

Fix and bugs algo

Certainly! Here's an explanation of the algorithm implemented in the provided code:

1. The code begins by printing a welcome message to the user.
2. The `optionsSelection()` method displays a menu of options to the user and takes their input to perform various operations.
3. The menu options are stored in the `arr` array, and the corresponding option numbers are stored in the `arr1` array.
4. The `expenses` ArrayList is used to store the expenses entered by the user. Initially, it contains some predefined expenses.
5. The `optionsSelection()` method displays the menu options and takes the user's choice as input.
6. The user's choice is then matched with the menu option number in the `switch` statement. The corresponding operation is performed based on the selected option.
 - Option 1: It prints the list of saved expenses.
 - Option 2: It allows the user to add a new expense to the list.
 - Option 3: It prompts the user to confirm the deletion of all expenses and clears the `expenses` list if confirmed.
 - Option 4: It calls the `sortExpenses()` method to sort the expenses in ascending order.
 - Option 5: It calls the `searchExpenses()` method to search for a specific expense.
 - Option 6: It closes the application.
7. The `closeApp()` method is called when the user chooses to close the application. It simply prints a closing message.
8. The `searchExpenses()` method prompts the user to enter an expense value to search for. It uses the `Collections.binarySearch()` method to perform a binary search on the sorted `ArrayList` of expenses and returns the index of the found expense. The result is then printed.

9. The `sortExpenses()` method uses the `Collections.sort()` method to sort the `ArrayList` of expenses in ascending order. The sorted list is then printed.

10. After performing the selected operation, the `optionsSelection()` method is called again to display the menu options and continue the program flow.

11. The program continues to run until the user chooses to close the application.

Overall, the code provides a simple expense management system where the user can review, add, delete, sort, and search for expenses. The sorting is done using the `Collections.sort()` method, and the searching is performed using the `Collections.binarySearch()` method.