

@startuml

```
class Buyer {
    -buyerId: int
    -имя: string
    -email: string
    -телефон: string
    -адрес: string
    +регистрация(): void
    +авторизация(): void
    +поискТоваров(): void
    +добавитьВКорзину(): void
    +оформитьЗаказ(): void
    +оставитьОтзыв(): void
    +добавитьВИзбранное(): void
}
class Seller {
    -sellerId: int
    -имя: string
    -email: string
    -телефон: string
    -категорииТоваров: string
    -рейтинг: double
    +регистрация(): void
    +авторизация(): void
    +добавитьТовар(): void
    +обновитьТовар(): void
    +управлятьАкциями(): void
}

class Product {
    -productId: int
    -название: string
    -описание: string
    -цена: double
    -наличие: bool
    -рейтинг: double
    -категория: Category
    -скидка: Discount
    +обновитьРейтинг(): void
    +получитьОписание(): string
}
class Category {
    -categoryId: int
    -название: string
    -описание: string
    +поискТоваровПоКатегории(): List<Product>
}

class Cart {
    -cartId: int
    -buyerId: int
    -списокТоваров: List<Product>
    -общаяСтоимость: double
}
```

```

        +добавитьТовар(): void
        +удалитьТовар(): void
        +рассчитатьСтоимость(): double
    }

class Order {
    -orderId: int
    -buyerId: int
    -списокТоваров: List<Product>
    -датаЗаказа: Date
    -статус: string
    -методОплаты: PaymentMethod
    -адресДоставки: string
    +подтвердитьЗаказ(): void
    +отменитьЗаказ(): void
    +отслеживатьСтатус(): string
}

class PaymentMethod {
    -paymentMethodId: int
    -типМетода: string
    -статусМетода: string
    +выбратьМетодОплаты(): void
    +оплатить(): void
}

class Delivery {
    -deliveryId: int
    -orderId: int
    -адресДоставки: string
    -статусДоставки: string
    -стоимостьДоставки: double
    +отследитьДоставку(): void
    +отменитьДоставку(): void
}

class Review {
    -reviewId: int
    -buyerId: int
    -productId: int
    -рейтинг: int
    -комментарий: string
    -датаОтзыва: Date
    +оставитьОтзыв(): void
    +удалитьОтзыв(): void
}

class Wishlist {
    -wishlistId: int
    -buyerId: int
    -списокТоваров: List<Product>
    +добавитьВИзбранное(): void
    +удалитьИзИзбранного(): void
    +просмотретьИзбранное(): List<Product>
}

```

```

}

class Discount {
    -discountId: int
    -productId: int
    -размерСкидки: double
    -срокДействия: Date
    +применитьСкидку(): void
    +отменитьСкидку(): void
}

class Buyer << (C,#ADD8E6) >>
class Seller << (C,#ADD8E6) >>

class Order << (C,#FFA500) >>
class Cart << (C,#FFA500) >>
class PaymentMethod << (C,#FFA500) >>
class Delivery << (C,#FFA500) >>

class Product << (C,#90EE90) >>
class Category << (C,#90EE90) >>
class Discount << (C,#90EE90) >>

class Wishlist << (C,#9370DB) >>
class Review << (C,#9370DB) >>

Buyer --> Cart
Buyer --> Order
Buyer --> Review
Buyer --> Wishlist

Order ..> PaymentMethod
Order ..> Delivery
Order o-- Product
Cart o-- Product
Product ..> Category
Product --> Review
Product *-- Discount
Product --> Wishlist

Seller --|> Product

@enduml

```

