O objetivo deste exercício é criar um conjunto de classes Java para simular a operação de um sistema de autorização de acesso baseado em leitores de código de barra. As autorizações de acesso são armazenadas em um banco de dados controlado pela classe **DBAuthorization**, que será descrita mais adiante. Todas as vezes que um usuário quiser ter acesso a uma determinada área da empresa será necessário passar o seu cartão de acesso na leitora de código de barras e fornecer uma **password**. Estas informações serão validadas contra as autorizações existentes na base de dados. Isto é, caso exista uma autorização com o mesmo código lido do cartão do usuário e cuja **password** seja a mesma que a digitada por ele, o acesso será autorizado e uma porta será automaticamente aberta.

As classes e interfaces envolvidas nesse sistema são descritas abaixo.

```
package validation;
public interface IAuthorization
{
      public int getCode();
      public String getPsw();
      public void incorrectAttempts();
      public int getNumberIncorrectAttempts();
}
```

A interface **IAuthorization** define um conjunto de operações que devem ser usadas para validar uma tentativa de acesso. Estas operações devem fornecer os seguintes serviços:

getCode() – retorna o código associado a uma autorização. Este código deve ser o mesmo que estará normalmente impresso, através de código de barra, em cartões de acesso às áreas controladas pelo sistema;

getPsw() – retorna a password associada a uma autorização;

incorrectAttempt() – deve ser chamada todas a vezes que for informada uma password incorreta em uma tentativa de acesso. Esta operação deve somar uma unidade à quantidade de tentativas de acesso rejeitadas associadas a uma autorização;

getNumberIncorrectAttempts() – retorna o número de tentativas de acessos rejeitadas associadas a uma determinada autorização.

Para implementar os serviços definidos pela interface **IAuthorization** você deverá criar uma classe chamada **Authorization**. Essa classe deverá estar localizada em um pacote chamado **security**.

O banco de dados de autorizações é gerenciado por uma classe chamada **DBAuthorization**, localizada no pacote **security**. Esta classe armazena uma lista de autorizações (objetos do tipo **IAuthorization**), que serão acessadas pela classe responsável por validar uma tentativa de acesso.

```
package security;
import validation.*;
public class DBAuthorization
{
      static IAuthorization []list;
```

```
      public static IAuthorization getAuthorization(int code)
      {
      }
}
```

O método **getAuthorization()**, que deverá ser implementado por você, irá percorrer sequencialmente a lista de autorizações e procurar por uma autorização cujo código seja igual ao valor do parâmetro **code**. Caso a encontre, a autorização deverá ser retornada, caso contrário o método deverá retornar **null**.

A classe **Checker**, localizada no pacote **validation**, é a principal responsável pela validação de uma tentativa de acesso. Todas as vezes que alguém passar o cartão de acesso no leitor e digitar uma password o método **check()** será chamado. Ele irá solicitar à classe **DBAuthorization** que lhe devolva a autorização associada ao código digitado, para, então, proceder com a validação da password.

```
public class Checker
{
      public boolean check(int cod,String psw)
      {
      }
}
```

O método **check()** deverá retornar **true** ou **false**. O valor **true** será retornado se todas as condições a seguir forem verdadeiras. Caso contrário, será retornado o valor **false**. As condições são as seguintes:

- Deve existir uma autorização cujo código seja igual ao valor do parâmetro **cod**;
- Caso exista tal autorização, a sua **password** deve ser igual ao valor do parâmetro **psw**;
- O número de tentativas rejeitadas **não** pode ser maior ou igual a **três**.

Além da validação de uma tentativa de acesso, este método deverá solicitar que seja somada uma unidade ao número de tentativas rejeitadas todas as vezes que for informada uma password diferente da **password** existente em uma autorização.

Implemente a função **check()** descrita acima.