

Linguagem de Programação I - Entrega 2 – Lucas Teles

diretório.arquivo: controle.escola_animal_aquatico

```
from src.util.gerais import imprimir_objetos, imprimir_objeto, imprimir_objetos_internos,
imprimir_objetos_associacao_filtros
from src.util.data import Data
from src.entidades.escola import inserir_escola, Escola, get_escolas
from src.entidades.animal_aquatico import inserir_animal_aquatico, AnimalAquatico, get_animais_aquaticos
from src.entidades.aquario import inserir_aquario, Aquario, get_aquarios
from src.entidades.passeio_turistico import criar_passeio_turistico, get_passeios_turisticos,
selecionar_passeios_turisticos

def cadastrar_escolas():
    inserir_escola(Escola('12.345.678/0001-01', 'Colégio Alpha', 1200, True))
    inserir_escola(Escola('23.456.789/0001-02', 'Escola Horizonte', 2000, False))
    inserir_escola(Escola('34.567.890/0001-03', 'Instituto Saber', 2000, True))
    inserir_escola(Escola('45.678.901/0001-04', 'Colégio Excelência', 950, False))
    inserir_escola(Escola('56.789.012/0001-05', 'Escola Pioneira', 1800, True))
    inserir_escola(Escola('67.890.123/0001-06', 'Centro Aurora', 600, False))
    inserir_escola(Escola('78.901.234/0001-07', 'Escola Nova Geração', 2500, True))
    inserir_escola(Escola('89.012.345/0001-08', 'Colégio Nexus', 1300, False))
    inserir_escola(Escola('90.123.456/0001-09', 'Escola do Futuro', 3000, True))
    inserir_escola(Escola('01.234.567/0001-10', 'Instituto Liberdade', 400, False))

def cadastrar_animais_aquaticos():
    inserir_animal_aquatico(AnimalAquatico('Beta', 'oceano', 19, True))
    inserir_animal_aquatico(AnimalAquatico('Tubarão', 'oceano', 250, False))
    inserir_animal_aquatico(AnimalAquatico('Dory', 'mar fechado', 200, False))
    inserir_animal_aquatico(AnimalAquatico('Nemo', 'mar fechado', 1, False))
    inserir_animal_aquatico(AnimalAquatico('Golfinho', 'oceano', 150, True))
    inserir_animal_aquatico(AnimalAquatico('Pirarucu', 'água doce', 220, True))
    inserir_animal_aquatico(AnimalAquatico('Estrela do Mar', 'mar fechado', 0.5, False))
    inserir_animal_aquatico(AnimalAquatico('Cavalo-marinho', 'oceano', 0.3, True))
    inserir_animal_aquatico(AnimalAquatico('Tartaruga Marinha', 'oceano', 200, True))
    inserir_animal_aquatico(AnimalAquatico('Peixe-boi', 'água doce', 500, True))

def cadastrar_aquarios():
    aquario = Aquario('Aqua Mundo', 'Guarujá - SP', 1000)
    inserir_aquario(aquario)
    aquario.inserir_animal_aquatico(AnimalAquatico('Tubarão', 'oceano', 250, False))

    aquario = Aquario('Oceanário de Aracaju', 'Aracaju - SE', 800)
    inserir_aquario(aquario)
    aquario.inserir_animal_aquatico(AnimalAquatico('Golfinho', 'oceano', 150, True))

    aquario = Aquario('Aquário de Ubatuba', 'Ubatuba - SP', 700)
    inserir_aquario(aquario)
    aquario.inserir_animal_aquatico(AnimalAquatico('Pirarucu', 'água doce', 220, True))

    aquario = Aquario('Projeto Tamar', 'Salvador - BA', 900)
    inserir_aquario(aquario)
    aquario.inserir_animal_aquatico(AnimalAquatico('Tartaruga Marinha', 'oceano', 200, True))

    aquario = Aquario('Aquário de Bonito', 'Recife - PE', 600)
    inserir_aquario(aquario)
    aquario.inserir_animal_aquatico(AnimalAquatico('Dory', 'mar fechado', 200, False))

    aquario = Aquario('Sea Life', 'São Paulo - SP', 1200)
    inserir_aquario(aquario)
```

```

aquario.inserir_animal_aquatico(AnimalAquatico('Estrela do Mar', 'mar fechado', 0.5, False))

aquario = Aquario('Mundo Submarino', 'Recife - PE', 1500)
inserir_aquario(aquario)
aquario.inserir_animal_aquatico(AnimalAquatico('Cavalo-marinho', 'oceano', 0.3, True))

aquario = Aquario('Aquário Amazônico', 'Manaus - AM', 1300)
inserir_aquario(aquario)
aquario.inserir_animal_aquatico(AnimalAquatico('Peixe-boi', 'água doce', 500, True))

def cadastrar_passeios_turisticos():
    criar_passeio_turistico('Aqua Mundo', 'Tubarão', 'Colégio Alpha', Data(10, 5, 2014))
    criar_passeio_turistico('Aquário de Bonito', 'Dory', 'Escola Horizonte', Data(12, 5, 2012))
    criar_passeio_turistico('Sea Life', 'Estrela do Mar', 'Centro Aurora', Data(7, 6, 2022))
    criar_passeio_turistico('Projeto Tamar', 'Tartaruga Marinha', 'Escola do Futuro', Data(5, 6, 2015))
    criar_passeio_turistico('Mundo Submarino', 'Cavalo-marinho', 'Instituto Liberdade', Data(30, 5, 2011))

def imprimir_somente_para_alinhar_formatação():
    print("\nAquários : nome - cidade - capacidade")
    for índice, aquario in enumerate(get_aquarios().values()): print(aquario)
    print("\nAnimais Aquáticos dos Aquários:")
    for aquario in get_aquarios().values():
        for animal in aquario.animal_aquatico.values(): print(animal)

if __name__ == '__main__':
    cadastrar_escolas()
    imprimir_objetos("\nEscolas : nome - CNPJ - número de alunos - possui programa ambiental",
get_escolas().values())

    cadastrar_aquarios()
    imprimir_somente_para_alinhar_formatação()

    print("\nAquários : nome - cidade - capacidade")
    print(' - Animais Aquáticos : nome - habitat - peso - risco de extinção')
    for índice, aquario in enumerate(get_aquarios().values()):
        imprimir_objeto(índice, str(aquario))
        imprimir_objetos_internos(aquario.animal_aquatico.values())

    cadastrar_passeios_turisticos()
    cabeçalho = 'Passeios Turísticos : nome do aquário - nome do animal aquático - nome da escola - data
do passeio'
    imprimir_objetos("\n" + cabeçalho, get_passeios_turisticos())

    cabeçalho_filtros = cabeçalho + '\n -- número de alunos da escola - cidade do aquário - peso do animal
aquático'

    filtros, selecionados = selecionar_passeios_turisticos()
    imprimir_objetos_associação_filtros(cabeçalho_filtros, selecionados, filtros)

    filtros, selecionados = selecionar_passeios_turisticos(data_minima_passeio=Data(1, 1, 2012))
    imprimir_objetos_associação_filtros(cabeçalho_filtros, selecionados, filtros)

    filtros, selecionados = selecionar_passeios_turisticos(Data(1, 1, 2012), peso_minimo_animal=150)
    imprimir_objetos_associação_filtros(cabeçalho_filtros, selecionados, filtros)

    filtros, selecionados = selecionar_passeios_turisticos(Data(1, 1, 2012), 150, max_alunos_escola=2000)
    imprimir_objetos_associação_filtros(cabeçalho_filtros, selecionados, filtros)

    filtros, selecionados = selecionar_passeios_turisticos(Data(1, 1, 2012), 150, 2000, cidade_aquario='Recife
- PE')
    imprimir_objetos_associação_filtros(cabeçalho_filtros, selecionados, filtros)

```

diretório.arquivo: entidades.animal_aquatico

```
animais_aquaticos = []
```

```
def get_animais_aquaticos():  
    return animais_aquaticos
```

```
def inserir_animal_aquatico(animal):  
    animais_aquaticos.append(animal)
```

```
def selecionar_animal_aquatico(habitat=None, risco_extincao=None, peso_máximo=None):  
    filtros = '\nFiltros -- '  
    if habitat: filtros += 'habitat: ' + habitat  
    if risco_extincao in (True, False): filtros += ' - risco de extinção: ' + str(risco_extincao)  
    if peso_máximo: filtros += ' - peso máximo: ' + str(peso_máximo)
```

```
    selecionados = []  
    for animal in animais_aquaticos:  
        if habitat and animal.habitat != habitat:  
            continue  
        if risco_extincao in (True, False) and animal.risco_extincao != risco_extincao:  
            continue  
        if peso_máximo and animal.peso > peso_máximo:  
            continue  
        selecionados.append(animal)  
    return filtros, selecionados
```

```
class AnimalAquatico:  
    def __init__(self, nome, habitat, peso, risco_extincao):  
        self.nome = nome  
        self.habitat = habitat if habitat in ('oceano', 'água doce', 'mar fechado') else 'indefinido'  
        self.peso = peso  
        self.risco_extincao = risco_extincao  
  
    def __str__(self):  
        risco = 'em risco' if self.risco_extincao else 'não ameaçado'  
        formato = '{} {:<20} {} {:<12} {} {:<6} {} {:<12} {}'  
        return formato.format('|', self.nome, '|', self.habitat, '|', str(self.peso) + ' kg', '|', risco, '|')
```

diretório.arquivo: entidades.aquario

```
aquarios = {}
```

```
def get_aquarios():  
    return aquarios
```

```
def inserir_aquario(aquario):  
    nome_aquario = aquario.nome  
    if nome_aquario not in aquarios:  
        aquarios[nome_aquario] = aquario  
        return True  
    else:  
        print('Aquário ' + nome_aquario + ' já está cadastrado')  
        return False
```

```
class Aquario:  
    def __init__(self, nome, cidade, capacidade):  
        self.nome = nome  
        self.cidade = cidade  
        self.capacidade = capacidade  
        self.animal_aquatico = {}  
  
    def __str__(self):  
        formato = '{} {:<25} {} {:<18} {} {:<6} {}'  
        return formato.format('|', self.nome, '|', self.cidade, '|', str(self.capacidade), '|')  
  
    def inserir_animal_aquatico(self, animal):  
        nome_animal = animal.nome  
        if nome_animal not in self.animal_aquatico:  
            self.animal_aquatico[nome_animal] = animal  
        else:  
            print(f'Animal aquático {nome_animal} já está cadastrado no aquário')
```

diretório.arquivo: entidades.escola

```
escolas = {}

def get_escolas():
    return escolas

def inserir_escola(escola):
    cnpj = escola.cnpj
    if cnpj not in escolas:
        escolas[cnpj] = escola
        return True
    else:
        print('Escola com CNPJ ' + cnpj + ' já está cadastrada')
        return False

def selecionar_escola(max_alunos=None, possui_programa_ambiental=None, prefixo_nome=None):
    filtros = '\nFiltros -- '
    if max_alunos is not None: filtros += 'máximo de alunos: ' + str(max_alunos)
    if possui_programa_ambiental in (True, False): filtros += ' - possui programa ambiental: ' +
str(possui_programa_ambiental)
    if prefixo_nome is not None: filtros += ' - prefixo do nome: ' + prefixo_nome

    selecionadas = []
    for escola in escolas.values():
        if max_alunos is not None and escola.n_alunos > max_alunos:
            continue
        if possui_programa_ambiental in (True, False) and escola.programa_ambiental !=
possui_programa_ambiental:
            continue
        if prefixo_nome and not escola.nome.startswith(prefixo_nome):
            continue
        selecionadas.append(escola)
    return filtros, selecionadas

class Escola:
    def __init__(self, cnpj, nome, n_alunos, programa_ambiental):
        self.cnpj = cnpj
        self.nome = nome
        self.n_alunos = n_alunos
        self.programa_ambiental = programa_ambiental

    def __str__(self):
        ambiental = 'sim' if self.programa_ambiental else 'não'
        formato = '{} {:<22} {} {:<18} {} {:<5} {} {:<3} {}'
        return formato.format('|', self.nome, '|', self.cnpj, '|', self.n_alunos, '|', ambiental, '|')
```

diretório.arquivo: entidades.passeio_turistico

```
from src.entidades.aquario import get_aquarios
from src.entidades.animal_aquatico import get_animais_aquaticos
from src.entidades.escola import get_escolas

passeios_turisticos = []

def get_passeios_turisticos():
    return passeios_turisticos

def inserir_passeio_turistico(passeio):
    if passeio not in passeios_turisticos:
        passeios_turisticos.append(passeio)
    else:
        print('Passeio turístico já cadastrado --- ' + str(passeio))

def criar_passeio_turistico(nome_aquario, nome_animal, nome_escola, data):
    aquario = get_aquarios().get(nome_aquario)
    if not aquario:
        print('Aquário ' + nome_aquario + ' não cadastrado')
        return
    animal = aquario.animal_aquatico.get(nome_animal)
    if not animal:
        print('Animal aquático ' + nome_animal + ' não cadastrado no aquário ' + nome_aquario)
        return
    escola = None
    for e in get_escolas().values():
        if e.nome == nome_escola:
            escola = e
            break
    if not escola:
        print('Escola ' + nome_escola + ' não cadastrada')
        return
    passeio = PasseioTuristico(aquario, animal, escola, data)
    inserir_passeio_turistico(passeio)

def selecionar_passeios_turisticos(data_minima_passeio=None, peso_minimo_animal=None,
                                   max_alunos_escola=None, cidade_aquario=None):
    filtros = '\nFiltros -- '
    if data_minima_passeio: filtros += 'data minima: ' + str(data_minima_passeio)
    if peso_minimo_animal: filtros += f' - peso minimo do animal: ' + str(peso_minimo_animal)
    if max_alunos_escola: filtros += f' - máximo de alunos na escola: ' + str(max_alunos_escola)
    if cidade_aquario: filtros += f' - cidade do aquário: ' + str(cidade_aquario)

    selecionados = []
    for passeio in passeios_turisticos:
        if data_minima_passeio and passeio.data < data_minima_passeio:
            continue
        if peso_minimo_animal and passeio.animal_aquatico.peso < peso_minimo_animal:
            continue
        if max_alunos_escola and passeio.escola.n_alunos > max_alunos_escola:
            continue
        if cidade_aquario and passeio.aquario.cidade != cidade_aquario:
            continue
        selecionados.append(passeio)
    return filtros, selecionados

class PasseioTuristico:
    def __init__(self, aquario, animal_aquatico, escola, data):
        self.aquario = aquario
        self.animal_aquatico = animal_aquatico
```

```
self.escola = escola
self.data = data

def __str__(self):
    formato = '{ } {:<25} { } {:<18} { } {:<20} { } {:<10} { }'
    return formato.format('|', self.aquario.nome, '|', self.animal_aquatico.nome, '|',
                        self.escola.nome, '|', str(self.data), '|')

def str_filtro(self):
    formato = '{:>5} { } {:<18} { } {:<6} { }'
    return self.__str__() + formato.format(self.escola.n_alunos, '|',
                                          self.aquario.cidade, '|',
                                          str(self.animal_aquatico.peso) + 'kg', '|')
```

diretório.arquivo: util.data

```
from datetime import date
```

```
class Data:
```

```
    def __init__(self, dia, mês, ano):  
        self.dia = dia  
        self.mês = mês  
        self.ano = ano
```

```
    def __str__(self):  
        if self.dia < 10:  
            data_str = '0' + str(self.dia)  
        else:  
            data_str = str(self.dia)  
  
        if self.mês < 10:  
            data_str += "/0" + str(self.mês) + "/"  
        else:  
            data_str += "/" + str(self.mês) + "/"
```

```
        data_str += str(self.ano)  
        return data_str
```

```
    def __eq__(self, data):  
        if self.dia == data.dia and self.mês == data.mês and self.ano == data.ano:  
            return True  
        return False
```

```
    def __ne__(self, data):  
        return not self == data
```

```
    def __gt__(self, data):  
        if self.ano > data.ano:  
            return True  
        elif self.ano < data.ano:  
            return False  
  
        if self.mês > data.mês:  
            return True  
        elif self.mês < data.mês:  
            return False
```

```
        if self.dia > data.dia:  
            return True  
        elif self.dia < data.dia:  
            return False
```

```
        return False
```


diretório.arquivo: util.gerais

```
def imprimir_objetos(cabeçalho, objetos, filtros=None):
    if filtros is not None: print(filtros)
    print(cabeçalho)
    for índice, objeto in enumerate(objetos):
        imprimir_objeto(indíce, str(objeto))

def imprimir_objeto(indíce, objeto_str):
    formato = '{} {} {}'
    ordem = índice + 1
    separador = '-'
    string_formatado = formato.format(f'{ordem:2d}', separador, objeto_str)
    print(string_formatado)

def imprimir_objetos_associação_filtros(cabeçalho, objetos, filtros=None):
    if filtros is not None: print(filtros)
    print(cabeçalho)
    for índice, objeto in enumerate(objetos):
        imprimir_objeto(indíce, objeto.str_filtro())

def imprimir_objetos_internos(objetos):
    for objeto in objetos: print('- ' + str(objeto))
```

Dourados MS, 22/04/2025