Lab 3 - Structure-Based (White-box) Testing Techniques

1. Implement the functionality of the booking process of train tickets in the Railway ticketing portal which includes the following modules:
   1. to search train using the following capabilities: destination and time;
   2. to calculate the ticket price according to one-way or round trip;
   3. to calculate the ticket price according to dynamic pricing rules;
   4. to book tickets.
      The user could book the train tickets using some GUI. => POSTMAN

2. Write input test data for a set of test cases to achieve:
   1. **Statement coverage (SC) && Decision coverage (DC) && Condition coverage (CC) for functions:**
      1. <u>search train using the following capabilities: destination and time;</u>
            If startDestination is Vienna
                  return tickets filtered only by 'Vienna'
            Else If endDestination is Munich
                  return tickets filtered only by 'Munich'
            Else If departureTime is '08:20:00'
                  return tickets filtered only by '08:20:00'
            Else If startDestination is 'Vienna' and departureTime is '08:20:00'
                  return tickets filtered by 'Vienna' and '08:20:00'
            Else If startDestination is 'Vienna' and endDestination is 'Munich'
                   return tickets filtered by 'Vienna' and 'Munich'
            Else If departureTime is '08:20:00' and endDestination is 'Munich'
                  return tickets filtered by '08:20:00' and 'Munich'
            Else If startDestination is 'Vienna', endDestination is 'Munich' and departureTime is '08:20:00'
                   return tickets filtered by 'Vienna', 'Munich' and '08:20:00'
            EndIf
      2. <u>calculate the ticket price according to one-way or round trip && calculate the ticket price according to dynamic pricing rules;</u>

      - **Testdata: startDestination is 'Vienna'; endDestination is 'Munich'; rushHour is '09:20:00'; notRushHour is '10:10:00'**

            If startDestination and endDestination and rushHour
                  return fullPrice
            Else if (startDestination and endDestination) and (startDestination and endDestination) and rushHour
                  return fullPrice
            Else if (startDestination and endDestination) and notRushHour
                  return fullPrice with 5% discount
            Else if (startDestination and endDestination) and (startDestination and endDestination) and notRushHour
                  return price with 5% discount
            Else if (startDestination and endDestination) and (startDestination and endDestination) and over60s
                  return price with 34% discount
            Else if (startDestination and endDestination) and over60s
                  return price with 34% discount

Else if (startDestination and endDestination) and (startDestination and endDestination) and withChildUnder16
  return price with 10% discount
Else if (startDestination and endDestination) and withChildUnder16
  return price with 10% discount
Else if (startDestination and endDestination) and withChildUnder16 and withFamilyCard
  return price with 50% discount
Else if (startDestination and endDestination) and (startDestination and endDestination) and withChildUnder16 and withFamilyCard
  return price with 50% discount
EndIf

3. <u>to book tickets</u>
If userExists is 'true' and ticketExists is 'true'
  return list of bookedTickets of the user
Else if userExists is 'false'
  throw exception
Else if ticketExists is 'false'
  throw exception
EndIf

Draw the Control flow graphs for the functions - in another pdf;