Karold Lizeth Mejia Orozco
Algoritmos y programación I
Tarea integradora I

Genshin Impact Damage Calculator

1. Description

   Based on Genshin Impact game, we want to create a damage calculator. This program would have a menu that allows the user to select if they want calculate damage, view highest damage, check last 10 results or exit. When user selects first option, the program will distinguish between calculating damage for a single character or a combo. Later on, the program would select one type of reaction, would ask user to input specific information and based on strict formulas would return the final result (or results if we did calculations for multiple characters), while inserting it on two arrays that are useful for calculating the highest damage registered and last 10 results.
   These arrays are going to be used for the other two options, where the user would finally view those results. Finally, the fourth option would allow the user to get out, taking into consideration that the menu is iterative.
2. Inputs:
   - 2.1. Character's attack - Variable: atk
   - 2.2. CRIT damage - Variable: crit
   - 2.3. CRIT rate - Variable: critr
   - 2.4. Elemental mastery - Variable: elm
   - 2.5. Transformative multiplier - Variable: multtra
   - 2.6. Character's level - Variable: lvl
   - 2.7. Enemy's resistance multiplier - Variable: multres
   - 2.8. Reaction multiplier - Variable: multreac
   - 2.9. Additive multiplier - Variable: multadd

   All inputs are double type
3. Output:
   - 3.1. Total additive damage - Variable: totAddDmg (double type)
   - 3.2. Amplified damage - Variable: dmgamp (double type)
   - 3.3. Transformative damage - Variable: dmgtra (double type)
   - 3.4. Last 10 damage records - Array: dmgArray
   - 3.5. Highest damage registered - Array: dmgArray
4. Example: User wants to calculate damage. Selects that he wants to calculate it just for one character and wants it to be transformative. He selects the transformative multiplier, and enter the elemental mastery, character's level and resistance multiplier. Based on the formulas, the program calculate the

result and print it for the user. The program gets back to main menu and user selects to see highest damage calculated, and obtains as a result the only result that it had registered. Program gets back to main menu and finally, user selects to get out. Program asks if it’s sure about that decision and user confirms, so program ends.

5. Flowchart: https://miro.com/app/board/uXjVMt-FAIg=/?share_link_id=68077724452

6. Method contracts

   6.1. dmgCalculus

   - Description: The dmgCalculus method performs calculations based on character's ATK, critical hit chance, and CRIT rate.
   - pre: The method should be called after declaring 'atk', 'crit', and 'critr'.
   - post: Doesn't affect global variables.
   - @param: atk double Must be a double-type number that has been previously initialized
   - @param: crit double Must be a double-type number that has been previously initialized
   - @param: critr double Must be a double-type number that has been previously initialized
   - @return: dmg double The calculated base damage considering ATK, critical hit chance, and CRIT rate.

   6.2. ampCalculus

   - Description: The ampCalculus method calculates amplified damage based on base damage (dmg), elemental mastery, and reaction multiplier.
   - pre: The method should be called after 'dmg' is correctly calculated and 'elm', 'multreac' are initialized with valid numerical values.
   - post: Results are added to arrays and doesn't affect global variables.
   - @param: dmg double Must be a double-type number that has been previously calculated
   - @param: elm double Must be a double-type number that has been previously declared
   - @param: multreac double Must be 1.5 or 2 (double-type number)
   - @return: dmgamp double The calculated amplified damage considering base damage, elemental mastery, and reaction multiplier.

   6.3. addCalculus

   - Description: The addCalculus method calculates additive damage based on base damage (dmg), character level,

elemental mastery, additive multiplier, and enemy's resistance multiplier.
- pre: The method should be called after 'dmg', 'lvl', 'elm', 'multadd', and 'multres' are initialized with valid numerical values.
- post: Results are added to arrays and doesn't affect global variables.
- @param: dmg double Must be a double-type number that has been previously calculated
- @param: lvl double Must be a double-type number that has been previously declared
- @param: elm double Must be a double-type number that has been previously declared
- @param: multadd double Must be 1.15 or 1.25 (double type number)
- @param: multres double Must be randomly generated in a range of 0.5 and 2
- @return: totAddDmg double The calculated additive damage considering base damage, level, elemental mastery, additive multiplier, and resistance multiplier.

6.4. traCalculus

- Description: The traCalculus method calculates transformative damage based on character level, elemental mastery, additive multiplier, and transformative multiplier.
- pre: The method should be called after 'lvl', 'elm', 'multtra', and 'multres' are initialized with valid numerical values.
- post: Results are added to arrays and doesn't affect global variables.
- @param: dmg double Must be a double-type number that has been previously calculated
- @param: lvl double Must be a double-type number that has been previously declared
- @param: elm double Must be a double-type number that has been previously declared
- @param: multtra double Must be 0.25, 0.50, 0.6, 1.2, 1.5, 2.0 or 3.0 (double type number)
- @param: multres double Must be randomly generated in a range of 0.5 and 2
- @return: dmgtra double The calculated transformative damage considering level, elemental mastery, additive multiplier, and transformative multiplier.

6.5. dmgHigh

- Description: The dmgHigh method finds and displays the maximum value in an ArrayList of damages.

- pre: The 'dmgArray' ArrayList must contain valid numerical values.
- post: Does not modify global variables.
- @param: max double Must be a double-type number
- @param: i int Must be a natural number
- @param: dmgArray Must be a predefined ArrayList
- @return: void

6.6. dmgRating

- Description: The dmgRating method displays the latest damage results stored in the 'dmgArray' ArrayList.
- pre: The 'dmgArray' ArrayList must contain valid numerical values.
- post: Does not modify global variables.
- @param: dmgArray Must be a predefined ArrayList
- @param: max double Must be a double-type number
- @param: i int Must be a natural number
- @return: void