**Practice “Control Flow”**

1. Add new enum DiscountType with following values **None, Low, Medium, High**
   * Add new property DiscountType to Customer class.
   * Add new class DiscountCalculator with method Calculate which receive input parameter price and input parameter discount type:
     1. if discount type is None, discount is 0%;
     2. if discount type is **Low**, discount is 3%;
   * Try to use DiscountCalculator, create user with different discount type and calculate discount for different sum;
2. Add logic to class DiscountCalculator method Calculate
   * if discount type is **Medium**, discount is 5%;
   * if discount type is **High**, discount is 7%;
   * Try to use DiscountCalculator, create user with different discount type and calculate discount for different sum;
3. Rewrite method calculate to switch statement
   * Try to use DiscountCalculator, create user with different discount type and calculate discount for different sum;
4. Create order with 5 purchase and calculate discount for each product and calculate sum of discount with using statement **for**;
5. Create order with 5 purchase and calculate discount for each product and calculate sum of discount with using statement **foreach**;
6. Create order with 5 purchase and calculate discount for each product and calculate sum of discount with using statement **while**;
7. Create order with 5 purchase and calculate discount for each product and calculate sum of discount with using statement **do while**;
8. Rewrite method CalculateDiscount with following logic:   
   if discountType is High – throw Exception with message “High discount type is unavaialable”;
9. Create order with 5 purchase and calculate discount for each product and calculate sum of discount with using statement **foreach**;
10. Add handling of throwing exceptions – if exception was throw then skip current product
11. Add logging of product name, price and discount in finally statement;