/\*\*

\* Represents a loyalty card for a fictional university

\* bookshop.

\*

\* Amended for inheritency and abstract.

\*

\* @author (Grant Allenby w18013678)

\* @version (4/3/19)

\*/

public abstract class LoyaltyCard

{

private Name name;

private String cardNumber;

private int points;

/\*\* Constructor for objects of class LoyaltyCard where they obtain

\* a card without purchasing any book(s) at that point.

\* The number of points / spend should be set to 0.

\* @param title - The Customer's title.

\* @param firstName - The Customer's first name.

\* @param lastName - The Customer's last name.

\* @param cardNumber - The number for this card.

\*/

public LoyaltyCard(String title, String firstName, String lastName, String cardNumber)

{

name = new Name(title, firstName, lastName);

this.cardNumber = cardNumber;

points = 0;

}

/\*\* Constructor for objects of class Customer where they purchase

\* a book(s) at the same time as they obtain their card.

\* @param title - The Customer's title.

\* @param firstName - The Customer's first name .

\* @param lastName - The Customer's last name.

\* @param cardNumber - The number for this card.

\* @param points - The number of points earned. Can be 0.

\*/

public LoyaltyCard(String title, String firstName, String lastName,

String cardNumber, int points)

{

name = new Name(title, firstName, lastName);

this.cardNumber = cardNumber;

this.points = points;

}

/\*\* Get the Customer's card Number.

\* @return cardNumber - The Customer's card number.

\*/

public String getCardNumber()

{

return cardNumber;

}

/\*\* Get the Customer's full name.

\* @return toString() - The Customer's full name.

\*/

public String getFullName()

{

return toString();

}

/\*\* Get the number of points earned.

\* @return points - The number of points available.

\*/

public double getPoints()

{

return points;

}

/\*\* Record a new title.

\* @param title - The revised Customer's title.

\*/

public void setTitle(String title)

{

name.setTitle(title);

}

/\*\* Record a new firstName.

\* @param firstName - The revised Customer's first name.

\*/

public void setFirstName(String firstName)

{

name.setFirstName(firstName);

}

/\*\* Record a new last name.

\* @param lastName - The revised Customer's last name.

\*/

public void setLastName(String lastName)

{

name.setLastName(lastName);

}

/\*\* Record a new Customer spend and add it to the current amount.

\* @param numberOfPoints - Amount of points to add to the Customer's existing points.

\*/

public void addPoints(int numberOfPoints)

{

points = points + numberOfPoints;

System.out.println("Points now: " + points);

}

/\*\* Record spending the Customer's points and subtract from the Customer's current points.

\* If points are less than 0 as a consequence of the spend, they will not be reduced and and error

\* will display.

\* @param numberOfPoints - Amount of points to subtract from the Customer's existing points.

\*/

public void reducePoints(int numberOfPoints)

{

if (points - numberOfPoints < 0)

{

System.out.println("Transaction refused, only " + points + " points available.");

}

else

{

points = points - numberOfPoints;

System.out.println("Points now: " + points);

}

}

/\*\* Output the Customer's details.

\* @param cardNumber - The Customer's card number.

\* @param points - The Customer's points.

\*/

public void printHolderDetails()

{

System.out.println(toString()

+ "\nCard Number: " + cardNumber

+ "\nPoints available: " + points);

}

/\*\* Returns the name, card number and points of the Customer.

\* @return name - The Customer's name.

\* @return cardNumber - The Customer's card number.

\* @return points - The Customer's points.

\*/

public String toString()

{

return name + "\nCard Number: " + cardNumber + "\nPoints available: " + points;

}

} // End of class LoyaltyCard.

/\*\* Represents a Staff loyalty card for a fictional university

\* bookshop.

\* @author (Grant Allenby w18013678)

\* @version (4/3/19)

\*/

public class StaffLoyaltyCard extends LoyaltyCard

{

private String location;

private String staffNumber;

/\*\* Constructor for a staff loyalty card. this constructor assigns 0 points.

\* @param title - The Staff Member's title.

\* @param firstName - The Staff Member's first name.

\* @param lastName - The Staff Member's last name.

\* @param cardNumber - The Staff Member's card number.

\* @param location - The Staff Member's location.

\* @param staffNumber - The Staff Member's staff number.

\*/

public StaffLoyaltyCard(String title, String firstName, String lastName, String cardNumber,

String staffNumber, String location)

{

super(title, firstName, lastName, cardNumber);

this.staffNumber = staffNumber;

this.location = location;

}

/\*\* Constructor for a staff loyalty card. this constructor assigns 0 points.

\* @param title - The Staff Member's title.

\* @param firstName - The Staff Member's first name.

\* @param lastName - The Staff Member's last name.

\* @param cardNumber - The Staff Member's card number.

\* @param location - The Staff Member's location.

\* @param staffNumber - The Staff Member's staff number.

\* @param points - The number of points on the Staff Member's card. This can be 0.

\*/

public StaffLoyaltyCard(String title, String firstName, String lastName, String cardNumber,

String staffNumber, String location, int points)

{

super(title, firstName, lastName, cardNumber, points);

this.staffNumber = staffNumber;

this.location = location;

}

/\*\* Method for returning the location of the Staff Member.

\* @return location - the location of the Staff Member.

\*/

public String getLocation()

{

return location;

}

/\*\* Method for returning the Staff Member's staff number.

\* @return staffNumber - the staff number of the Staff Member.

\*/

public String getStaffNumber()

{

return staffNumber;

}

/\*\* Prints the Staff Member's name, card number and location.

\*/

public void print()

{

System.out.println(toString());

}

/\*\* Allows the change of location of the Staff Member.

\* @param location - The location of the Staff Member.

\*/

public void setLocation(String location)

{

this.location = location;

}

/\*\* Returns the Staff Member's name, card number and location.

\* @return super.toString() - The Staff Member's name and card number.

\* @return staffNumber - The Staff Member's staff number.

\* @return location - The Staff Member's location.

\*/

public String toString()

{

return super.toString() + "\nStaff Number: " + staffNumber + "\nLocation: " + location;

}

} // End of class StaffLoyaltyCard

/\*\* Represents a Student loyalty card for a fictional university

\* bookshop.

\* @author (Grant Allenby w18013678)

\* @version (4/3/19)

\*/

public class StudentLoyaltyCard extends LoyaltyCard

{

private String studentID;

private Address termAddress;

/\*\* Constructor for a student loyalty card. This constructor assigns 0 points.

\* @param title - The Student's title.

\* @param firstName - The Student's first name.

\* @param lastName - The Student's last name.

\* @param cardNumber - The Student's card number.

\* @param street - The Student's street.

\* @param town - The Student's town.

\* @param postcode - The Student's postcode.

\* @param studentID - The Student's ID.

\*/

public StudentLoyaltyCard(String title, String firstName, String lastName, String cardNumber,

String street, String town, String postcode, String studentID)

{

super(title, firstName, lastName, cardNumber);

termAddress = new Address(street, town, postcode);

this.studentID = studentID;

}

/\*\* Constructor for a student loyalty card. This constructor includes points. The points may be 0.

\* @param title - The Student's title.

\* @param firstName - The Student's first name.

\* @param lastName - The Student's last name.

\* @param cardNumber - The Student's card number.

\* @param street - The Student's street.

\* @param town - The Student's town.

\* @param postcode - The Student's postcode.

\* @param studentID - The Student's ID.

\* @param points - The Student's points.

\*/

public StudentLoyaltyCard(String title, String firstName, String lastName, String cardNumber,

String street, String town, String postcode, String studentID, int points)

{

super(title, firstName, lastName, cardNumber, points);

termAddress = new Address(street, town, postcode);

this.studentID = studentID;

}

/\*\* Amends the Student's term address.

\* @param street - The Student's street.

\* @param town - The Student's town.

\* @param postcode - The Student's postcode.

\*/

public void amendTermAddress(String street, String town, String postcode)

{

termAddress = new Address(street, town, postcode);

}

/\*\* Returns the Student's ID.

\* @return studentID - The Student's ID.

\*/

public String getID()

{

return studentID;

}

/\*\* Returns the Student's term address.

\* @return termAddress.toString() - The Student's term address.

\*/

public String getTermAddress()

{

return termAddress.toString();

}

/\*\* Prints the Student's name, card number, student id, and term address.

\*/

public void print()

{

System.out.println(toString());

}

/\*\* Returns the Student's name, card number, student ID and term address.

\* @return super.toString() - The Student's name and card number.

\* @return studentID - The Student's ID.

\* @return termAddress - The Student's term address.

\*/

public String toString()

{

return super.toString() + "\nStudent ID: " + studentID + "\n" + termAddress;

}

} // End of class StudentLoyaltyCard

import java.util.\*;

/\*\* A list comprising of all loyalty cards for LoyaltyCard, StaffLoyaltyCard and StudentLoyaltyCard.

\* @author (Grant Allenby w18013678)

\* @version (4/3/19)

\*/

public class LoyaltyCardList

{

private ArrayList<LoyaltyCard> loyaltyCards;

/\*\* Constructor for objects of class LoyaltyCardList

\*/

public LoyaltyCardList()

{

loyaltyCards = new ArrayList<LoyaltyCard>();

}

/\*\* Method that allows for adding of a loyalty card to the loyalty card array list.

\*/

public void addLoyaltyCard(LoyaltyCard loyaltyCard)

{

loyaltyCards.add(loyaltyCard);

}

/\*\* Returns the number of loyalty cards.

\* @return - The number of loyalty cards.

\*/

public int getNumberOfLoyaltyCards()

{

return loyaltyCards.size();

}

/\*\* Prints all loyalty cards.

\*/

public void getAllLoyaltyCards()

{

for(LoyaltyCard loyaltyCard : loyaltyCards)

{

loyaltyCard.printHolderDetails();

System.out.println();

}

}

/\*\* Method that finds a loyalty card by its specific card number. returns true if found.

\* @param cardNumber - the card number to be found.

\* @return true - Card is found matching the card number.

\* @return false - Card is not found matching the card number.

\*/

public boolean getLoyaltyCard(String cardNumber)

{

int index = 0;

for (LoyaltyCard loyaltyCard: loyaltyCards)

{

if (cardNumber.equals(loyaltyCard.getCardNumber()))

{

loyaltyCards.get(index);

return true;

}

}

return false;

}

/\*\* Method that removes a loyalty card from the array. Prints if entry is negative or non-existant.

\* @param loyaltyCardEntry - The number of the loyalty card in the array.

\*/

public void removeLoyaltyCard(int loyaltyCardEntry)

{

if(loyaltyCardEntry < 0)

{

System.out.println("Negative entry :" + loyaltyCardEntry);

}

else if(loyaltyCardEntry < getNumberOfLoyaltyCards())

{

loyaltyCards.remove(loyaltyCardEntry);

}

else

{

System.out.println("No such entry :" + loyaltyCardEntry);

}

}

/\*\* Method that removes a loyalty card from the array. Returns true if successful.

\* @param cardNumber - The card number to be entered for removal.

\* @return true - Card matching the card number is removed from the array successfully.

\* @return false - Card has failed to be found matching the card number.

\*/

public boolean removeLoyaltyCard(String cardNumber)

{

int index = 0;

for (LoyaltyCard loyaltyCard: loyaltyCards)

{

if (cardNumber.equals(loyaltyCard.getCardNumber()))

{

loyaltyCards.remove(index);

return true;

}

index++;

}

return false;

}

/\*\* Searches for a loyalty card based on the card Number.

\* @param cardNumber - The card number to be found.

\* @return index - The entry for the card number entered.

\* @return -1 - Search has failed.

\*/

public int search(String cardNumber)

{

int index = 0;

for (LoyaltyCard loyaltyCard : loyaltyCards)

{

if (cardNumber.equals(loyaltyCard.getCardNumber()))

{

return index;

}

else

{

index++;

}

}

return -1;

}

/\*\* Method for returning the number of Student loyalty cards.

\* @return index - The number of Student loyalty cards.

\*/

public int getNumberOfStudentLoyaltyCards()

{

int index = 0;

for (LoyaltyCard loyaltyCard: loyaltyCards)

{

if (loyaltyCard instanceof StudentLoyaltyCard)

{

index++;

}

}

return index;

}

/\*\* Method for returning the number of Staff loyalty cards.

\* @return index - The number of Staff loyalty cards.

\*/

public int getNumberOfStaffLoyaltyCards()

{

int index = 0;

for (LoyaltyCard loyaltyCard: loyaltyCards)

{

if (loyaltyCard instanceof StaffLoyaltyCard)

{

index++;

}

}

return index;

}

} // End of class LoyaltyCardList