

Fundação CECIERJ
Tecnologia em Sistemas de Computação
Programação com Interfaces Gráficas
AD1 2º semestre de 2017
Gabarito

Professores: Mario Benevides e Paulo Roma

1 Introdução

O objetivo deste trabalho é proporcionar alguma experiência no uso de herança. Assim, será desenvolvido um código para uma locadora de vídeo. Já é fornecido parte do código, e informado as classes derivadas que devem ser implementadas.

Classes fornecidas:

- GenreSearch
- Item
- SearchCondition
- SimpleDate
- StatusException
- VideoStore

A serem implementadas:

- Customer: Representa o cliente da locadora. Principal método para implementar: **bringBackItem**. Subclasses: ClubMember e PremierMember
- AbstractItem: Esta classe é onde temos a implementação dos métodos da classe Item. Aqui temos a codificação de **setRented**, **setReturned**, **calculateLateFee**, entre outras. Subclasses: DVD, Game e NewReleaseDVD.
- TitleKeywordSearch: Implementar o método **matches**.

2 Item

A classe **AbstractItem** servirá de base para três novas classes. Essas classes devem ser divididas em três arquivos: DVD.py, Game.py e NewReleaseDVD.py. Desta forma, podemos programar os comportamentos especializados de cada item. A seguir, temos exemplos de como pode ser essa implementação.

2.1 DVD.py

```
from AbstractItem import AbstractItem

class DVD(AbstractItem):
    def __init__(self, title, genre, barcode):
        dias limite de aluguel = 5
        custo do aluguel = 350
        taxa atraso = dias_atraso * 100
```

2.2 Game.py

```
from AbstractItem import AbstractItem
from math import ceil
class Game(AbstractItem):
    def __init__(self, title, barcode):
        genre="GAME"
        dias limite de aluguel = 7
        custo do aluguel = 500
        taxa de atraso = 500 * ceil(dias_atraso / 7)
```

2.3 NewReleaseDVD.py

```
from AbstractItem import AbstractItem

class NewReleaseDVD(AbstractItem):
    def __init__(self, title, genre, barcode):
        dias limite de aluguel = 2
        custo do aluguel = 400
        entrega antecipada = -100
        taxa atraso = (dias_atraso - 1) * 150 + 400
```

3 Customer

A classe Customer modela um cliente comum, que não pode alugar novos itens caso possua itens em atraso. O método que faz essa verificação é o **canRent**, já implementado. O método **bringBackItem** deve ser codificado para o cliente regular, seguindo as regras definidas na AD.

Para cada tipo de cliente (ClubMember, PremierMember) essas regras são diferentes. Cada uma possui sua própria implementação de **bringBackItem**, e para essas subclasses, **canRent** sempre retorna True, pois é possível alugar mesmo com itens em atraso.

4 TitleKeywordSearch

Essa classe implementa SearchCondition em TitleKeywordSearch.py. Assim, temos o seguinte pseudocódigo para matches:

```
def matches(self, item):
    if allowSubStrings:
        pesquisa se existe a chave dentro titulo
```

```

        se achar retorna True
    else:
        pesquisa se a chave == ao titulo
        se for retorne True
    return False

```

5 Test.py

A forma mais básica de criar um arquivo de teste é importar todas a classes:

```

from SimpleDate import SimpleDate
from StatusException import StatusException
from AbstractItem import AbstractItem
from DVD import DVD
from NewReleaseDVD import NewReleaseDVD
from Game import Game
from Customer import Customer
from ClubMember import ClubMember
from PremierMember import PremierMember
from TitleKeywordSearch import TitleKeywordSearch

```

A partir deste ponto, criar instâncias das classes, aplicar as entradas e verificar os casos de teste.