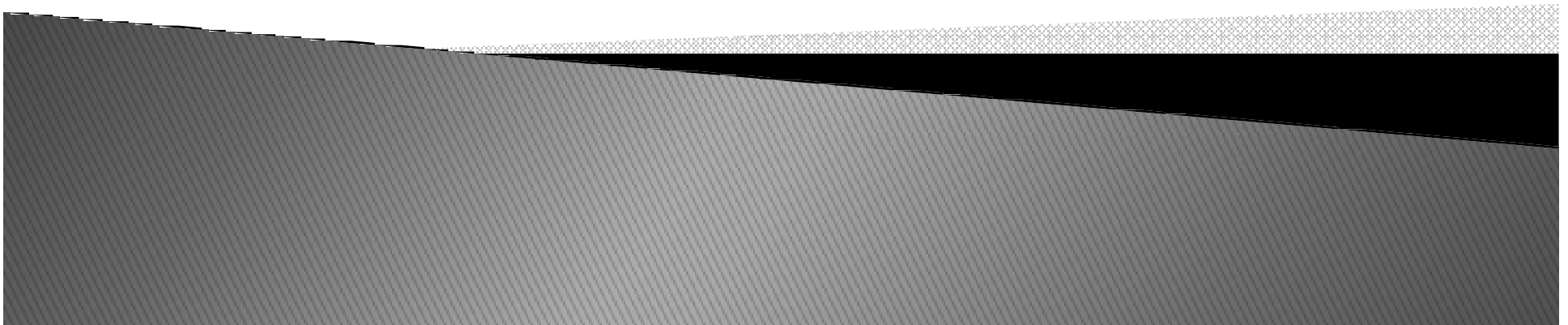


# Orientação a Objetos

Aula06

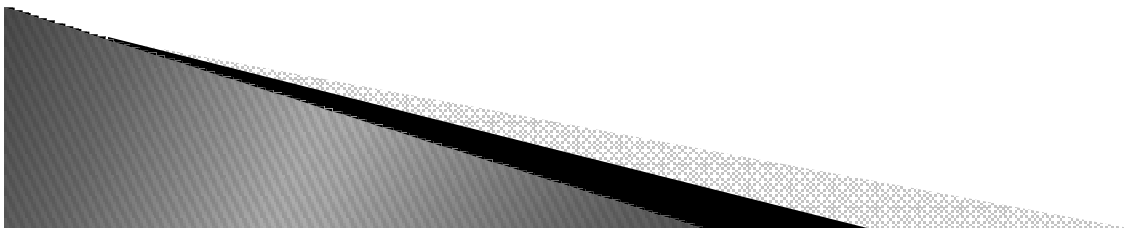
BSI - UFRPE

Prof. Gustavo Callou  
gcallou@gmail.com



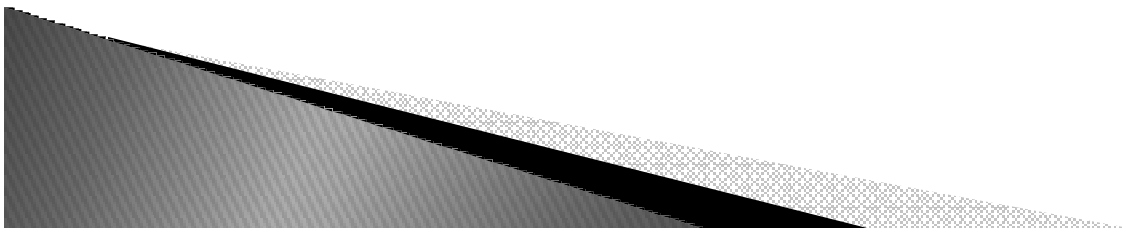
# Classe Abstrata

- ▶ O que é?
- ▶ Para que serve?



# Classe Abstrata

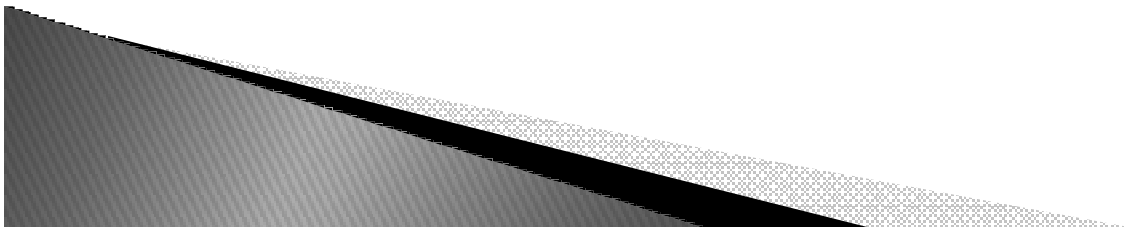
- ▶ O que é?
- ▶ Para que serve?
- ▶ Quando desejamos especificar quais métodos uma classe vai ter sem termos de implementar os métodos, declaramos esses métodos como abstratos e conseqüentemente a classe também.



# Classe Abstrata

## ▶ Exemplo:

```
from abc import ABCMeta, abstractmethod, abstractproperty
class Foo:
    __metaclass__ = ABCMeta # In Python 3, you use the syntax
    # class Foo(metaclass=ABCMeta)
    @abstractmethod
    def spam(self, a, b):
        pass
    @abstractproperty
    def name(self):
        pass
```

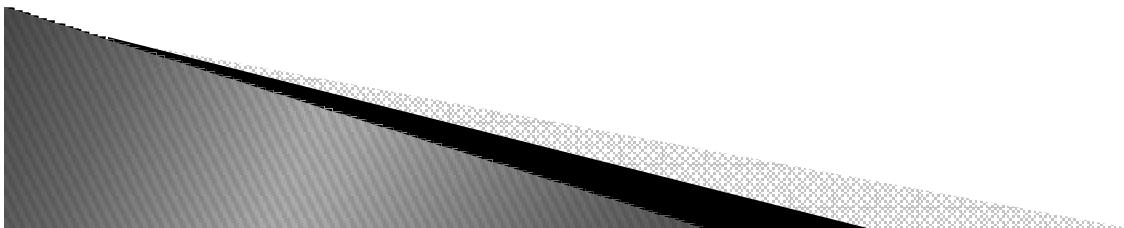


# Classe Abstrata

- ▶ Característica:
  - Não pode ser instanciada.

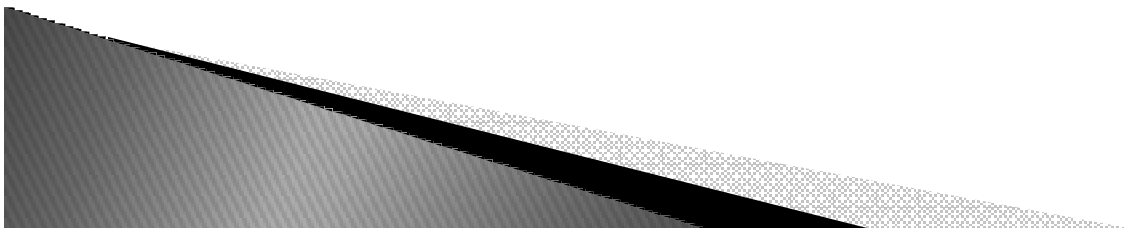
```
>>> f = Foo()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't instantiate abstract class Foo with abstract methods spam
>>>
```

- A subclasse que vai ter de implementar todos os métodos definidos como abstratos da classe abstrata.



# String (revisão)

- ▶ `str.format()`
- ▶ `>>> "The sum of 1 + 2 is {0}".format(1+2)`
- ▶ `'The sum of 1 + 2 is 3'`
- ▶ `>>> "The sum of 1 + 2 is {0}{1}".format(1+2,5)`
- ▶ `'The sum of 1 + 2 is 35'`
- ▶ `>>> "The sum {1}of 1 + 2 is {0}{1}".format(1+2,5)`
- ▶ `'The sum 5of 1 + 2 is 35'`

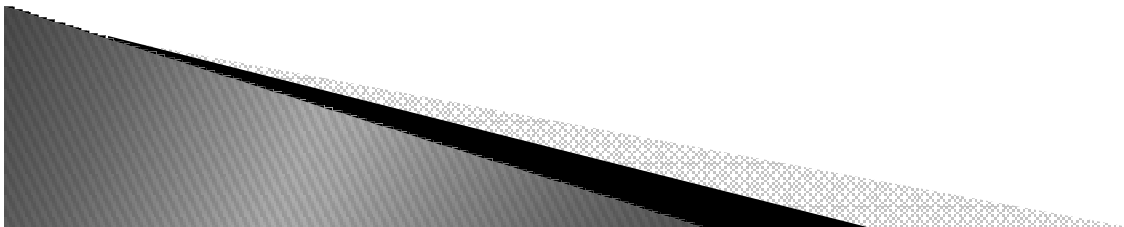


# Classe Abstrata

▶ Exemplo

```
from abc import *
```

```
class SchoolMember(metaclass=ABCMeta):  
    """Representa um membro qualquer da escola."""  
    def __init__(self, name, age):  
        self.name = name  
        self.age = age  
        print('(Iniciado SchoolMember: {0})'.format(self.name))  
  
    @abstractmethod  
    def tell(self):  
        """Imprime os dados da instância."""  
        print('Nome:"{0}" Idade:"{1}"'.format(self.name, self.age),  
              end=" ")
```



# Classe Abstrata

- ▶ Exemplo

```
class Teacher(SchoolMember):  
    """Representa um professor."""  
    def __init__(self, name, age, salary):  
        SchoolMember.__init__(self, name, age)  
        self.salary = salary  
        print('(Iniciado Teacher: {0})'.format(self.name))  
  
    def tell(self):  
        SchoolMember.tell(self)  
        print('Salário: "{0:.2f}"'.format(self.salary))
```



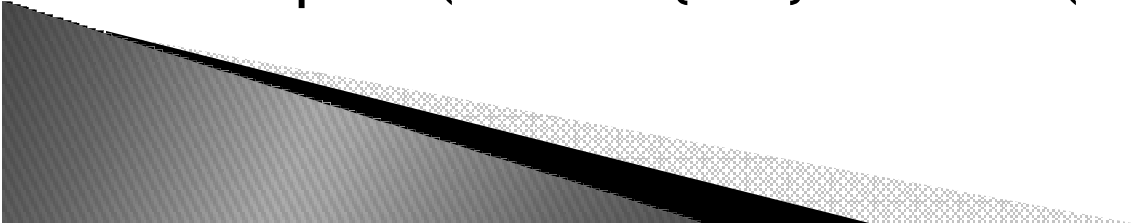


# Classe Abstrata

▶ Exemplo

```
class Student(SchoolMember):
    """Representa um aluno."""
    def __init__(self, name, age, marks):
        SchoolMember.__init__(self, name, age)
        self.marks = marks
        print('(Iniciado Student: {0})'.format(self.name))

    def tell(self):
        SchoolMember.tell(self)
        print('Nota: "{0:d}"'.format(self.marks))
```



# Classe Abstrata

- ▶ Exemplo

```
t = Teacher('Mrs. Shrividya', 40, 30000.00)  
s = Student('Swaroop', 25, 75)
```

```
t.tell()
```

```
m = SchoolMember('abc', 10)
```

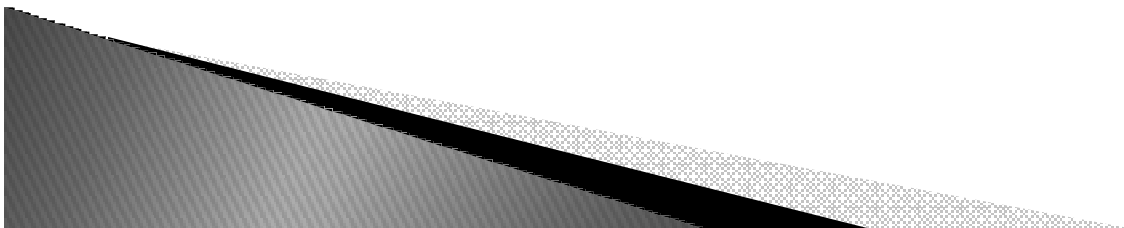
```
m.tell()
```

```
print() # imprime uma linha em branco
```

```
members = [t, s]
```

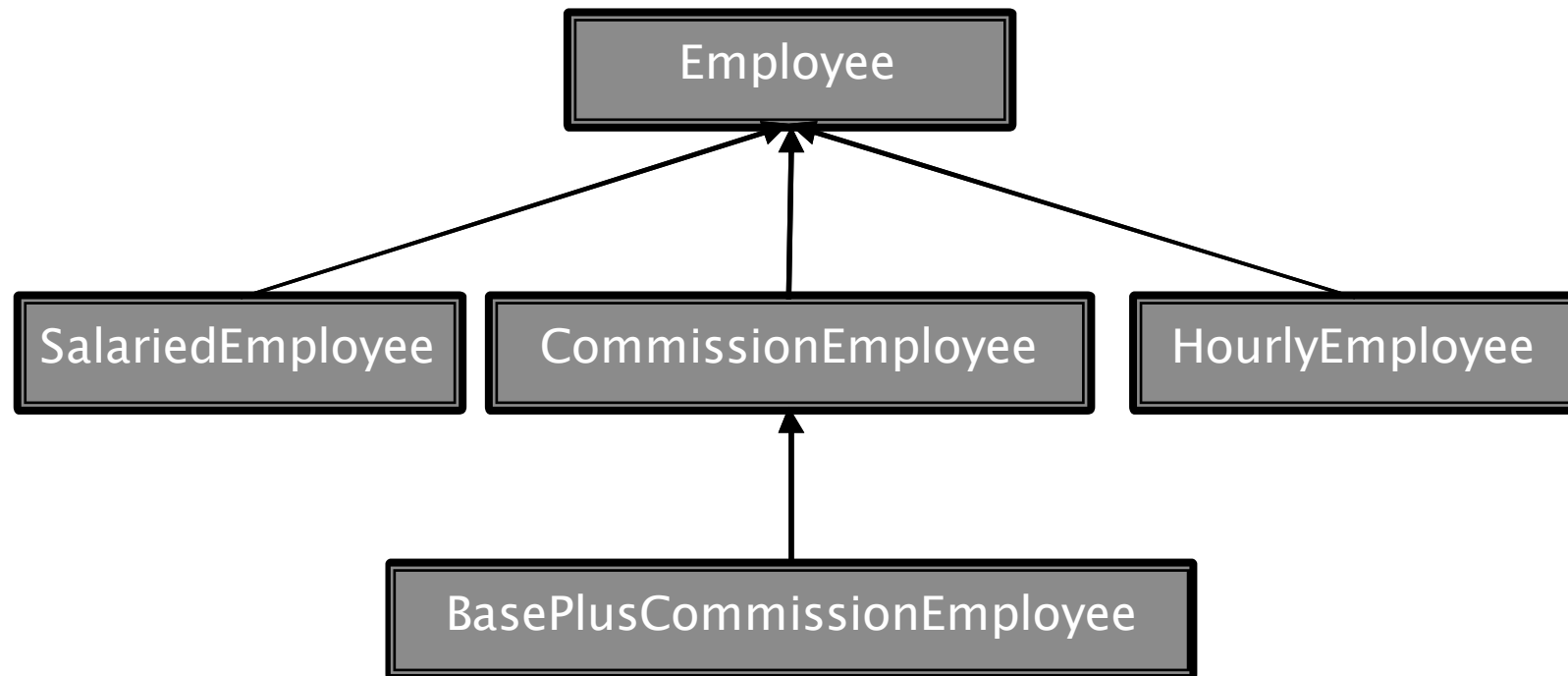
```
for member in members:
```

```
    member.tell() # funciona tanto para Teacher como para Student
```



# Classe Abstrata

## ▶ Exercício



# Exercício(cont.)

	earnings	toString
Employee	abstract	<i>firstName lastName</i> social security number: <i>SSN</i>
Salaries- Employee	weeklySalary	salaries employee: <i>firstName lastName</i> social security number: <i>SSN</i> weekly salary: <i>weeklysalar</i>
Hourly- Employee	<pre>if (hours &lt;= 40)     wage * hours else if (hours &gt; 40) {     40 * wage +     ( hours - 40 ) *     wage * 1.5 }</pre>	hourly employee: <i>firstName lastName</i> social security number: <i>SSN</i> hourly wage: <i>wage</i> ; hours worked: <i>hours</i>
Commission- Employee	commissionRate * grossSales	commission employee: <i>firstName lastName</i> social security number: <i>SSN</i> gross sales: <i>grossSales</i> ; commission rate: <i>commissionRate</i>
BasePlus- Commission- Employee	(commissionRate * grossSales) + baseSalary	base salaries commission employee: <i>firstName lastName</i> social security number: <i>SSN</i> gross sales: <i>grossSales</i> ; commission rate: <i>commissionRate</i> ; base salary: <i>baseSalary</i>