



Título da Prática: Criação do Cadastro em Modo Texto

Objetivo da Prática: Criar um código Java para gerenciar informações de Pessoas Físicas e Jurídicas. Foram utilizados conceitos de herança de classes, serialização de objetos para salvar em arquivos e a implementação de uma interface para persistência.

Classe Main:

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        PessoaFisicaRepo repoPessoaFisica = new PessoaFisicaRepo();
        PessoaJuridicaRepo repoPessoaJuridica = new PessoaJuridicaRepo();

        int escolha;

        do {
            System.out.println("Opções:");
            System.out.println("1. Incluir");
            System.out.println("2. Alterar");
            System.out.println("3. Excluir");
            System.out.println("4. Obter por ID");
            System.out.println("5. Obter Todos");
            System.out.println("6. Salvar Dados");
            System.out.println("7. Recuperar Dados");
            System.out.println("0. Sair");
            System.out.print("Escolha uma opção: ");

            escolha = scanner.nextInt();
            scanner.nextLine();

            switch (escolha) {
                case 1:
                    System.out.print("Tipo (1 para Física, 2 para Jurídica): ");
                    int tipo = scanner.nextInt();
                    scanner.nextLine();

                    if (tipo == 1) {
                        System.out.print("ID: ");
                        int id = scanner.nextInt();
                        scanner.nextLine();
                        System.out.print("Nome: ");
                        String nome = scanner.nextLine();
                        System.out.print("CPF: ");
                        String cpf = scanner.nextLine();
                        System.out.print("Idade: ");
                        int idade = scanner.nextInt();
                        scanner.nextLine();
```

```

        repoPessoaFisica.inserir(new PessoaFisica(id, nome, cpf, idade));
    } else if (tipo == 2) {
        System.out.print("ID: ");
        int id = scanner.nextInt();
        scanner.nextLine();
        System.out.print("Nome: ");
        String nome = scanner.nextLine();
        System.out.print("CNPJ: ");
        String cnpj = scanner.nextLine();

        repoPessoaJuridica.inserir(new PessoaJuridica(id, nome, cnpj));
    } else {
        System.out.println("Tipo inválido.");
    }
}
break;

```

case 2:

```

System.out.print("Tipo (1 para Física, 2 para Jurídica): ");
int tipoAlterar = scanner.nextInt();
scanner.nextLine();

if (tipoAlterar == 1) {
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    PessoaFisica pessoaFisica = repoPessoaFisica.obter(id);
    if (pessoaFisica != null) {
        System.out.println("Dados atuais:");
        pessoaFisica.exibir();
        System.out.print("Novo nome: ");
        String novoNome = scanner.nextLine();
        System.out.print("Novo CPF: ");
        String novoCpf = scanner.nextLine();
        System.out.print("Nova idade: ");
        int novaIdade = scanner.nextInt();
        scanner.nextLine();

        pessoaFisica.setNome(novoNome);
        pessoaFisica.setCpf(novoCpf);
        pessoaFisica.setIdade(novaIdade);
        repoPessoaFisica.alterar(pessoaFisica);
        System.out.println("Pessoa Física alterada com sucesso.");
    } else {
        System.out.println("Pessoa Física não encontrada.");
    }
} else if (tipoAlterar == 2) {
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    PessoaJuridica pessoaJuridica = repoPessoaJuridica.obter(id);
    if (pessoaJuridica != null) {
        System.out.println("Dados atuais:");
        pessoaJuridica.exibir();
        System.out.print("Novo nome: ");
        String novoNome = scanner.nextLine();

```

```

        System.out.print("Novo CNPJ: ");
        String novoCnpj = scanner.nextLine();

        pessoaJuridica.setNome(novoNome);
        pessoaJuridica.setCnpj(novoCnpj);
        repoPessoaJuridica.alterar(pessoaJuridica);
        System.out.println("Pessoa Jurídica alterada com sucesso.");
    } else {
        System.out.println("Pessoa Jurídica não encontrada.");
    }
} else {
    System.out.println("Tipo inválido.");
}
break;

```

case 3:

```

System.out.print("Tipo (1 para Física, 2 para Jurídica): ");
int tipoExcluir = scanner.nextInt();
scanner.nextLine();

if (tipoExcluir == 1) {
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    repoPessoaFisica.excluir(id);
    System.out.println("Pessoa Física excluída com sucesso.");
} else if (tipoExcluir == 2) {
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    repoPessoaJuridica.excluir(id);
    System.out.println("Pessoa Jurídica excluída com sucesso.");
} else {
    System.out.println("Tipo inválido.");
}
break;

```

case 4:

```

System.out.print("Tipo (1 para Física, 2 para Jurídica): ");
int tipoObter = scanner.nextInt();
scanner.nextLine();

if (tipoObter == 1) {
    System.out.print("ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    PessoaFisica pessoaFisica = repoPessoaFisica.obter(id);
    if (pessoaFisica != null) {
        System.out.println("Dados da Pessoa Física:");
        pessoaFisica.exibir();
    } else {
        System.out.println("Pessoa Física não encontrada.");
    }
} else if (tipoObter == 2) {
    System.out.print("ID: ");

```

```

int id = scanner.nextInt();
scanner.nextLine();
PessoaJuridica pessoaJuridica = repoPessoaJuridica.obter(id);
if (pessoaJuridica != null) {
    System.out.println("Dados da Pessoa Jurídica:");
    pessoaJuridica.exibir();
} else {
    System.out.println("Pessoa Jurídica não encontrada.");
}
} else {
    System.out.println("Tipo inválido.");
}
break;

```

case 5:

```

System.out.print("Tipo (1 para Física, 2 para Jurídica): ");
int tipoObterTodos = scanner.nextInt();
scanner.nextLine();

if (tipoObterTodos == 1) {
    List<PessoaFisica> pessoasFisicas = repoPessoaFisica.obterTodos();
    System.out.println("Pessoas Físicas:");
    for (PessoaFisica pf : pessoasFisicas) {
        pf.exibir();
    }
} else if (tipoObterTodos == 2) {
    List<PessoaJuridica> pessoasJuridicas = repoPessoaJuridica.obterTodos();
    System.out.println("Pessoas Jurídicas:");
    for (PessoaJuridica pj : pessoasJuridicas) {
        pj.exibir();
    }
} else {
    System.out.println("Tipo inválido.");
}
break;

```

case 6:

```

System.out.print("Prefixo dos arquivos: ");
String prefixo = scanner.nextLine();
try {
    repoPessoaFisica.persistir(prefixo + ".fisica.bin");
    repoPessoaJuridica.persistir(prefixo + ".juridica.bin");
    System.out.println("Dados salvos com sucesso.");
} catch (IOException e) {
    System.err.println("Erro ao salvar os dados: " + e.getMessage());
}
break;

```

case 7:

```

System.out.print("Prefixo dos arquivos: ");
prefixo = scanner.nextLine();
try {
    repoPessoaFisica = PessoaFisicaRepo.recuperar(prefixo + ".fisica.bin");
    repoPessoaJuridica = PessoaJuridicaRepo.recuperar(prefixo + ".juridica.bin");
    System.out.println("Dados recuperados com sucesso.");
} catch (IOException | ClassNotFoundException e) {

```

```

        System.err.println("Erro ao recuperar os dados: " + e.getMessage());
    }
    break;

    case 0:
        System.out.println("Encerrando o programa.");
        break;

    default:
        System.out.println("Opção inválida.");
    }

} while (escolha != 0);

scanner.close();
}
}

```

Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 1

Tipo (1 para Física, 2 para Jurídica): 1

ID: 1

Nome: joao

CPF: 1111

Idade: 20

Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 2

Tipo (1 para Física, 2 para Jurídica): 1

ID: 1

Dados atuais:

ID: 1

Nome: joao

CPF: 1111

Idade: 20

Novo nome: carlos

Novo CPF: 0000

Nova idade: 30

Pessoa Física alterada com sucesso.

Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 3

Tipo (1 para Física, 2 para Jurídica): 1

ID: 1

Pessoa Física excluída com sucesso.

Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 4

Tipo (1 para Física, 2 para Jurídica): 1

ID: 2

Dados da Pessoa Física:

ID: 2

Nome: pedro

CPF: 2222

Idade: 25

Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 6

Prefixo dos arquivos: teste

Dados salvos com sucesso.

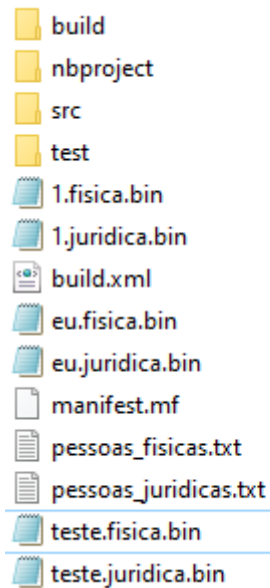
Opções:

1. Incluir
2. Alterar
3. Excluir
4. Obter por ID
5. Obter Todos
6. Salvar Dados
7. Recuperar Dados
0. Sair

Escolha uma opção: 7

Prefixo dos arquivos: teste

Dados recuperados com sucesso.



Elementos estáticos em Java:

São partes de uma classe que pertencem à classe como um todo, não a instâncias específicas. O método 'Main' é declarado como estático para ser o ponto de entrada global em um programa Java, acessível de qualquer lugar sem a necessidade de criar objetos, permitindo que o programa seja iniciado.

Classe Scanner:

Serve para ler dados de entrada do usuário ou arquivos. Ela fornece métodos para ler e analisar diferentes tipos de dados, como números inteiros, números de ponto flutuante, strings e muito mais. Em resumo, o Scanner facilita a interação com dados de entrada no programa.

Uso de classes:

O uso de classes de repositório melhora a organização do código, mantendo a lógica de armazenamento de dados separada e facilitando a reutilização, manutenção e testabilidade do código.