CORRECTION RUBRIC FOR THE PROJECT OF PROGRAMMING FUNDAMENTALS II (2024/2025)

CRITERIA FOR ASSESSING THE DOCUMENTATION (15% of the final grade)

Positive points awarded for:

- General class diagram design (correct UML).
  - Checking inheritance, association, and dependency relationships.
  - Including arrays to implement association relationships in the relevant classes.
  - Modeling (if needed) abstract methods/classes and interfaces.
- Requirements analysis of the problem.
- User manual.
- Overall quality of the report, which should include:
  - A **cover page** with the names of both team members and the **percentage of contribution of each student**.
  - **A table of contents**, a **requirements analysis**, and **UML diagrams used for design.**
  - Documentation for each class.
  - Appropriate style and presentation format.
- Attendance and defense of Milestone 1.

Negative points awarded for:

- Incorrect class diagram. The practice will automatically fail.
- Mismatch between the proposed design (UML diagram) and the implementation. **This is an important error!**

CRITERIA FOR ASSESSING THE IMPLEMENTATION (25% of the final grade)

Positive points awarded for:

- Correct and error-free functionality. The application runs without issues and delivers expected results.
- Exception handling. Execution errors are captured using exceptions.
- Efficiency and code quality. Efficient code, appropriate use of interfaces, etc.

- Internal code documentation. Comments are included throughout the code.
- Attendance and defense of Milestone 2.

Negative points awarded for:

- Using ArrayList instead of arrays.
- Performing reading and writing operations in classes other than the Main class. **This is an important mistake!!!**
- Poorly modularized Main class code. If the main method is too long and lacks modular methods, this will be penalized.

EXECUTION TESTS

Description and Validation Checklist:

QUERIES

1. **Query 1**: Select option 1 from the menu to verify that the data of all artists is displayed correctly.
    - The output should be properly formatted and not an exact copy of the Artistas.txt file format.
2. **Query 2**: Select option 2 from the menu. Please, verify that the calculation of security service costs is correct.
3. **Query 3**: Select option 3 from the menu. Please, verify that the ticket price for a specific concert attendee is displayed, including:
    - Initial ticket price.
    - Applicable discount percentage.
    - Final price after discount.
    - For:
        - Artist not performing at the festival: Program should notify the user.
        - Non-registered attendee: Program should notify the user.
        - Different attendee conditions (VIP/non-VIP, frequent/non-frequent): Discounts should be applied accordingly.

4. **Query 4**: Select option 4 from the menu. Please, verify the program calculates the cost of attending all headliner artist concerts and buying a T-shirt (merchandise), for a specific attendee.
    - Non-registered attendee: Program should notify the user.
    - Discounts for different attendee conditions should be verified.
5. **Query 5**: Select option 5 from the menu (buying a concert ticket for a specific attendee).
    - Artist not performing at the festival: Program should notify the user.
    - Unconfirmed artist: Program should notify the user.
    - Non-registered attendee: Program should request their information and add them to the system (it is not necessary to include the new attendee in the file).
6. **Query 6**: Select option 6 from the menu. Please, verify all purchased ticket information is displayed for a specific attendee.
    - Non-registered attendee: Program should notify the user.
    - Attendee with no purchased tickets: Program should notify the user.
7. **Query 7**: Select option 7 from the menu. Please, verify all concert information with merchandise stands for which a VIP attendee has purchased tickets.
    - Non-registered attendee: Program should notify the user.
    - Non-VIP attendee: Program should notify the user.
    - Attendee with no purchased tickets: Program should notify the user.
8. **Case Sensitivity**: Ensure the program does not distinguish between uppercase and lowercase for artist names or attendee IDs in queries 3, 4, 5, 6, and 7.

EXCEPTIONS

9. **Exception 1**: Missing files (Artistas.txt or Asistentes.txt). Renaming or deleting these files should:
    - Result in exception capture with an error message displayed on-screen.
    - The program will terminate without recovery.
10. **Exception 2**: Selecting an invalid menu option (e.g., -1 or 12).

    - Please, show an error message and ask for another option until a valid option is selected.

11. **Exception 3**: Entering a letter in the menu instead of a number.

    - The user should receive an error message and will be prompted to type a valid value. Repeat until a valid option is typed.

12. **Exception 4**: Selecting option 5 and the attendee already has 7 tickets.

    - Program should notify the attendee cannot buy more tickets (exception captured) due to exceeding the ticket limit. Please create a new exception to handle this issue.