**Explanation of the workflow of the application for discount cards**

The application entry point is the Startup.cs class. It runs after a command on the console, which has to contain the discount card type, the turnover for the previous month and the value of the purchase. The application proceeds only if the input data is valid.

Once the card type is verified, the app instantiates a relevant card object using the Factory pattern with base class instead of an interface. The three supported card types – Bronze, Silver and Gold derive from the abstract class Card, which has an abstract property DiscountRate. Thus, every child class could override it and get the value depending on its specific requirements for the previously generated turnover. By invoking the public property of the current card, we get the appropriate discount rate. Together with the purchase value we pass it to the PayDesk.GetOutput() method to retrieve the expected result. Since the method is static, we don’t need to have an instance of the class, but simply to call it.

PayDesk.cs has another two methods – GetDiscount() and GetTotalValue(). They are private, because they are needed only within the class, in order to pass the outcome to the GetOutput() method. The latter returns a string, which in the end is being written on the console.