Лабораторная работа №1

**Простейшие классы и объекты. Пакеты.**

**Разработать классы для описанных ниже объектов. Включить в класс методы setX (…) – задает значение поля X объекта, getX(…) – возвращает значение поля X, show (…) – вывод на экран значений полей объекта. Определить для класса свои методы, позволяющие решить предложенные задачи.**

**Класс поместить в пакет. Обработать массив объектов, подключив к основной программе созданный пакет.**

Вариант №10

**10.Customer:** Фамилия, Имя, Отчество, Номер кредитной карточки, Номер банковского счета.   
Создать массив объектов. Вывести:   
a) список покупателей в алфавитном порядке;   
b) список покупателей, у которых номер кредитной карточки находится в заданном интервале.

Код:

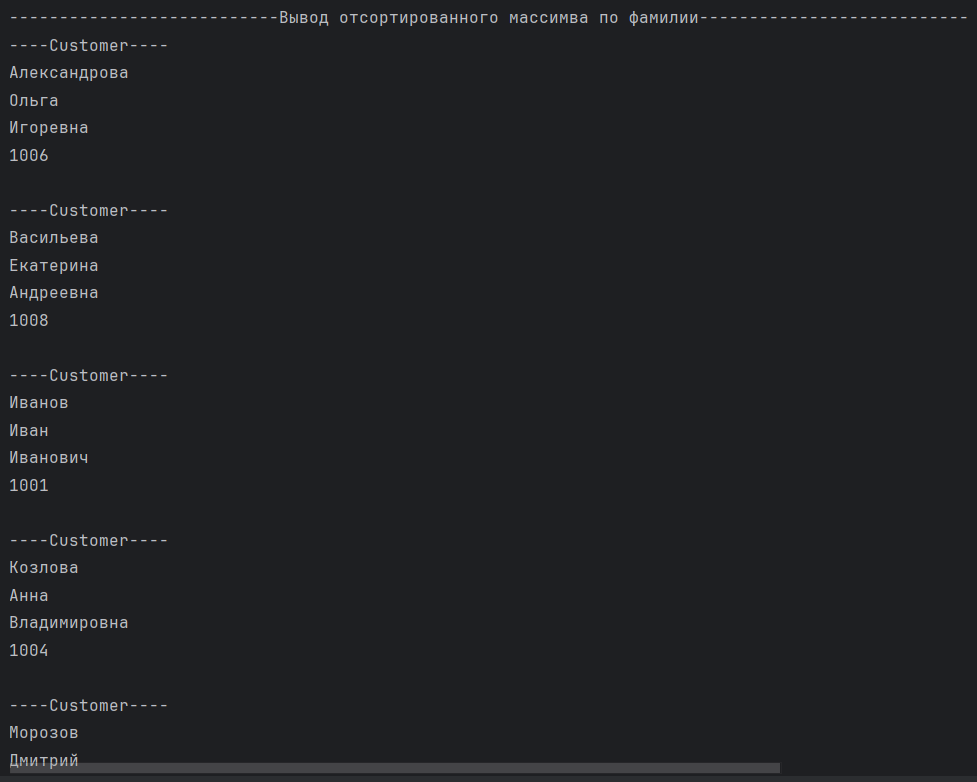
Main.java

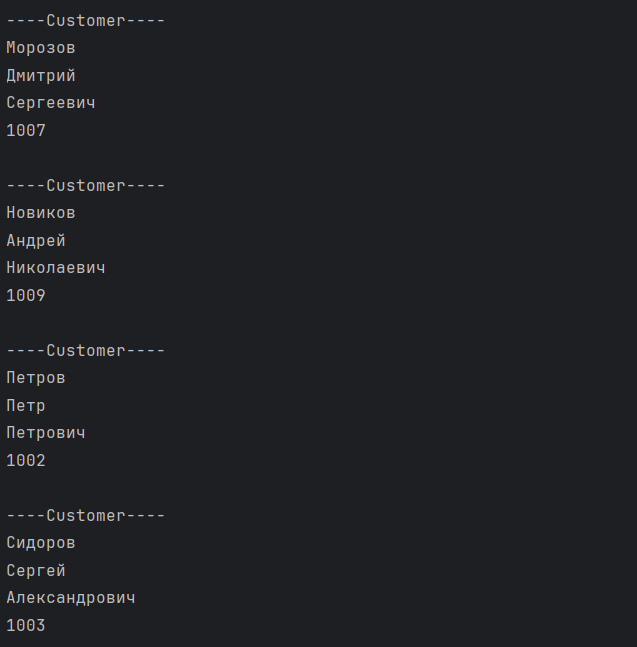
import Customer.Customer;  
  
import java.util.Arrays;  
  
public class Main {  
 public static void main(String[] args) {  
 Customer[] customers = new Customer[10];  
 customers[0] = new Customer("Иванов", "Иван", "Иванович", 1001);  
 customers[1] = new Customer("Петров", "Петр", "Петрович", 1002);  
 customers[2] = new Customer("Сидоров", "Сергей", "Александрович", 1003);  
 customers[3] = new Customer("Козлова", "Анна", "Владимировна", 1004);  
 customers[4] = new Customer("Смирнов", "Алексей", "Дмитриевич", 1005);  
 customers[5] = new Customer("Александрова", "Ольга", "Игоревна", 1006);  
 customers[6] = new Customer("Морозов", "Дмитрий", "Сергеевич", 1007);  
 customers[7] = new Customer("Васильева", "Екатерина", "Андреевна", 1008);  
 customers[8] = new Customer("Новиков", "Андрей", "Николаевич", 1009);  
 customers[9] = new Customer("Федорова", "Мария", "Викторовна", 1010);  
 //sortCustomer(customers);  
 Arrays.*sort*(customers);  
 System.*out*.println("---------------------------Вывод отсортированного массимва по фамилии---------------------------");  
 for (var customer : customers) {  
 System.*out*.println(customer.show());  
 }  
  
 System.*out*.println("---------------------------Вывод покупателе, входящие в интервал---------------------------");  
 int from = 1000;  
 int to = 1004;  
 for (var customer : customers) {  
 if(customer.isInInterval(from, to)) {  
 System.*out*.println(customer.show());  
 }  
 }  
 }  
  
 public static void sortCustomer(Customer[] customers) {  
 int n = customers.length;  
 boolean swapped;  
 for (int i = 0; i < n - 1; i++) {  
 swapped = false;  
 for (int j = 0; j < n - 1 - i; j++) {  
 if (customers[j].isLastNameMore(customers[j + 1])) {  
 Customer temp = customers[j];  
 customers[j] = customers[j + 1];  
 customers[j + 1] = temp;  
 swapped = true;  
 }  
 }  
 if (!swapped) break;  
 }  
 }  
}

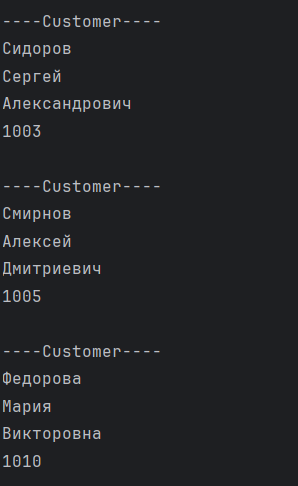
Customer.java

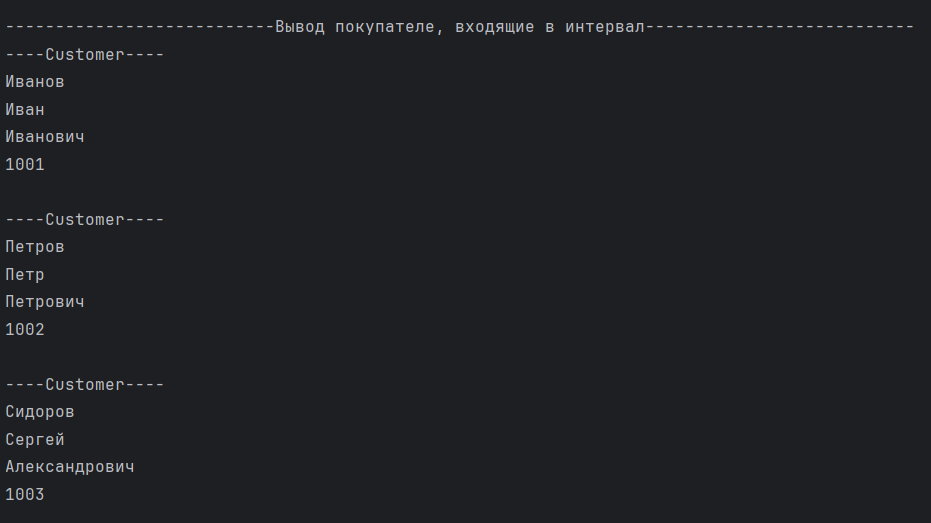
package Customer;  
  
import java.util.Comparator;  
  
public class Customer implements Comparable<Customer> {  
 private String lastName;  
 private String firstName;  
 private String middleName;  
 private int cardNumber;  
  
 public Customer(String lastName, String firstName, String middleName, int cardNumber) {  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.middleName = middleName;  
 this.cardNumber = cardNumber;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public String getMiddleName() {  
 return middleName;  
 }  
  
 public void setMiddleName(String middleName) {  
 this.middleName = middleName;  
 }  
  
 public int getCardNumber() {  
 return cardNumber;  
 }  
  
 public void setCardNumber(int cardNumber) {  
 this.cardNumber = cardNumber;  
 }  
  
 public boolean isFirstNameMore(Customer customer) {  
 return this.firstName.compareTo(customer.firstName) > 0;  
 }  
  
 public boolean isLastNameMore(Customer customer) {  
 return this.lastName.compareTo(customer.lastName) > 0;  
 }  
  
 public boolean isMiddleNameMore(Customer customer) {  
 return this.middleName.compareTo(customer.middleName) > 0;  
 }  
  
 public boolean isInInterval(int from, int to) {  
 return this.cardNumber > from && this.cardNumber < to;  
 }  
  
 public String show() {  
 return "----Customer----" +  
 '\n' +  
 getLastName() +  
 '\n' +  
 getFirstName() +  
 '\n' +  
 getMiddleName() +  
 '\n' +  
 getCardNumber() +  
 '\n';  
 }  
  
 @Override  
 public int compareTo(Customer o) {  
 return this.lastName.compareTo(o.lastName);  
 }  
}

Вывод программа:









Описание

Класс Customer содержит в себе поля, описанные в задачи.

Для кадого поля есть геттер и сеттер:

У поля lastName есть getLastName и setLastName

У поля firstName есть getFirstName и setFirstName

У поля middleName есть getMiddleName и setMiddleName

У поля cardNumber есть getCardNumber и setCardNumber

show – вывод всех полей класса

Так же класс имеет 4 логических метода:

isFirstNameMore – метод сравнивает экзмепляры класса по полю firstName и возращает true или false в зависимости от значений у полей.

isLastNameMore – метод сравнивает экзмепляры класса по полю lastName и возращает true или false в зависимости от значений у полей.

isMiddleNameMore – метод сравнивает экзмепляры класса по полю middleName и возращает true или false в зависимости от значений у полей.

isInInterval – метод принимает два числа и проверяет входит ли в этот диапозон поле cardName. Возвращает true, если входит и false, если нет.