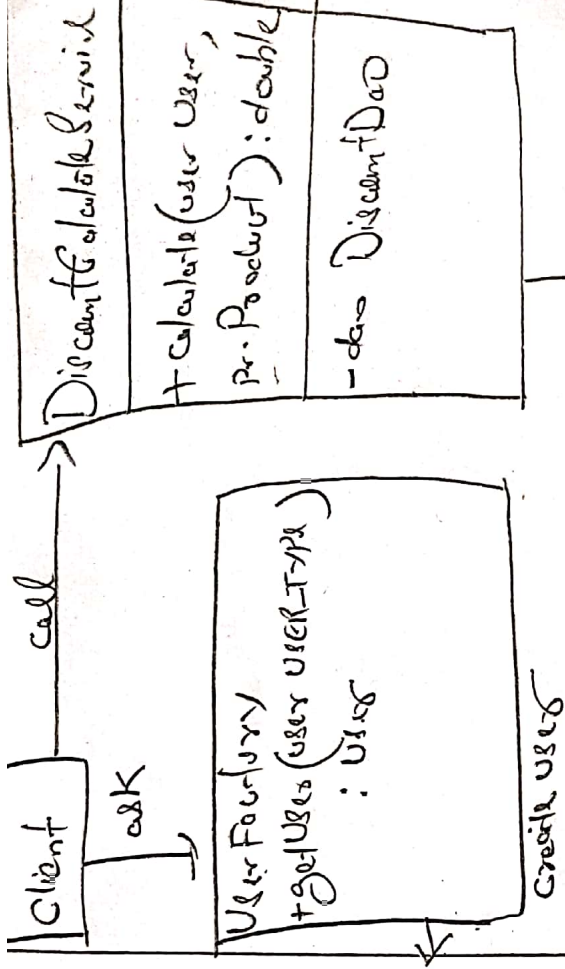


User << abstract >>
registerData: LocalData
discountRate: int
+ calculateDiscountPrice(Product p): double
+ User (discountRate: int, product: Product)

extends	extends	extends															
<table> <tr> <td>Affiliate</td><td>Customer</td><td>Employee</td></tr> <tr> <td>+ Affiliate (discountRate: int)</td><td>+ Customer (discountRate: int)</td><td>+ Employee (discountRate: int)</td></tr> <tr> <td>registerData: LocalData</td><td>registerData: LocalData</td><td>registerData: LocalData</td></tr> <tr> <td>+ calculateDiscountPrice(Product p): double</td><td>+ calculateDiscountPrice(Product p): double</td><td>+ calculateDiscountPrice(Product p): double</td></tr> <tr> <td>- discountRate: int</td><td>- discountRate: int</td><td>- discountRate: int</td></tr> </table>	Affiliate	Customer	Employee	+ Affiliate (discountRate: int)	+ Customer (discountRate: int)	+ Employee (discountRate: int)	registerData: LocalData	registerData: LocalData	registerData: LocalData	+ calculateDiscountPrice(Product p): double	+ calculateDiscountPrice(Product p): double	+ calculateDiscountPrice(Product p): double	- discountRate: int	- discountRate: int	- discountRate: int		
Affiliate	Customer	Employee															
+ Affiliate (discountRate: int)	+ Customer (discountRate: int)	+ Employee (discountRate: int)															
registerData: LocalData	registerData: LocalData	registerData: LocalData															
+ calculateDiscountPrice(Product p): double	+ calculateDiscountPrice(Product p): double	+ calculateDiscountPrice(Product p): double															
- discountRate: int	- discountRate: int	- discountRate: int															



DiscountDaoImp
+ calculateDiscountForGroup(u User, p Product): double