Entity-классы (например, BankService, Client, Account, Product, Agreement, Transaction, Manager):

@Entity: Обозначает класс как сущность, которая будет сохранена в базе данных.

@Table: Позволяет настраивать параметры таблицы базы данных.

Repository-интерфейсы (например, BankServiceRepository, ClientRepository, AccountRepository, ProductRepository, AgreementRepository, TransactionRepository, ManagerRepository):

@Repository: Обозначает класс как Spring компонент, управляемый контейнером, отвечающий за доступ к данным.

Service-классы (например, BankServiceService, ClientService, ...):

@Service: Помечает класс как сервисный компонент Spring.

Controller-классы (например, BankServiceController, ClientController, ...):

@RestController: Обозначает класс как контроллер, который обрабатывает HTTP-запросы и возвращает данные в формате JSON.

@RequestMapping: Указывает базовый путь для всех методов контроллера.

@GetMapping, @PostMapping, @PutMapping, @DeleteMapping: Определяют HTTP-методы, обрабатываемые методами контроллера.

JPA-классы (например, Account, Agreement, ...):

@Id: Определяет поле как идентификатор сущности.

@GeneratedValue: Указывает стратегию генерации значения для идентификатора.

Lombok-аннотации (например, @Data, @NoArgsConstructor, ...):

@Data: Генерирует геттеры, сеттеры, методы toString, equals и hashCode.

@NoArgsConstructor: Генерирует конструктор без аргументов.