Primeiramente me debrucei em um emaranhado de dúvidas, Java é uma das coisas que mais odeio, perdendo apenas para mim mesmo. Não entendo por que pra fazer um print eu tenho que usar System.out.println ao invés de apenas print. Enfim, a partir do entendimento do problema, montei uma simples(pra não dizer preguiçosa) página no aplicativo Notion onde destaquei as principais demandas do projeto.

Add cover Add comment

Pagamentos Java

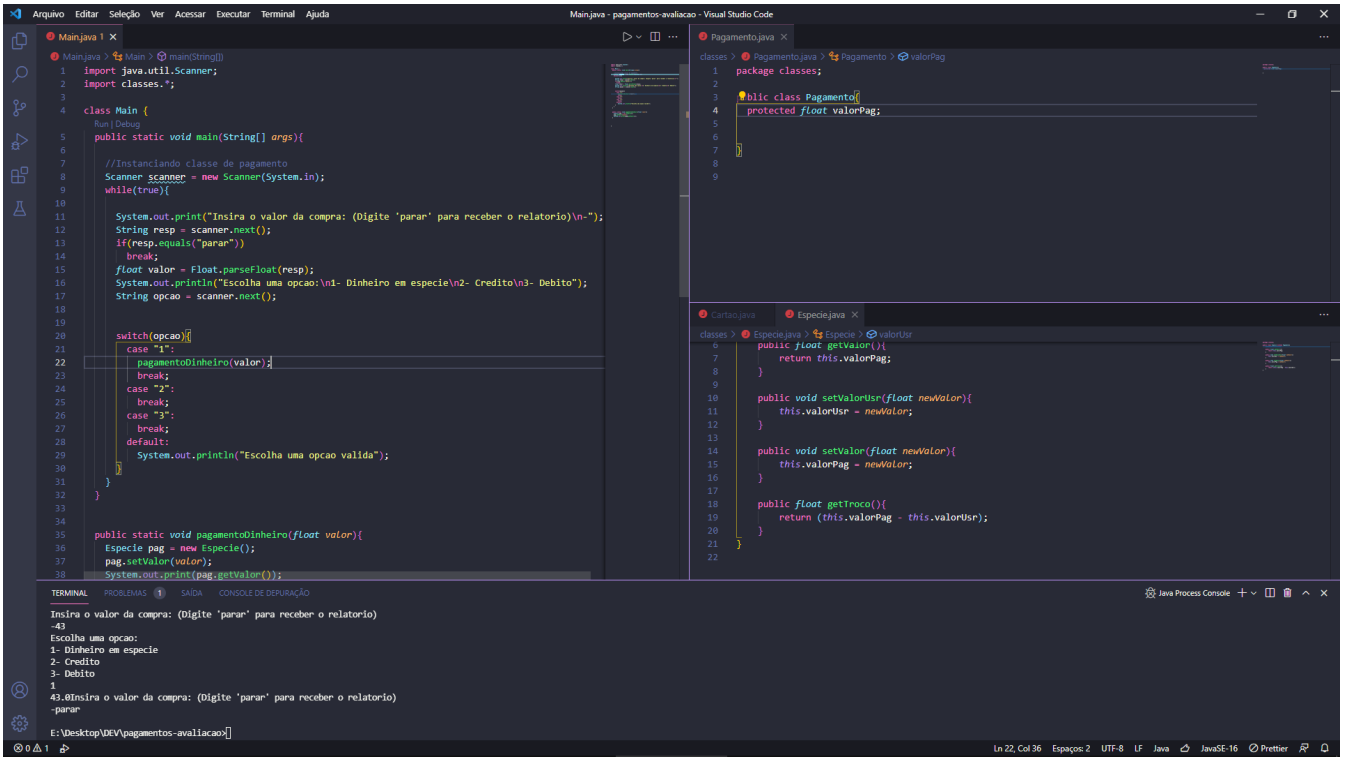
- [ ] dinheiro em espécie
- [ ] Fazer operação de troco
- ▼ dinheiro em cartão a vista
  - [ ] parcelamento
  - [ ] retornar valor da parcela
- [ ] cartão no debito

Pagamentos recebidos utilizando a função crédito à vista ou débito deverão ser tratados como uma operação equivalente (são iguais)

Conclusão

- [ ] Quantidade total de operações registradas
- [ ] Quantidade de operações em dinheiro, no cartão à vista (débito ou crédito, tanto faz) e a quantidade de operações parceladas
- [ ] Valor total recebido em todos os pagamentos

Agora com tudo mais claro, comecei a fazer o código pelo replit, mas estava um tanto quanto lento então coloquei em minha máquina local. Infelizmente, descobri apenas agora (datetime.date.now()) às 17:25 que devo fazer um roteiro e não um relatório, ou seja, estou escrevendo este documento enquanto falta apenas 1 hora e 35 minutos para o fim do prazo. Quando descobri isso, esse era o estado atual do projeto:



Nesse estado, eu já tinha a classe “mãe” Pagamento e suas “filhas” Cartao e Espécie, referentes ao métodos de pagamento. Já no arquivo Main.java, acontece a obtenção do valor de compra e o método escolhido pelo usuário. Um switch direciona cada opção para a função correta.

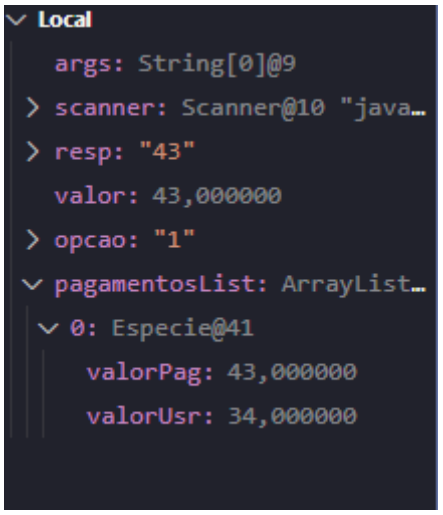
Minha ideia era colocar tudo em uma lista no final, então usei um ArrayList para armazenar todas as classes derivadas de Pagamento().

```
    String opcao = scanner.next();
    List<Pagamento> pagamentosList = new ArrayList<Pagamento>();

    switch(opcao){
      case "1":
        pagamentosList.add(pagamentoDinheiro(valor));
        break;
      case "2":
        break;
      case "3":
        break;
      default:
        System.out.println("Escolha uma opcao valida");
    }
  }
}


public static Especie pagamentoDinheiro(float valor){
  Especie pag = new Especie();
  pag.setValor(valor);
  System.out.print(pag.getValor());
  return pag;
}
```

Primeiro teste:



Dentro de pagamentosList, o objeto que está na posição 0 está uma classe do tipo Espécie. Tudo certo por aqui… Só que não :(.

Tive que criar uma lista pra cada tipo de objeto pois nao consegui usar as funções a partir do método get da ArrayList.

```
float valor = Float.parseFloat(resp); valor = 40,000000, resp = "40"
System.out.println("Escolha uma opcao:\n1- Dinheiro em especie\n2- Credito\n3-
String opcao = scanner.next(); opcao = "1", scanner = Scanner@10 "java.util.S
List<Especie> especieList = new ArrayList<Especie>();  especieList = ArrayLis
List<Cartao> cartaoList = new ArrayList<Cartao>();  cartaoList = ArrayList@14
List<CreditoeDebito> cartaoDebitoList = new ArrayList<CreditoeDebito>();  car

switch(opcao){ opcao = "1"
  case "1":
    especieList.add(pagamentoDinheiro(valor, scanner)); especieList = ArrayLi
    System.out.print(especieList.get(0).getTroco()); especieList = ArrayList@
    break;
  case "2":
```

Seguindo a ordem natural das coisas, a implantação dos outros tipos de pagamentos foi bastante tranquila, tem um if que quando o valor de prestações for igual a 1 ele redireciona para a função de creditoDebito.

```
    switch(opcao){
      case "1":
        especieList.add(pagDinheiro(valor, scanner));
        System.out.print(especieList.get(0).getTroco());
        break;
      case "2":
        cartaoList.add(pagCartao(valor, scanner));
        System.out.print(cartaoList.get(0).getValorPrest());
        break;
      case "3":
        break;
      default:
        System.out.println("Escolha uma opcao valida");
    }
  }
}
```

```
public static Especie pagDinheiro(float valor, Scanner scanner){
  Especie pag = new Especie();
  pag.setValor(valor);
  System.out.print("Insira o valor pago pelo usuário-");
  float valorUsr = scanner.nextFloat();
  pag.setValorUsr(valorUsr);
  return pag;
}

public static Cartao pagCartao(float valor, Scanner scanner){
  Cartao pag = new Cartao();
  pag.setValor(valor);
  System.out.print("Insira o numero de prestacoes -");
  int numPrest = scanner.nextInt();

  if(numPrest==1){
    pagCreditoDebito(valor, scanner);
  }
  pag.setPrest(numPrest);
  return pag;
}

public static CreditoeDebito pagCreditoDebito(float valor, Scanner scanner){
```

Lembra quando eu falei que estava indo tudo certo? Então, nem tudo… Após passar por um *Neuron Activation* percebi que quando a funcao pagCreditoDebito e chamada dentro da funcao pagCartao ela nao vai adicionar essa classe na lista de Credito e debito, fazendo assim uma mudança ser necessária, a partir de agora

(datetime.date.now()) a lista de cada método de pagamento será atualizada dentro da própria função e não dentro do switch case. Ficando assim:

```
  public static void pagDinheiro(float valor, Scanner scanner, List<Especie>
  especieList){
    Especie pag = new Especie();
    pag.setValor(valor);
    System.out.print("Insira o valor pago pelo usuário-");
    float valorUsr = scanner.nextFloat();
    pag.setValorUsr(valorUsr);
    especieList.add(pag);
  }

  public static void pagCartao(float valor, Scanner scanner, List<Cartao>
  cartaoList, List<CreditoeDebito> cartaoDebitoList){
    Cartao pag = new Cartao();
    pag.setValor(valor);
    System.out.print("Insira o numero de prestacoes -");
    int numPrest = scanner.nextInt();

    if(numPrest==1){
      pagCreditoDebito(valor, cartaoDebitoList);
    }
    else{
      pag.setPrest(numPrest);
      cartaoList.add(pag);
    }
  }

  public static void pagCreditoDebito(float valor, List<CreditoeDebito>
  cartaoDebitoList){
    CreditoeDebito pag = new CreditoeDebito();
    pag.setValor(valor);
    cartaoDebitoList.add(pag);
  }
}
```

Agora falta apenas fazer o print final com todas as informações.

```java
public static void printFinal(List<CreditoeDebito> cartaoDebitoList,
List<Cartao> cartaoList, List<Especie> especieList){
  System.out.print("\n\n\n");

  System.out.println("Quantidade de operações: " + (cartaoDebitoList.size() +
  cartaoList.size() + especieList.size()));

  System.out.println("Operações por cada tipo");
  System.out.println("Dinheiro em Espécie: " + especieList.size());
  System.out.println("Cartão de Credito: " + cartaoList.size());
  System.out.println("Débito ou 1x no Crédito: " + cartaoDebitoList.size());

  System.out.print("\nO valor total gasto foi de: R$" + valorTotal);


}
```

A variável valorTotal é uma variável global que é incrementada à medida que cada nova compra é computada.

```java
ass Main {
  static float valorTotal;
  Run | Debug
  public static void main(String[] args){

    //Instanciando classe de pagamento
    Scanner scanner = new Scanner(System.in);
    //listas com cada tipo de objeto
    List<Especie> especieList = new ArrayList<Especie>();
    List<Cartao> cartaoList = new ArrayList<Cartao>();
    List<CreditoeDebito> cartaoDebitoList = new ArrayList<CreditoeDebito>();
    while(true){

      System.out.print("Insira o valor da compra: (Digite 'parar' para receber o
      relatorio)\n-");
      String resp = scanner.next();
      if(resp.equals("parar"))
        break;
      float valor = Float.parseFloat(resp);
      valorTotal += valor;
```

No fim, o programa funcionou extremamente bem :)

```
Insira o valor da compra: (Digite 'parar' para receber o relatorio)
-50
Escolha uma opcao:
1- Dinheiro em especie
2- Credito
3- Debito
1
Insira o valor pago pelo usuário-55
Insira o valor da compra: (Digite 'parar' para receber o relatorio)
-60
Escolha uma opcao:
1- Dinheiro em especie
2- Credito
3- Debito
2
Insira o numero de prestacoes -4
Insira o valor da compra: (Digite 'parar' para receber o relatorio)
-600
Escolha uma opcao:
1- Dinheiro em especie
2- Credito
3- Debito
2
Insira o numero de prestacoes -1
Insira o valor da compra: (Digite 'parar' para receber o relatorio)
-56
Escolha uma opcao:
1- Dinheiro em especie
2- Credito
3- Debito
3
Insira o valor da compra: (Digite 'parar' para receber o relatorio)
-parar



Quantidade de operações: 4
Operações por cada tipo
Dinheiro em Espécie: 1
Cartão de Credito: 1
Débito ou 1x no Crédito: 2

O valor total gasto foi de: R$766.0
```