

Formatando o código e renomeando variáveis

Agora que já praticamos algumas técnicas de navegação dentro do projeto, vá até a classe `RegistroDePagamento`. Note que temos o seguinte código:

```
public class RegistroDePagamento {  
  
    private final static List<Pagamento> psr = new ArrayList<>();  
  
    public void registra(List<Pagamento> ps) {  
        for (int i = 0; i < ps.size(); i++) {  
            Pagamento p = ps.get(i);  
            if (p.getTipo().equals(Tipo.CREDITO)) { p.setValor(p.getValor() * 1.02);  
                psr.add(p); System.out.println("Pagamento realizado " + p);  
            }if (p.getTipo().equals(Tipo.DEBITO)) { p.setValor(p.getValor() * 1.001);  
                psr.add(p); System.out.println("Pagamento realizado " + p);  
            } if (p.getTipo().equals(Tipo.DINHEIRO)) { psr.add(p); System.out.println("Pagamento realizado " + p);  
        } if (!psr.isEmpty()) { System.out.println("Todos os pagamentos");  
        for (int i = 0; i < psr.size(); i++) {  
            System.out.println(psr.get(i));  
        }  
    } else {System.out.println("Não temos pagamentos");}  
    }  
}
```

Observe que inicialmente é bem difícil de compreender o que essa classe faz.

Em outras palavras, precisamos realizar alguns processos de refatoração para que a leitura do código seja mais compreensível para qualquer desenvolvedor.

Copiando uma classe

Antes de começarmos com as técnicas de refatoração, faremos uma cópia da `RegistroDePagamento` para comparmos com o resultado final que faremos nos exercícios a seguir. Para isso, coloque o cursor no nome da classe ou selecione ela na view project utilize o atalho **F5**.

Em seguida, dê o nome de `RegistroDePagamentoAlpha` e deixe no mesmo pacote, por fim tecle **Enter** ou clique em **OK**. Observe que foi feita uma cópia da classe.

Agora podemos começar com o nosso processo de refatoração na classe `RegistroDePagamento`

Formatando o código automaticamente

A primeira das técnicas que podemos aplicar neste instante é a famosa formatação automática do código (no IntelliJ é conhecido como **Reformat Code**) que permite ajustar a indentação. Para formatar o código basta apenas usar o atalho **Ctrl + Alt + L / Alt + CMD + L**.

Observações: Existe um detalhe peculiar neste atalho, por exemplo, no Ubuntu, ele faz parte de um atalho do sistema operacional que bloqueia a tela. Ou seja, existem duas maneiras para usar esse recurso por atalho - Mudar o atalho padrão do sistema - Mudar o atalho padrão do IntelliJ

Para esse abordagem não existe o certo ou o errado, você pode modificar conforme achar mais coerente para você.

De modo geral, prefiro mudar o atalho do sistema, por já ter me acostumado com os do IntelliJ, no caso eu deixo atalho de bloquear tela como **Super/Windows + L** que é bem comum em sistemas Windows.

Renomeando variáveis

Com o código ajustado precisamos renomear as variáveis para que sejam comprehensíveis para qualquer programador. Em outras palavras, realize a seguinte mudança:

- `psr` para `PAGAMENTOS` (Por ser `final`)
- `ps` para `pagamentos`
- `p` para `pagamento`

Para realizar esse ajuste visando o processo de refatoração, podemos utilizar o **Rename** do IntelliJ com o atalho **Shift + F6**.

Após esse ajuste, execute novamente a classe `Principal` e verifica se tudo funciona como antes.