DESCRIPTION OF THE CODE

Certainly! Let's break down the code into smaller sections and explain each part in detail:

# Date Class:

* The Date class represents a date with day, month, and year attributes.
* It has a constructor that initializes the day, month, and year of the date.
* It provides getter methods (getDay(), getMonth(), getYear()) to retrieve the day, month, and year values.
* The toString() method is overridden to provide a string representation of the date.

# Member Class:

* The Member class represents a member of the golf club with various attributes such as MemberID, LastName, FirstName, etc.
* It has a constructor that initializes all the attributes of a member.
* The display() method prints out all the details of a member.
* Getter methods are provided for retrieving specific attributes (getJoinDate (), getHandicap (), getGender (), getTeam (), getMemberType()).

# Main Class:

* The Main class contains the main method, which is the entry point of the program.
* It creates instances of Date and Member classes to represent members of the golf club.
* The isBefore() method compares two Date objects to check if one date is earlier than the other.
* In the main method, it performs the following tasks:
* Creates instances of Date and Member classes.
* Displays members who joined before a specific date by iterating through the members array and using the isBefore() method.
* Displays senior members with a handicap less than 12 by iterating through the members array and checking the conditions.
* Displays female senior members of TeamB by iterating through the members array and applying the necessary conditions.

# Overall:

The code follows object-oriented principles by encapsulating data and behaviour within classes.

It uses constructors to initialize object attributes and methods to perform specific tasks.

The main method orchestrates the execution of the program by creating objects, performing operations, and displaying results.

This code demonstrates the use of classes, objects, constructors, methods, and control structures to manage and process data related to golf club members.

# An ULM (Unified Modelling Language) DESIGN FO THIS CODE

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| <<Class>> |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

| Date |

|-------------------------------------|

| - day: int |

| - month: String |

| - year: int |

|--------------------------------------|

| + Date(int, String, int) |

| + getDay(): int |

| + getMonth(): String |

| + getYear(): int |

| + toString(): String |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| <<Class>> |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

| Member |

|----------------------- |

| - MemberID: int |

| - LastName: String |

| - FirstName: String |

| - Handicap: int |

| - Gender: char |

| - Team: String |

| - MemberType: String |

| - Coach: int |

| - Phone: long |

| - JoinDate: Date |

|--------------------------------------|

| + Member(int, String, String, int, char, String, String, int, long, Date) |

| + display(): void |

| + getJoinDate(): Date |

| + getHandicap(): int |

| + getGender(): char |

| + getTeam(): String |

| + getMemberType(): String |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

| <<Class>> |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

| Main |

|--------------------------------------|

| + main(String[]): void |

| - isBefore(Date, Date): boolean |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

# OUTPUT OF THE CODE

