Commit

⚠ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Refatoração em PessoaDAO

Browse files

```
PessoaDAO - Técnicas: Extract Method, Extract Constant
```

 **moisesocosta** committed on Jun 21

1 parent 9d330b1 commit a9b78f6

Showing **1 changed file** with **141 additions** and **178 deletions**.

Whitespace Ignore whitespace Split Unified

319 PessoaDAO.java

```
13  13
14  14  public class PessoaDAO implements GenericoDAO<PessoaDTO> {
15  15
16      -       @Override
17      -       public void inserir(PessoaDTO pessoaDTO) throws PersistenciaExcpetion {
18      -               try {
19      -                       Connection connection = ConexaoUtil.getInstance().getConnection();
20      -                       String sql = "INSERT INTO TB_PESSOA(NOME, CPF, ENDERECO, SEXO, DT_NASC) " +
21      -                                       "VALUES(?, ?, ?, ?, ?)";
22      -
23      -                       PreparedStatement statement = connection.prepareStatement(sql);
24      -                       statement.setString(1, pessoaDTO.getNome());
25      -                       statement.setLong(2, pessoaDTO.getCpf());
26      -                       statement.setString(3, pessoaDTO.getEndereco());
27      -                       statement.setString(4, String.valueOf(pessoaDTO.getSexo()));
28      -                       statement.setDate(5, new Date(pessoaDTO.getDtNascimento().getTime()));
29      -
30      -                       statement.execute();
31      -                       connection.close();
```

```
-                } catch (Exception e) {
-                        e.printStackTrace();
-                        throw new PersistenciaExcpetion(e.getMessage(), e);
-                }
-        }
-
-        @Override
-        public void atualizar(PessoaDTO pessoaDTO) throws PersistenciaExcpetion {
-                try {
-                        Connection connection =
  ConexaoUtil.getInstance().getConnection();
-
-                        String sql = "UPDATE TB_PESSOA " +
-                                        " SET NOME = ?, " +
-                                        " CPF = ?," +
-                                        " ENDERECO = ?," +
-                                        " SEXO = ?," +
-                                        " DT_NASC = ? " +
-                                        " WHERE ID_PESSOA = ?";
-
-                        PreparedStatement statement =
  connection.prepareStatement(sql);
-                        statement.setString(1, pessoaDTO.getNome());
-                        statement.setLong(2, pessoaDTO.getCpf());
-                        statement.setString(3, pessoaDTO.getEndereco());
-                        statement.setString(4,
  String.valueOf(pessoaDTO.getSexo()));
-                        statement.setDate(5, new
  Date(pessoaDTO.getDtNascimento().getTime()));
-                        statement.setInt(6, pessoaDTO.getIdPessoa());
-
-                        statement.execute();
-                        connection.close();
-                } catch(Exception e) {
-                        e.printStackTrace();
-                        throw new PersistenciaExcpetion(e.getMessage(), e);
-                }
-        }
-
-        @Override
-        public void deletar(Integer idPessoa) throws PersistenciaExcpetion {
-                try {
-                        Connection connection =
  ConexaoUtil.getInstance().getConnection();
-
-                        String sql = "DELETE FROM TB_PESSOA WHERE ID_PESSOA = ?";
-
-                        PreparedStatement statement =
  connection.prepareStatement(sql);
-                        statement.setInt(1, idPessoa);
-
-                        statement.execute();
```

```
78  -                       connection.close();
79  -               } catch(Exception e) {
80  -                       e.printStackTrace();
81  -                       throw new PersistenciaExcpetion(e.getMessage(), e);
82  -               }
83  -       }
84  -
85  -       @Override
86  -       public List<PessoaDTO> listarTodos() throws PersistenciaExcpetion {
87  -               List<PessoaDTO> listaPessoas = new ArrayList<PessoaDTO>();
88  -               try {
89  -                       Connection connection =
      ConexaoUtil.getInstance().getConnection();
90  -
91  -                       String sql = "SELECT * FROM TB_PESSOA";
92  -
93  -                       PreparedStatement statement =
      connection.prepareStatement(sql);
94  -                       ResultSet resultSet = statement.executeQuery();
95  -
96  -                       while (resultSet.next()) {
97  -                               PessoaDTO pessoaDTO = new PessoaDTO();
98  -
      pessoaDTO.setIdPessoa(resultSet.getInt("id_pessoa"));
99  -                               pessoaDTO.setNome(resultSet.getString("nome"));
100 -                               pessoaDTO.setCpf(resultSet.getLong("cpf"));
101 -
      pessoaDTO.setDtNascimento(resultSet.getDate("dt_nasc"));
102 -
      pessoaDTO.setEndereco(resultSet.getString("endereco"));
103 -
      pessoaDTO.setSexo(resultSet.getString("sexo").charAt(0));
104 -
105 -                               listaPessoas.add(pessoaDTO);
106 -                       }
107 -                       connection.close();
108 -               } catch (Exception e) {
109 -                       e.printStackTrace();
110 -                       throw new PersistenciaExcpetion(e.getMessage(), e);
111 -               }
112 -               return listaPessoas;
113 -       }
114 -
115 -       @Override
116 -       public PessoaDTO buscarPorId(Integer id) throws PersistenciaExcpetion {
117 -               PessoaDTO pessoaDTO = null;
118 -               try {
119 -                       Connection connection =
      ConexaoUtil.getInstance().getConnection();
120 -
121 -                       String sql = "SELECT * FROM TB_PESSOA WHERE ID_PESSOA = ?";
122 -
```

```
123 -                       PreparedStatement statement =
    connection.prepareStatement(sql);
124 -                       statement.setInt(1, id);
125 -
126 -                       ResultSet resultSet = statement.executeQuery();
127 -                       if (resultSet.next()) {
128 -                               pessoaDTO = new PessoaDTO();
129 -
    pessoaDTO.setIdPessoa(resultSet.getInt("id_pessoa"));
130 -                               pessoaDTO.setNome(resultSet.getString("nome"));
131 -                               pessoaDTO.setCpf(resultSet.getLong("cpf"));
132 -
    pessoaDTO.setDtNascimento(resultSet.getDate("dt_nasc"));
133 -
    pessoaDTO.setEndereco(resultSet.getString("endereco"));
134 -
    pessoaDTO.setSexo(resultSet.getString("sexo").charAt(0));
135 -                       }
136 -                       connection.close();
137 -               } catch (Exception e) {
138 -                       e.printStackTrace();
139 -                       throw new PersistenciaExcpetion(e.getMessage(), e);
140 -               }
141 -               return pessoaDTO;
142 -       }
143 -
144 -       public List<PessoaDTO> filtraPessoa(String nome, Long cpf, String sexo)
    throws PersistenciaExcpetion {
145 -               List<PessoaDTO> lista = new ArrayList<PessoaDTO>();
146 -               try {
147 -                       Connection connection =
    ConexaoUtil.getInstance().getConnection();
148 -
149 -                       String sql = "SELECT * FROM TB_PESSOA ";
150 -                       if (nome != null && !nome.equals("")) {
151 -                               sql += " WHERE NOME = ?";
152 -                       }
153 -
154 -                       if (cpf != null && !cpf.equals("")) {
155 -                               sql += " AND CPF = ?";
156 -                       }
157 -
158 -                       if (sexo != null && !sexo.equals("")) {
159 -                               sql += " AND SEXO = ?";
160 -                       }
161 -
162 -                       PreparedStatement statement =
    connection.prepareStatement(sql);
163 -                       if (nome != null && !nome.equals("")) {
164 -                               statement.setString(1, nome);
165 -                       }
166 -
```

```
167 -                       if (cpf != null && !cpf.equals("")) {
168 -                               statement.setLong(2, cpf);
169 -                       }
170 -
171 -                       if (sexo != null && !sexo.equals("")) {
172 -                               statement.setString(3, sexo);
173 -                       }
174 -
175 -                       ResultSet resultSet = statement.executeQuery();
176 -
177 -                       while (resultSet.next()) {
178 -                               PessoaDTO pessoaDTO = new PessoaDTO();
179 -
    pessoaDTO.setIdPessoa(resultSet.getInt("id_pessoa"));
180 -                               pessoaDTO.setNome(resultSet.getString("nome"));
181 -                               pessoaDTO.setCpf(resultSet.getLong("cpf"));
182 -
    pessoaDTO.setDtNascimento(resultSet.getDate("dt_nasc"));
183 -
    pessoaDTO.setEndereco(resultSet.getString("endereco"));
184 -
    pessoaDTO.setSexo(resultSet.getString("sexo").charAt(0));
185 -
186 -                               lista.add(pessoaDTO);
187 -                       }
188 -               } catch(Exception e) {
189 -                       e.printStackTrace();
190 -                       throw new PersistenciaExcpetion(e.getMessage(), e);
191 -               }
192 -               return lista;
193 -       }
 16 +     private static final String INSERT_SQL = "INSERT INTO TB_PESSOA(NOME, CPF,
    ENDERECO, SEXO, DT_NASC) VALUES(?, ?, ?, ?, ?)";
 17 +     private static final String UPDATE_SQL = "UPDATE TB_PESSOA SET NOME = ?, CPF =
    ?, ENDERECO = ?, SEXO = ?, DT_NASC = ? WHERE ID_PESSOA = ?";
 18 +     private static final String DELETE_SQL = "DELETE FROM TB_PESSOA WHERE ID_PESSOA
    = ?";
 19 +     private static final String SELECT_ALL_SQL = "SELECT * FROM TB_PESSOA";
 20 +     private static final String SELECT_BY_ID_SQL = "SELECT * FROM TB_PESSOA WHERE
    ID_PESSOA = ?";
 21 +     private static final String SELECT_BY_FILTER_SQL = "SELECT * FROM TB_PESSOA
    WHERE 1 = 1";
 22 +
 23 +     @Override
 24 +     public void inserir(PessoaDTO pessoaDTO) throws PersistenciaExcpetion {
 25 +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
 26 +              PreparedStatement statement = connection.prepareStatement(INSERT_SQL))
    {
 27 +             setPessoaStatementValues(statement, pessoaDTO);
 28 +             statement.execute();
 29 +         } catch (Exception e) {
 30 +             e.printStackTrace();
```

```
31 +             throw new PersistenciaExcpetion(e.getMessage(), e);
32 +         }
33 +     }
34 +
35 +     @Override
36 +     public void atualizar(PessoaDTO pessoaDTO) throws PersistenciaExcpetion {
37 +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
38 +              PreparedStatement statement = connection.prepareStatement(UPDATE_SQL))
   {
39 +             setPessoaStatementValues(statement, pessoaDTO);
40 +             statement.setInt(6, pessoaDTO.getIdPessoa());
41 +             statement.execute();
42 +         } catch (Exception e) {
43 +             e.printStackTrace();
44 +             throw new PersistenciaExcpetion(e.getMessage(), e);
45 +         }
46 +     }
47 +
48 +     @Override
49 +     public void deletar(Integer idPessoa) throws PersistenciaExcpetion {
50 +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
51 +              PreparedStatement statement = connection.prepareStatement(DELETE_SQL))
   {
52 +             statement.setInt(1, idPessoa);
53 +             statement.execute();
54 +         } catch (Exception e) {
55 +             e.printStackTrace();
56 +             throw new PersistenciaExcpetion(e.getMessage(), e);
57 +         }
58 +     }
59 +
60 +     @Override
61 +     public List<PessoaDTO> listarTodos() throws PersistenciaExcpetion {
62 +         List<PessoaDTO> listaPessoas = new ArrayList<>();
63 +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
64 +              PreparedStatement statement =
   connection.prepareStatement(SELECT_ALL_SQL);
65 +              ResultSet resultSet = statement.executeQuery()) {
66 +             while (resultSet.next()) {
67 +                 PessoaDTO pessoaDTO = getPessoaFromResultSet(resultSet);
68 +                 listaPessoas.add(pessoaDTO);
69 +             }
70 +         } catch (Exception e) {
71 +             e.printStackTrace();
72 +             throw new PersistenciaExcpetion(e.getMessage(), e);
73 +         }
74 +         return listaPessoas;
75 +     }
76 +
77 +     @Override
78 +     public PessoaDTO buscarPorId(Integer id) throws PersistenciaExcpetion {
79 +         PessoaDTO pessoaDTO = null;
```

```
80  +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
81  +              PreparedStatement statement =
    connection.prepareStatement(SELECT_BY_ID_SQL)) {
82  +             statement.setInt(1, id);
83  +             try (ResultSet resultSet = statement.executeQuery()) {
84  +                 if (resultSet.next()) {
85  +                     pessoaDTO = getPessoaFromResultSet(resultSet);
86  +                 }
87  +             }
88  +         } catch (Exception e) {
89  +             e.printStackTrace();
90  +             throw new PersistenciaExcpetion(e.getMessage(), e);
91  +         }
92  +         return pessoaDTO;
93  +     }
94  +
95  +     public List<PessoaDTO> filtraPessoa(String nome, Long cpf, String sexo) throws
    PersistenciaExcpetion {
96  +         List<PessoaDTO> lista = new ArrayList<>();
97  +         StringBuilder sqlBuilder = new StringBuilder(SELECT_BY_FILTER_SQL);
98  +
99  +         if (nome != null && !nome.equals("")) {
100 +             sqlBuilder.append(" AND NOME = ?");
101 +         }
102 +
103 +         if (cpf != null) {
104 +             sqlBuilder.append(" AND CPF = ?");
105 +         }
106 +
107 +         if (sexo != null && !sexo.equals("")) {
108 +             sqlBuilder.append(" AND SEXO = ?");
109 +         }
110 +
111 +         try (Connection connection = ConexaoUtil.getInstance().getConnection();
112 +              PreparedStatement statement =
    connection.prepareStatement(sqlBuilder.toString())) {
113 +             int paramIndex = 1;
114 +             if (nome != null && !nome.equals("")) {
115 +                 statement.setString(paramIndex++, nome);
116 +             }
117 +
118 +             if (cpf != null) {
119 +                 statement.setLong(paramIndex++, cpf);
120 +             }
121 +
122 +             if (sexo != null && !sexo.equals("")) {
123 +                 statement.setString(paramIndex, sexo);
124 +             }
125 +
126 +             try (ResultSet resultSet = statement.executeQuery()) {
127 +                 while (resultSet.next()) {
128 +                     PessoaDTO pessoaDTO = getPessoaFromResultSet(resultSet);
```

```
+                    lista.add(pessoaDTO);
+                }
+            }
+        } catch (Exception e) {
+            e.printStackTrace();
+            throw new PersistenciaExcpetion(e.getMessage(), e);
+        }
+        return lista;
+    }
+
+    private void setPessoaStatementValues(PreparedStatement statement, PessoaDTO pessoaDTO) throws Exception {
+        statement.setString(1, pessoaDTO.getNome());
+        statement.setLong(2, pessoaDTO.getCpf());
+        statement.setString(3, pessoaDTO.getEndereco());
+        statement.setString(4, String.valueOf(pessoaDTO.getSexo()));
+        statement.setDate(5, new Date(pessoaDTO.getDtNascimento().getTime()));
+    }
+
+    private PessoaDTO getPessoaFromResultSet(ResultSet resultSet) throws Exception {
+        PessoaDTO pessoaDTO = new PessoaDTO();
+        pessoaDTO.setIdPessoa(resultSet.getInt("id_pessoa"));
+        pessoaDTO.setNome(resultSet.getString("nome"));
+        pessoaDTO.setCpf(resultSet.getLong("cpf"));
+        pessoaDTO.setDtNascimento(resultSet.getDate("dt_nasc"));
+        pessoaDTO.setEndereco(resultSet.getString("endereco"));
+        pessoaDTO.setSexo(resultSet.getString("sexo").charAt(0));
+        return pessoaDTO;
+    }
}
```

0 comments on commit `a9b78f6`

Please sign in to comment.