🖥 **arthurgregorio** / **web-budget**   `Public`

`<> ` **Code**   ⊙ Issues **15**   ⑂ Pull requests **26**   ▶ Actions   ▦ Projects **1**   📖 Wiki   ⓘ Security   ⤴ In:

Commit

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

Removi comentários redundantes e mantive apenas os necessários para f...   **Browse files**

…ornecer informações relevantes sobre a classe e os métodos.

👤 **deniserisia** committed on Jun 17

1 parent `4ff5904`   commit `a06e7d1`

Showing **1 changed file** with **1 addition** and **57 deletions**.   Whitespace | Ignore whitespace   Split | Unified

∨ ⤢ 58 ■■■■□ src/main/java/br/com/webbudget/domain/entities/financial/Apportionment.java ⧉

```
34   34        import static br.com.webbudget.infrastructure.utils.DefaultSchemes.FINANCIAL;
35   35        import static br.com.webbudget.infrastructure.utils.DefaultSchemes.FINANCIAL_AUDIT;
36   36
37      -  /**
38      -   * The representation of an apportionment of a {@link Movement}
39      -   *
40      -   * @author Arthur Gregorio
41      -   *
42      -   * @version 1.0.0
43      -   * @since 1.0.0, 02/02/2015
44      -   */
     37 +
45   38    @Entity
46   39    @Audited
47   40    @ToString(callSuper = true)
77   70        @JoinColumn(name = "id_movement_class", nullable = false)
78   71        private MovementClass movementClass;
79   72
80      -      /**
81      -       * Constructor...
82      -       */
83   73      public Apportionment() {
84   74          this.code = RandomCode.alphanumeric(6);
85   75      }
86   76
87      -      /**
88      -       * Constructor
89      -       *
90      -       * @param value the value of the apportionment
91      -       */
92   77      public Apportionment(BigDecimal value) {
93   78          this();
```

```
 94   79              this.value = value;
 95   80          }
 96      -
 97      -      /**
 98      -       * Constructor...
 99      -       *
100      -       * @param value the value of this apportionment
101      -       * @param movementClass the apportionment {@link MovementClass} we use this to take the {@link
          CostCenter}
102      -       */
103   81          public Apportionment(BigDecimal value, MovementClass movementClass) {
104   82              this();
105   83              this.value = value;
106   84              this.movementClass = movementClass;
107   85              this.costCenter = movementClass.getCostCenter();
108   86          }
109   87
110      -      /**
111      -       * To check if this apportionment is an income
112      -       *
113      -       * @return true if it is a revenue
114      -       */
115   88          public boolean isRevenue() {
116   89              return this.movementClass.isRevenue();
117   90          }
118   91
119      -      /**
120      -       * To check if this apportionment is an expense
121      -       *
122      -       * @return true if it is a expense
123      -       */
124   92          public boolean isExpense() {
125   93              return this.movementClass.isExpense();
126   94          }
127   95
128      -      /**
129      -       * Get the {@link MovementClass} name
130      -       *
131      -       * @return the name if {@link MovementClass} is not null
132      -       */
133   96          public String getMovementClassName() {
134   97              return this.movementClass != null ? this.movementClass.getName() : "";
135   98          }
136      -
137      -      /**
138      -       * Get the {@link CostCenter} name
139      -       *
140      -       * @return the name if the {@link CostCenter} is not null
141      -       */
142   99          public String getCostCenterName() {
143  100              return this.costCenter != null ? this.costCenter.getName() : "";
144  101          }
145  102
146      -      /**
147      -       * Compare if the given apportionment has the same {@link CostCenter} and {@link MovementClass}
148      -       *
149      -       * @param apportionment the instance to be compared
150      -       * @return true if is the same {@link CostCenter} and {@link MovementClass}, false otherwise
151      -       */
```

```
152    103            public boolean isCostCenterAndMovementClassEquals(Apportionment apportionment) {
153    104                return apportionment.getCostCenter().equals(this.costCenter)
154    105                        && apportionment.getMovementClass().equals(this.movementClass);
155    106            }
156          -
157          -        /**
158          -         * Method used to copy an apportionment to another
159          -         *
160          -         * @param toCopy apportionment to be copied
161          -         * @return an entire new apportionment
162          -         */
163    107            public static Apportionment copyOf(Apportionment toCopy){
164    108                return new Apportionment(toCopy.getValue(), toCopy.getMovementClass());
165    109            }
```

0 comments on commit `a06e7d1`

Please sign in to comment.