Modificación Patrón Adapter

Objetivo

Agregar un nuevo banco “zBank” con respuesta síncrona, en el cual, si no se autoriza el crédito se debe lanzar una excepción

Resolución

A continuación, se detallarán los pasos realizados para completar el objetivo de la tarea

i. Creación de los componentes para “zBank” (Api, Request, Response)

```
package oscarblancarte.ipd.adapter.creditapi.bankz;
public class ZBankCreditAPI {
      public ZBankCreditResponse sendCreditRequest(ZBankCreditRequest request) throws Exception {
         ZBankCreditResponse response =new ZBankCreditResponse();
         if(request.getRequestAmount() <= 50000) {
            response.setAproval(true);
         } else {
            response.setAproval(false);
            throw new Exception("Aprobacion rechazada zBank");
         }
      return response;
   }
}
```

```
public class ZBankCreditRequest {
   private String customerName;
   private double requestAmount;

   public String getCustomerName() { return customerName; }
   public void setCustomerName(String customerName) { this.customerName = customerName; }

   public double getRequestAmount() { return requestAmount; }
   public void setRequestAmount(double requestAmount) { this.requestAmount = requestAmount; }
}
```

```
public class ZBankCreditResponse {
   public boolean aproval;

   public boolean isAproval() { return aproval; }
   public void setAproval(boolean aproval) { .aproval = aproval; }
}
```

ii. Creación del Adapter para el nuevo banco

```
public class ZBankCreditAdapter implements IBankAdapter {
   @Override
   public BankCreditResponse sendCreditRequest(BankCreditRequest request) {
      ZBankCreditRequest zrequest = new ZBankCreditRequest();
      zrequest.setCustomerName(request.getCustomer());
      zrequest.setRequestAmount(request.getAmount());

      ZBankCreditAPI api = new ZBankCreditAPI();
      ZBankCreditResponse zresponse = new ZBankCreditResponse();
      try {
         zresponse = api.sendCreditRequest(zrequest);
      } catch(Exception ex) {
         ex.printStackTrace();
         zresponse.setAproval(false);
      }
```

```
    BankCreditResponse response = new BankCreditResponse();
    response.setApproved(zresponse.isAproval());
    return response;
  }

}
```

iii. Modificación de la clase “AdapterMain” para llamar al nuevo banco

```
public class AdapterMain {

  public static void main(String[] args) {
    //Generic request for the two API's
    BankCreditRequest request = new BankCreditRequest();
    request.setCustomer("Oscar Blancarte");
    request.setAmount(250000);

    System.out.println("Solicitante: " + request.getCustomer() +
                "\nMonto: " + request.getAmount() + "\n\n \n ");

    IBankAdapter xBank = new XBankCreditAdapter();
    BankCreditResponse xresponse = xBank.sendCreditRequest(request);
    System.out.println("xBank approved > " + xresponse.isApproved() + "\n");

    IBankAdapter yBank = new YBankCreditAdapter();
    BankCreditResponse yresponse = yBank.sendCreditRequest(request);
    System.out.println("yBank approved > " + yresponse.isApproved() + "\n");

    IBankAdapter zBank = new ZBankCreditAdapter();
    BankCreditResponse zresponse = zBank.sendCreditRequest(request);
    System.out.println("zBank approved > " + zresponse.isApproved() + "\n");

    if (xresponse.isApproved()) {
      System.out.println("xBank approved your credit, congratulations!!");
    } else if (yresponse.isApproved()) {
      System.out.println("yBank approved your credit, congratulations!!");
    } else if (zresponse.isApproved()) {
      System.out.println("zBank approved your credit, congratulations!!");
    } else {
      System.out.println("Sorry your credit has not been approved");
    }
  }
}
```

Patrón Adapter

Por medio de la clase ZBankCreditAPI vamos a ser capaces de realizar el procesamiento de la solicitud de crédito; es decir se encargará de simular la comunicación con el banco

Por medio de la clase ZBankCreditAdapter el cual se encargará de realizar la adaptación para poder consumir el nuevo api implementado; este adaptador heredado de nuestra clase genérica, con la finalidad de garantizar la estandarización dentro de los diferentes Apis que pertenecen a diferentes bancos que nuestra empresa ha implantado

Finalmente, en el método Main de la clase AdapterMain, vamos a realizar el llamado del api, por medio de nuestro Adapter, para validar si podemos o no acceder al crédito (funcionalidad del api)

Resultado

Resultado con crédito rechazado:

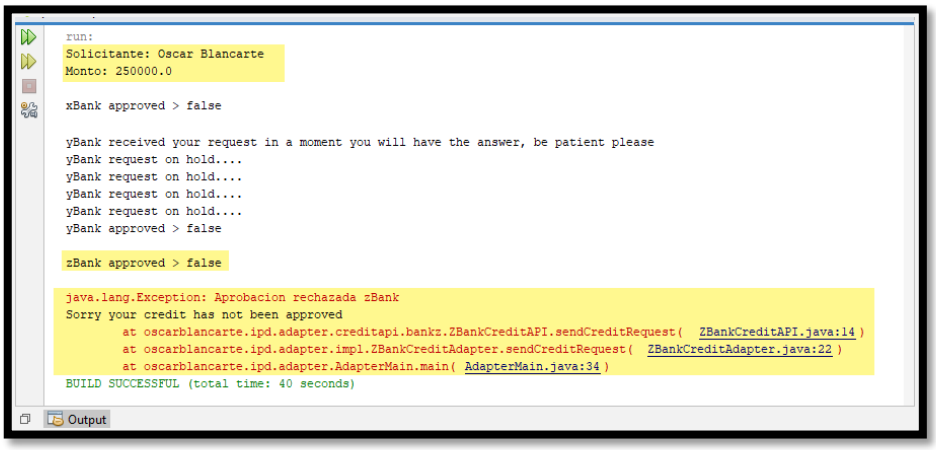
```
run:
Solicitante: Oscar Blancarte
Monto: 250000.0

xBank approved > false

yBank received your request in a moment you will have the answer, be patient please
yBank request on hold....
yBank request on hold....
yBank request on hold....
yBank request on hold....
yBank approved > false

zBank approved > false

java.lang.Exception: Aprobacion rechazada zBank
Sorry your credit has not been approved
   at oscarblancarte.ipd.adapter.creditapi.bankz.ZBankCreditAPI.sendCreditRequest(ZBankCreditAPI.java:14)
   at oscarblancarte.ipd.adapter.impl.ZBankCreditAdapter.sendCreditRequest(ZBankCreditAdapter.java:22)
   at oscarblancarte.ipd.adapter.AdapterMain.main(AdapterMain.java:34)
BUILD SUCCESSFUL (total time: 40 seconds)
```



Resultado con crédito aprobado:

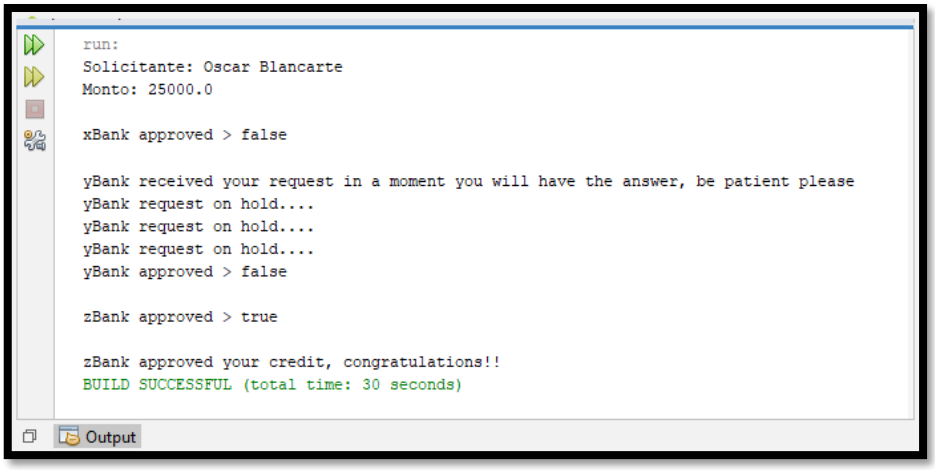
```
run:
Solicitante: Oscar Blancarte
Monto: 25000.0

xBank approved > false

yBank received your request in a moment you will have the answer, be patient please
yBank request on hold....
yBank request on hold....
yBank request on hold....
yBank approved > false

zBank approved > true

zBank approved your credit, congratulations!!
BUILD SUCCESSFUL (total time: 30 seconds)
```


run:
Solicitante: Oscar Blancarte
Monto: 25000.0
xBank approved > false
yBank received your request in a moment you will have the answer, be patient please
yBank request on hold....
yBank request on hold....
yBank request on hold....
yBank approved > false
zBank approved > true
zBank approved your credit, congratulations!!
BUILD SUCCESSFUL (total time: 30 seconds)
Output