combustível restante na bomba.

OBS: Sempre que acontecer um abastecimento é necessário atualizar a quantidade de combustível total na bomba

Python

```python
class BombaCombustivel:
    def __init__(self, tipoCombustivel, valorLitro, quantidadeCombustivel):
        self.tipoCombustivel = tipoCombustivel
        self.valorLitro = valorLitro
        self.quantidadeCombustivel = quantidadeCombustivel

    def alterarValor(self, valorLitro):
        self.valorLitro = valorLitro

    def alterarCombustivel(self, tipoCombustivel):
        self.tipoCombustivel = tipoCombustivel

    def alterarQuantidadeCombustivel(self, quantidadeCombustivel):
        self.quantidadeCombustivel = quantidadeCombustivel

    def abastecerPorValor(self, valor):
        temp = valor/self.valorLitro
        self.alterarQuantidadeCombustivel(self.quantidadeCombustivel - temp)
        return temp

    def abastecerPorLitro(self, qtd):
        temp2 = qtd * self.valorLitro
        self.alterarQuantidadeCombustivel(self.quantidadeCombustivel - qtd)
        return temp2


a1 = BombaCombustivel("Gasolina", 5, 500)
print(a1.abastecerPorValor(150))
print(a1.quantidadeCombustivel)
print(a1.abastecerPorLitro(30))
print(a1.quantidadeCombustivel)
```

Java

```java
public class BombaCombustivel
{
	private String tipoCombustivel;
	private float valorLitro;
	private float quantidadeCombustivel;

	public BombaCombustivel(String tipoCombustivel, float valorLitro, float quantidadeCombustivel) {
		this.tipoCombustivel = tipoCombustivel;
		this.valorLitro = valorLitro;
		this.quantidadeCombustivel = quantidadeCombustivel;
	}

	public void alterarValor(float valorLitro) {
		this.valorLitro = valorLitro;
	}

	public void alterarCombustivel(String tipoCombustivel) {
		this.tipoCombustivel = tipoCombustivel;
	}

	public void alterarQuantidadeCombustivel(float quantidadeCombustivel)
	{
		this.quantidadeCombustivel = quantidadeCombustivel;
	}
	public float abastecerPorValor(float valor) {
		float temp;
		temp = valor/valorLitro;
		alterarQuantidadeCombustivel(this.quantidadeCombustivel - temp);
		return temp;
	}
	public float abastecerPorLitro(float qtd) {
		float temp2;
		temp2 = qtd * valorLitro;
		alterarQuantidadeCombustivel(this.quantidadeCombustivel - qtd);
		return temp2;
	}

	public static void main(String[] args)
	{
		BombaCombustivel a1 = new BombaCombustivel("Gasolina", 5, 500);
		System.out.println(a1.abastecerPorValor(150));
		System.out.println(a1.quantidadeCombustivel);
		System.out.println(a1.abastecerPorLitro(30));
		System.out.println(a1.quantidadeCombustivel);
	}
}
```

11. Classe carro: Implemente uma classe chamada Carro com as seguintes propriedades:

a. Um veículo tem um certo consumo de combustível (medidos em km / litro) e uma certa quantidade de combustível no tanque.
b. O consumo é especificado no construtor e o nível de combustível inicial é 0.
c. Forneça um método andar( ) que simule o ato de dirigir o veículo por uma certa distância, reduzindo o nível de combustível no tanque de gasolina.
d. Forneça um método obterGasolina( ), que retorna o nível atual de combustível.
e. Forneça um método adicionarGasolina( ), para abastecer o tanque. Exemplo de uso:

```
f. meuFusca = Carro(15);            # 15 quilômetros por litro de combustível.
g. meuFusca.adicionarGasolina(20); # abastece com 20 litros de combustível.
h. meuFusca.andar(100);             # anda 100 quilômetros.
   meuFusca.obterGasolina()         # Imprime o combustível que resta no tanque.
```

Python

```
class Carro:
  def __init__(self, consumo):
    self.consumo = consumo
    self.nivelCombustivel = 0

  def andar(self, distancia):
    temp = distancia/self.consumo
    self.nivelCombustivel -= temp

  def obterGasolina(self):
    return self.nivelCombustivel

  def adicionarGasolina(self, qtd):
    self.nivelCombustivel += qtd

meuFusca = Carro(8)
meuFusca.adicionarGasolina(50)
meuFusca.andar(300)
print(vars(meuFusca))
print(meuFusca.obterGasolina())
```

Java

```
public class Carro {
	private float consumo;
	private float nivelCombustivel;

	public Carro(float consumo) {
		this.consumo = consumo;
	}

	public void andar(float distancia) {
		float temp;
		temp = distancia/this.consumo;
		this.nivelCombustivel -= temp;
	}

	public float obterGasolina() {
		return this.nivelCombustivel;
	}

	public void adicionarGasolina(float qtd) {
		this.nivelCombustivel += qtd;
	}



	@Override
	public String toString() {
		return "Carro [consumo=" + consumo + ", nivelCombustivel=" + nivelCombustivel + "]";
	}

	public static void main(String[] args)
	{
		Carro meuFusca = new Carro(8);
		meuFusca.adicionarGasolina(50);
		System.out.println(meuFusca.toString());
		meuFusca.andar(300);
		System.out.println(meuFusca.toString());
	}

}
```

12. Classe Conta de Investimento: Faça uma classe contaInvestimento que seja semelhante a classe contaBancaria, com a diferença de que se adicione um atributo taxaJuros. Forneça um construtor que configure tanto o saldo inicial como a taxa de juros. Forneça um método adicioneJuros (sem parâmetro explícito) que adicione juros à conta. Escreva um programa que construa uma poupança com um saldo inicial de R$1000,00 e uma taxa de juros de 10%. Depois aplique o método adicioneJuros() cinco vezes e imprime o saldo resultante.

Python

```
class Conta():
   def __init__(self, numero, nome, saldo=0):
      self.numero = numero
      self.nome = nome
      self.saldo = saldo

   def setNome(self, nome):
      self.nome = nome

   def deposito(self, valor):
      self.saldo += valor

   def saque(self, valor):
      if (self.saldo >= valor):
         self.saldo -= valor

class contaInvestimento(Conta):
   def __init__(self, numero, nome, saldo, taxaJuros):
      Conta.__init__(self, numero, nome, saldo)
      self.taxaJuros = taxaJuros

   def adicioneJuros(self):
      self.saldo += (self.saldo * self.taxaJuros/100)

poupanca = contaInvestimento(123, "Jose", 1000, 10)
print(vars(poupanca))
poupanca.adicioneJuros()
poupanca.adicioneJuros()
poupanca.adicioneJuros()
poupanca.adicioneJuros()
poupanca.adicioneJuros()
print(vars(poupanca))
```

Java

```
public class Conta
{
        private int numero;
        private String nome;
        protected float saldo;

        public Conta(int numero, String nome, float saldo)
        {
                this.numero = numero;
                this.nome = nome;
                this.saldo = saldo;
        }

        public Conta(int numero, String nome)
        {
                this.numero = numero;
                this.nome = nome;
                this.saldo = 0.0f;
        }

        public void setNome(String nome) {
                this.nome = nome;
        }

        public void deposito(float valor) {
                this.saldo += valor;
        }

        public void saque(float valor) {
                if (this.saldo >= valor) {
                        this.saldo -= valor;
                }
        }


        @Override
        public String toString() {
                return "Conta [numero=" + numero + ", nome=" + nome + ", saldo=" + saldo + "]";
        }
```

```java
public class ContaInvestimento extends Conta
{
	private float taxaJuros;

	public ContaInvestimento(int numero, String nome, float saldo, float taxaJuros)
	{
		super(numero, nome, saldo);
		this.taxaJuros = taxaJuros;
	}

	public void adicioneJuros() {
		this.saldo += (this.saldo * this.taxaJuros/100);
	}


	@Override
	public String toString() {
		return "ContaInvestimento [taxaJuros=" + taxaJuros + ", saldo=" + saldo + "]";
	}

	public static void main(String[] args)
	{
		ContaInvestimento poupanca = new ContaInvestimento(123, "Jose", 1000, 10);
		System.out.println(poupanca.toString());
		poupanca.adicioneJuros();
		poupanca.adicioneJuros();
		poupanca.adicioneJuros();
		poupanca.adicioneJuros();
		poupanca.adicioneJuros();
		System.out.println(poupanca.toString());


	}

}
```

13. Classe Funcionário: Implemente a classe Funcionário. Um empregado tem um nome (um string) e um salário(um double). Escreva um construtor com dois parâmetros (nome e salário) e métodos para devolver nome e salário. Escreva um pequeno programa que teste sua classe.

Python

```
class Funcionario():
   def __init__(self, nome, salario):
      self.nome = nome
      self.salario = salario

   def getNome(self):
      return self.nome

   def getSalario(self):
      return self.salario

Func = Funcionario("Jose", 1200)
print("Nome: ", Func.getNome(), ", Salario", Func.getSalario())
```

Java

```
public class Funcionario {
        private String nome;
        private float salario;

        public Funcionario(String nome, float salario)
        {
                this.nome = nome;
                this.salario = salario;
        }

        public String getNome() {
                return nome;
        }

        public float getSalario() {
                return salario;
        }

        public static void main(String[] args)
        {
                Funcionario Func = new Funcionario("Jose", 1200);
                System.out.println("Nome: " + Func.getNome() + ", Salário: " +
Func.getSalario());
        }

}
```

14. Aprimore a classe do exercício anterior para adicionar o método aumentarSalario (porcentualDeAumento) que aumente o salário do funcionário em uma certa porcentagem.

- Exemplo de uso:
- ```
  harry=funcionário("Harry",25000)
  harry.aumentarSalario(10)
  ```

Python

```
class Funcionario():
    def __init__(self, nome, salario):
        self.nome = nome
        self.salario = salario

    def getNome(self):
        return self.nome

    def getSalario(self):
        return self.salario

    def aumentarSalario(self, porcentagemDeAumento=0):
        self.salario += self.salario * (porcentagemDeAumento)/100

Func = Funcionario("Jose", 1200)
print("Nome: ", Func.getNome(), ", Salario", Func.getSalario())
Func.aumentarSalario(10)
print("Nome: ", Func.getNome(), ", Salario", Func.getSalario())
```

Java

```
public class Funcionario {
	private String nome;
	private float salario;

	public Funcionario(String nome, float salario)
	{
		this.nome = nome;
		this.salario = salario;
	}

	public String getNome() {
		return nome;
	}

	public float getSalario() {
		return salario;
	}

	public void aumentarSalario(float porcentagemDeAumento) {
		this.salario += this.salario * (porcentagemDeAumento)/100;

	}

	public static void main(String[] args)
	{
		Funcionario Func = new Funcionario("Jose", 1200);
		System.out.println("Nome: " + Func.getNome() + ", Salário: " +
Func.getSalario());
		Func.aumentarSalario(10);
		System.out.println("Nome: " + Func.getNome() + ", Salário: " +
Func.getSalario());
	}

}
```
```

15. Classe Bichinho Virtual++: Melhore o programa do bichinho virtual, permitindo que o usuário especifique quanto de comida ele fornece ao bichinho e por quanto tempo ele brinca com o bichinho. Faça com que estes valores afetem quão rapidamente os níveis de fome e tédio caem.

Python

```python
class Bichinho():
    def __init__(self, nome, fome, saude, idade):
        self.setNome(nome)
        self.setFome(fome)
        self.setSaude(saude)
        self.setIdade(idade)

    def setNome(self, nome):
        self.nome = nome

    def setFome(self, fome):
        self.fome = fome

    def setSaude(self, saude):
        self.saude = saude

    def setIdade(self, idade):
        self.idade = idade

    def getNome(self):
        return self.nome

    def getFome(self):
        return self.fome

    def getSaude(self):
        return self.saude

    def getIdade(self, idade):
        return self.idade

    def humor(self):
        return self.getFome() * self.getSaude()

    def alimenta(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.fome -= self.fome * (quantidade /100.0)

    def brincar(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.saude += self.saude * (quantidade / 100.0)

b = Bichinho("Tamagoshi", 5,5,5)
print(b.humor())
b.alimenta(30)
print(b.humor())
b.brincar(20)
print(b.humor())
```

Java

```java
public class Bichinho {
        private String nome;
        private float fome;
        private float saude;
        private float idade;

        public Bichinho(String nome, float fome, float saude, float idade) {
                this.nome = nome;
                this.fome = fome;
                this.saude = saude;
                this.idade = idade;
        }
        public String getNome() {
                return nome;
        }
        public void setNome(String nome) {
                this.nome = nome;
        }
        public float getFome() {
                return fome;
        }
        public void setFome(float fome) {
                this.fome = fome;
        }
        public float getSaude() {
                return saude;
        }
        public void setSaude(float saude) {
                this.saude = saude;
        }
        public float getIdade() {
                return idade;
        }
        public void setIdade(float idade) {
                this.idade = idade;
        }
        public float humor() {
                return getFome() * getSaude();
        }

        public void alimenta(float quantidade) {
                if ((quantidade >= 0) && (quantidade <= 100)) {
                        this.fome = this.fome - (this.fome * (quantidade/100.0f));
                }
        }
        public void brincar(int quantidade) {
                if ((quantidade >= 0) && (quantidade <= 100)) {
                        this.saude += this.saude * (quantidade/100.0f);
                }
        }
        public static void main(String[] args) {
                Bichinho b = new Bichinho("Tamagoshi", 5, 5, 5);
                System.out.println(b.humor());
                b.alimenta(30);
                System.out.println(b.humor());
                b.brincar(20);
                System.out.println(b.humor());
        }
}
```

16. Crie uma "porta escondida" no programa do programa do bichinho virtual que mostre os valores exatos dos atributos do objeto. Consiga isto mostrando o objeto quando uma opção secreta, não listada no menu, for informada na escolha do usuário. Dica: acrescente um método especial str() à classe Bichinho.

Python

```
class Bichinho():
    def __init__(self, nome, fome, saude, idade):
        self.setNome(nome)
        self.setFome(fome)
        self.setSaude(saude)
        self.setIdade(idade)

    def setNome(self, nome):
        self.nome = nome

    def setFome(self, fome):
        self.fome = fome

    def setSaude(self, saude):
        self.saude = saude

    def setIdade(self, idade):
        self.idade = idade

    def getNome(self):
        return self.nome

    def getFome(self):
        return self.fome

    def getSaude(self):
        return self.saude

    def getIdade(self):
        return self.idade

    def humor(self):
        return self.getFome() * self.getSaude()

    def alimenta(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.fome -= self.fome * (quantidade /100.0)

    def brincar(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.saude += self.saude * (quantidade / 100.0)

    def str(self):
        return ("Nome: "  + str(self.getNome()) + ", Fome: " + str(self.getFome()) + ", Saude: " + str(self.getSaude()) + ", Idade:
" + str(self.getIdade()))


b = Bichinho("Tamagoshi", 5,5,5)
print(b.humor())
b.alimenta(30)
print(b.humor())
b.brincar(20)
print(b.humor())
print(b.str())
```

Java

```
public class Bichinho {
            private String nome;
            private float fome;
            private float saude;
            private float idade;

            public Bichinho(String nome, float fome, float saude, float idade) {
                        this.nome = nome;
                        this.fome = fome;
                        this.saude = saude;
                        this.idade = idade;
            }


            public String getNome() {
                        return nome;
            }
            public void setNome(String nome) {
                        this.nome = nome;
            }
            public float getFome() {
                        return fome;
            }
            public void setFome(float fome) {
                        this.fome = fome;
            }
            public float getSaude() {
                        return saude;
            }
            public void setSaude(float saude) {
                        this.saude = saude;
            }
            public float getIdade() {
                        return idade;
            }
            public void setIdade(float idade) {
                        this.idade = idade;
            }
            public float humor() {
                        return getFome() * getSaude();
            }

            public void alimenta(float quantidade) {
                        if ((quantidade >= 0) && (quantidade <= 100)) {
                                    this.fome = this.fome - (this.fome * (quantidade/100.0f));
                        }
            }

            public void brincar(int quantidade) {
                        if ((quantidade >= 0) && (quantidade <= 100)) {
                                    this.saude += this.saude * (quantidade/100.0f);
                        }
            }

            public String str() {
                        return "Bichinho [nome=" + nome + ", fome=" + fome + ", saude=" + saude + ", idade=" + idade + "]";
            }


            public static void main(String[] args) {
                        Bichinho b = new Bichinho("Tamagoshi", 5, 5, 5);
                        System.out.println(b.humor());
                        b.alimenta(30);
                        System.out.println(b.humor());
                        b.brincar(20);
                        System.out.println(b.humor());
                        System.out.println(b.str());



            }

}
```

17. Crie uma Fazenda de Bichinhos instanciando vários objetos bichinho e mantendo o controle deles através de uma lista. Imite o funcionamento do programa básico, mas ao invés de exigis que o usuário tome conta de um único bichinho, exija que ele tome conta da fazenda inteira. Cada opção do menu deveria permitir que o usuário executasse uma ação para todos os bichinhos (alimentar todos os bichinhos, brincar com todos os bichinhos, ou ouvir a todos os bichinhos). Para tornar o programa mais interessante, dê para cada bichinho um nivel inicial aleatório de fome e tédio.

Python

```python
from random import randint

class Bichinho():
    def __init__(self, nome, fome, saude, idade):
        self.setNome(nome)
        self.setFome(fome)
        self.setSaude(saude)
        self.setIdade(idade)


    def setNome(self, nome):
        self.nome = nome

    def setFome(self, fome):
        self.fome = fome

    def setSaude(self, saude):
        self.saude = saude

    def setIdade(self, idade):
        self.idade = idade

    def getNome(self):
        return self.nome

    def getFome(self):
        return self.fome

    def getSaude(self):
        return self.saude

    def getIdade(self):
        return self.idade

    def humor(self):
        return self.getFome() * self.getSaude()

    def alimenta(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.fome -= self.fome * (quantidade /100.0)

    def brincar(self, quantidade):
        if (quantidade >= 0) and (quantidade <= 100):
            self.saude += self.saude * (quantidade / 100.0)
```

Java

```java
import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

public class Bichinho {
        private String nome;
        private float fome;
        private float saude;
        private float idade;

        public Bichinho(String nome, float fome, float saude, float idade) {
                this.nome = nome;
                this.fome = fome;
                this.saude = saude;
                this.idade = idade;
        }


        public String getNome() {
                return nome;
        }
        public void setNome(String nome) {
                this.nome = nome;
        }
        public float getFome() {
                return fome;
        }
        public void setFome(float fome) {
                this.fome = fome;
        }
        public float getSaude() {
                return saude;
        }
        public void setSaude(float saude) {
                this.saude = saude;
        }
        public float getIdade() {
                return idade;
        }
        public void setIdade(float idade) {
                this.idade = idade;
        }
        public float humor() {
                return getFome() * getSaude();
        }

        public void alimenta(float quantidade) {
```

```
    def str(self):
        return ("Nome: "  + str(self.getNome()) + ", Fome: " + str(self.getFome()) + ", Saude: " +
str(self.getSaude()) + ", Idade: " + str(self.getIdade()))

a = Bichinho("Cachorro", randint(0,10),randint(0,10),5)
b = Bichinho("Gato", randint(0,10),randint(0,10),5)
c = Bichinho("Coelho", randint(0,10),randint(0,10),5)
fazenda = []
fazenda.append(a)
fazenda.append(b)
fazenda.append(c)

while True:
    print(":::: FAZENDA ::::")
    print("1. Alimentar todos os bichos")
    print("2. Brincar com todos os bichos")
    print("3. Ouvir todos os bichos")
    print("4. Sair")
    op = int(input())

    if (op == 1):
        alimento = int(input("Alimentar todos com: "))
        for i in range(3):
            fazenda[i].alimenta(alimento)
    elif(op ==2):
        brinquedo = int(input("Brincar todos com: "))
        for i in range(3):
            fazenda[i].brincar(brinquedo)
    elif(op == 3):
        for i in range(3):
            print(fazenda[i].getNome() + ": " + str(fazenda[i].humor()))
    elif(op == 4):
        break
```

```
                if ((quantidade >= 0) && (quantidade <= 100)) {
                        this.fome = this.fome - (this.fome * (quantidade/100.0f));
                }
        }

        public void brincar(int quantidade) {
                if ((quantidade >= 0) && (quantidade <= 100)) {
                        this.saude += this.saude * (quantidade/100.0f);
                }
        }

        public String str() {
                return "Bichinho [nome=" + nome + ", fome=" + fome + ", saude=" + saude +
", idade=" + idade + "]";
        }


        public static void main(String[] args)
        {
                Random aleatorio = new Random();


                Bichinho a = new Bichinho("Cachorro", aleatorio.nextInt(10),
aleatorio.nextInt(10), 5);
                Bichinho b = new Bichinho("Gato", aleatorio.nextInt(10),
aleatorio.nextInt(10), 5);
                Bichinho c = new Bichinho("Coelho", aleatorio.nextInt(10),
aleatorio.nextInt(10), 5);

                ArrayList<Bichinho> Fazenda = new ArrayList();
                Fazenda.add(a);
                Fazenda.add(b);
                Fazenda.add(c);

                Scanner teclado = new Scanner(System.in);
                int op;
                int alimento, brinquedo;

                while(true)
                {
                        System.out.println(":::: FAZENDA ::::");
                        System.out.println("1. Alimentar todos os bichos");
                        System.out.println("2. Brincar com todos os bichos");

                        System.out.println("3. Ouvir todos os bichos");
                        System.out.println("4. Sair");
                        op = teclado.nextInt();

                        if (op == 1)
                        {
                                System.out.println("Alimentar todos com: ");
                                alimento = teclado.nextInt();
                                for(int i = 0; i<=2; i++)
                                        Fazenda.get(i).alimenta(alimento);
                        }
```

```
                        else if(op ==2)
                        {
                                System.out.println("Brincar com todos com: ");
                                brinquedo = teclado.nextInt();
                                for(int i = 0; i<=2; i++)
                                        Fazenda.get(i).brincar(brinquedo);

                        }else if(op == 3)
                        {
                                for(int i = 0; i<=2; i++)
                                {
                                        System.out.println(Fazenda.get(i).str());
                                        System.out.println(Fazenda.get(i).getNome() + ": "
+ Fazenda.get(i).humor());
                                }

                        }else if (op == 4)
                                break;
                }

        }
}
```