* Start the main method.
* Print the welcome message and available options.
* Call the optionsSelection method.
* Inside the optionsSelection method: a. Print the menu options. b. Read the user's choice from the console. c. Use a switch statement to handle the chosen option:
* If the choice is 1, call the reviewExpenses method to display the expenses.
* If the choice is 2, call the addExpense method to add a new expense.
* If the choice is 3, call the deleteExpenses method to delete all expenses.
* If the choice is 4, call the sortExpenses method to sort the expenses.
* If the choice is 5, call the searchExpenses method to search for a specific expense.
* If the choice is 6, call the closeApp method to exit the application.
* If the choice is invalid, print an error message and call optionsSelection again.
* Implement the reviewExpenses method: a. Check if the expenses list is empty. b. If not empty, print the expenses list. c. If empty, print a message indicating no expenses found.
* Implement the addExpense method: a. Prompt the user to enter the value of the expense. b. Read the value from the console. c. Add the value to the expenses list. d. Print a success message and the updated expenses list.
* Implement the deleteExpenses method: a. Check if the expenses list is empty. b. If empty, print a message indicating no expenses to delete. c. If not empty, prompt the user to confirm the deletion. d. Read the confirmation choice from the console. e. If the choice is valid, clear the expenses list and print a success message. f. If the choice is invalid, print an error message.
* Implement the searchExpenses method: a. Check if the expenses list is empty. b. If empty, print a message indicating no expenses to search. c. If not empty, prompt the user to enter the expense to search. d. Read the expense value from the console. e. Check if the expenses list contains the entered expense. f. Print a message indicating whether the expense was found or not.
* Implement the sortExpenses method: a. Check if the expenses list is empty. b. If empty, print a message indicating no expenses to sort. c. If not empty, sort the expenses list in ascending order. d. Print the sorted expenses list.
* Implement the closeApp method: a. Print a closing message.
* End the program.