Model::Main

UserAuthentication
(from shoppingapp)

-connection: Connection
-currentUserId: int
-currentUserRole: String

«constructor»+UserAuthentication()
+login(username: String, password: String): boolean
+logout(): void
+register(username: String, password: String, role: String): boolean
+getCurrentUserId(): int
+getCurrentUserRole(): String

ProductCatalog
(from shoppingapp)

-connection: Connection
-logger: Logger {readOnly}

«constructor»+ProductCatalog()
+addProduct(product: Product): void
+removeProduct(productId: int): void
+getProductById(productId: int): Product
+listProducts(): Product[*]
+addProductUsingFactory(type: String, id: int, name: String, price: double, description: String, additionalAttributes: String): void

CustomLogger
(from shoppingapp)

-logger: Logger {readOnly}

+log(level: Level, message: String): void
+log(level: Level, message: String, thrown: Throwable): void

-productCatalog

ProductFactory
(from shoppingapp)

-logger: Logger {readOnly}

+createProduct(type: String, id: int, name: String, price: double, description: String, additionalAttributes: String): Product

Electronics
(from shoppingapp)

-brand: String
-model: String
-warrantyPeriod: int

+getBrand(): String
+setBrand(brand: String): void
+getModel(): String
+setModel(model: String): void
+getWarrantyPeriod(): int
+setWarrantyPeriod(warrantyPeriod: int): void
+toString(): String

OrderProcessing
(from shoppingapp)

-logger: Logger {readOnly}

«constructor»+OrderProcessing()
+getInstance(): OrderProcessing
+placeOrder(order: Order): void
+cancelOrder(orderId: int): void

Product
(from shoppingapp)

#id: int
#name: String
#price: double
#description: String

+getId(): int
+setId(id: int): void
+getName(): String
+setName(name: String): void
+getPrice(): double
+setPrice(price: double): void
+getDescription(): String
+setDescription(description: String): void
+toString(): String

-paymentProcessor

PaymentProcessor
(from shoppingapp)

-logger: Logger {readOnly}

+processPayment(amount: double): boolean

*
productList
{collection="List"}

Clothing
(from shoppingapp)

-size: String
-color: String
-material: String

+getSize(): String
+setSize(size: String): void
+getColor(): String
+setColor(color: String): void
+getMaterial(): String
+setMaterial(material: String): void
+toString(): String

Order
(from shoppingapp)

-orderId: int
-userId: int
-totalPrice: double
-connection: Connection
-logger: Logger {readOnly}

«constructor»+Order(userId: int, productCatalog: ProductCatalog)
+addProduct(product: Product): void
+removeProduct(productId: int): void
-calculateTotal(): void
+getOrderId(): int
+setOrderId(orderId: int): void
+getUserId(): int
+setUserId(userId: int): void
+getProductList(): Product[*]
+setProductList(productList: Product[*]): void
+getTotalPrice(): double
+setTotalPrice(totalPrice: double): void
+saveOrder(): void
-saveOrderProducts(): void
+getOrderById(orderId: int, productCatalog: ProductCatalog): Order
-getProductsByOrderId(orderId: int, connection: Connection, productCatalog: ProductCatalog): Product[*]
+clearCart(): void
+deleteOrder(): void