

Person.java

```
import java.util.ArrayList;
import java.util.List;

public class Person {
    private String name;
    private int money;
    private List<Product> products;

    public Person (String name, int money) {
        if (name.trim().equals("")) {
            System.out.println("Имя не может быть пустым.");
            System.exit(0);
        } else if (name.trim().length() < 3) {
            System.out.println("Имя не может быть короче 3 символов.");
            System.exit(0);
        } else if (money < 0) {
            System.out.println("Деньги не могут быть отрицательными.");
            System.exit(0);
        } else {
            this.name = name;
            this.money = money;
            this.products = new ArrayList<>();
        }
    }

    public String getName() {
        return name;
    }

    public int getMoney() {
        return money;
    }

    public List<Product> getProducts() {
        return products;
    }
}
```

```
public void buy(Product product) {
    int price = product.getCost();

    if (money >= price) {
        money -= price;
        products.add(product);
        System.out.println(name + " купил " + product.getName());
    } else {
        System.out.println(name + " не может позволить себе " + product.getName());
    }
}

@Override
public String toString() {
    return name;
}

@Override
public int hashCode() {
    return name.hashCode() + money;
}

@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (!(obj instanceof Person)) return false;
    Person other = (Person) obj;
    return this.name.equals(other.name) && this.money == other.money;
}
}
```

Product.java

```
public class Product {
    private String name;
    private int cost;

    public Product (String name, int cost) {
        if (name.trim().equals("")) {
            System.out.println("Имя не может быть пустым.");
            System.exit(0);
        } else if (name.trim().length() < 3) {
            System.out.println("Имя не может быть короче 3 символов.");
            System.exit(0);
        } else if (cost < 0) {
            System.out.println("Деньги не могут быть отрицательными.");
            System.exit(0);
        } else {
            this.name = name;
            this.cost = cost;
        }
    }

    public String getName() {
        return name;
    }

    public int getCost() {
        return cost;
    }

    public String toString() {
        return name;
    }

    public int hashCode() {
```

```

        return name.hashCode() + cost;
    }

    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (!(obj instanceof Product)) return false;
        Product other = (Product) obj;
        return this.name.equals(other.name) && this.cost == other.cost;
    }
}

```

App.java

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class App {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Person> persons = new ArrayList<>();
        List<Product> products = new ArrayList<>();

        System.out.println("Введите количество покупателей: ");
        int peopleAmount = Integer.parseInt(scanner.nextLine().trim());

        for (int i = 0; i < peopleAmount; i++) {
            System.out.println("Введите Имя покупателя №" + (i + 1) + " и сумму денег в его кошельке (формат: Имя=Сумма):");

            String[] peopleDB = scanner.nextLine().trim().split("=");
            if (peopleDB.length != 2) {
                System.out.println("Ошибка! Формат: Имя пользователя=Сумма (например: Борис=10)");
                i--;
            }
        }
    }
}

```

```

        continue;
    } else {
        int money = Integer.parseInt(peopleDB[1].trim());
        Person person = new Person(peopleDB[0].trim(), money);
        persons.add(person);
    }
}

System.out.println("Введите количество продуктов:");
int productAmount = Integer.parseInt(scanner.nextLine().trim());

for (int i = 0; i < productAmount; i++) {
    System.out.println("Введите название и цену продукта №" + (i+1) + " (формат: Название=Цена): ");
    String[] productDB = scanner.nextLine().trim().split("=");
    if (productDB.length != 2) {
        System.out.println("Ошибка! Формат: Название=Цена (например: Торт=1000)");
        i--;
        continue;
    } else {
        int priceProduct = Integer.parseInt(productDB[1].trim());
        Product product = new Product(productDB[0].trim(), priceProduct);
        products.add(product);
    }
}

System.out.print("Доступные продукты: ");
for (int i = 0; i < products.size(); i++) {
    if (i > 0) System.out.print(", ");
    System.out.print(products.get(i).getName());
}

System.out.println();
boolean buying = true;
while (buying) {
    for (Person person : persons) {
        System.out.println(person.getName() + ", введите название продукта для покупки (или END для завершения):");
        String input = scanner.nextLine().trim();

        if (input.equalsIgnoreCase("END")) {
            buying = false;
            break;
        }
    }
}

```

```
Product selected = null;
for (Product NameProduct : products) {
    if (NameProduct.getName().equalsIgnoreCase(input)) {
        selected = NameProduct;
        break;
    }
}

if (selected != null) {
    person.buy(selected);
} else {
    System.out.println("Продукт отсутствует в списке заявленных.");
}
}
```

```
System.out.println();
```

```
for (Person person : persons) {
    if (person.getProducts().isEmpty()) {
        System.out.println(person.getName() + " - Ничего не куплено");
    } else {
        StringBuilder estr = new StringBuilder();
        for (Product product : person.getProducts()) {
            if (estr.length() > 0) {
                estr.append(", ");
            }
            estr.append(product.getName());
        }
        System.out.println(person.getName() + " - " + estr.toString());
    }
}
scanner.close();
}
```

Скриншоты с результатами







