```
package com.company;

import java.io.*;
import java.nio.file.Files;
import java.nio.file.Path;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.LocalTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

public class GymFile {




    private List<String> betalandeKunder = new ArrayList<>();
    private LocalDate withinYear = LocalDate.now();
    private LocalDate transactionDate;

    public List<String> getBetalandeKunder() {
        return betalandeKunder;
    }

    public boolean checkCustomer(String nameOrPersNr) {

        String svar = ""; boolean svaret = true;
        try (BufferedReader reader = new BufferedReader(new
FileReader("src/com/company/Customers.txt"));){
            String person = ""; String persnr = "";
            String read; boolean foundWithinYear = false; boolean
foundOutsideYear = false; boolean notFound = false; withinYear =
withinYear.minusYears(1);
            while ((read = reader.readLine()) != null) {
                List<String> currentLine = List.of(read.split(","));
                transactionDate = LocalDate.parse(reader.readLine());
                if (currentLine.size() == 2) {


                    if (currentLine.get(0).equals(nameOrPersNr) ||
currentLine.get(1).strip().equals(nameOrPersNr)) {



                        if (withinYear.isBefore(transactionDate)) {
                            foundWithinYear = true;
                            person = currentLine.get(1).strip();
                            persnr = currentLine.get(0);
                            betalandeKunder.add(persnr + "," + person);
                            //betalandeKunder.add("Person: " + person + "\n" +
```

```
                                                //"Medlemsdatum: " +
transactionDate);

                        } else {
                            foundOutsideYear = true;
                            person = currentLine.get(1);

                        }
                    } else {
                        if (withinYear.isBefore(transactionDate)) {
                            betalandeKunder.add(currentLine.get(0) + "," +
currentLine.get(1).strip());
                        }
                    }
                }
            }

            if (foundWithinYear) {
                svaret = foundWithinYear;
                System.out.println(person + " är en aktiv kund");
            } else if (foundOutsideYear) {
                svaret = foundOutsideYear;
                System.out.println(person + " är en före detta medlem");
            } else {
                notFound = true;

                if (notFound) {
                    svaret = notFound;
                    System.out.println("Ny kund");
                }
            }

        } catch (FileNotFoundException e) {
            System.out.println("Filen kunde inte hittas");
        } catch (IOException e) {
            e.printStackTrace();
        }


        return svaret;
    }


    public void harTränat(String namnEllerPers) {

        Writer out;
        try (BufferedWriter writer = new BufferedWriter(new
FileWriter("src/com/company/TränatIdag.txt", true));) {


            if (!Files.exists(Path.of("src/com/company/TränatIdag.txt"))) {
                Files.createFile(Path.of("src/com/company/TränatIdag.txt"));
            } else {
                LocalTime time = LocalTime.now();
```

```
                DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("HH:mm");
                for (String namn : betalandeKunder) {
                    String[] namnSplit = namn.split(",");
                    if (namnSplit[0].equals(namnEllerPers) ||
namnSplit[1].equals(namnEllerPers)) {

                        writer.write(namnSplit[0] + "," + namnSplit[1] + "
tränade den " + LocalDate.now() + " " + time.format(formatter));
                        writer.newLine();
                    }
                }
            }



        } catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println(betalandeKunder);

    }
}
```