+ creationCompteClient(user: string,password: string,email:string) void creationCompteBancaire(user: string, type: string) void listerCompte(user) void rechercherCompte(user:string,serach:string) void Simulation(numCompte: int,Date:LocalDate) void +Depot(numCompte: int, montant: float) void +Retrait(numCompte: int. montant: float) void Log +Virement(source: int. destination:int) void Attribut + HistoriqueTranstsaction(numCompte: int) void + \$ systemeLog(message: string) void + ExportJson(numCompte: int) void + Help() void CompteUser -comptes:ArayListe<Compte> -compteBancaire:ArrayListe<CompteBancaire> mdp:String email:String +CompteUser (comptes:ArrayListe<Compte> , mdp:String , email:String) +getComptes() ArrayListe<Compte> +getMdp() String +getEmail()String +verfEmail()String +verfMdp() String + trier () void + rechercher() void ajouer() void exporterTous() void +Afficher() void **ItfCompteBancaire** + getTypeCompte() string + setTypeCompte(type: string) void CompteLdd plafond : float \$ Taux: float= 0.03 - \$ soldeMinimun : float= 1,5 +CompteLdd(plafond:float, Taux: float,soldeMin : float= 1,5) CompteBancaire deposer(som: float, soldeMin : float) : void calculerRendement(TauxLdd: floa): void -type:String -numeroCompte:int -solde :float Transaction -nom:String -date:LocalDate -historiques:ArrayListe<Transactions> -type:String plafond:float -amount:float ComptePel -date:LocalDate - compte:Compte plafond : float rib: string +Transaction(date:logodate ,type:String ,amount:float) \$ Taux: float= 0.0225 +Compte(type:String,numeroCompte:int,slode:float,nom:String,historiques:Historiques,date:LocalDate, rib: \$ soldeMinimun : int =50 string) +CompteLdd(plafond:float, Taux: float,soldeMin : float) + deposer(som: float, soldeMin : int) : void +exporterJson():string +exporterRib():string calculerRendement(Taux: floa): void +retrait(amount1:float)void {abstract} +depot(amount2:float)void {abstract} +solde(amount3:float)void +simuler(date:LocalDate):float LivretA plafond : float \$ Taux: float= 0.035 \$soldeMin : int=15 +CompteLdd(plafond:float, Taux: float,soldeMin : int) + deposer(som: float, soldeMin : int) : void + calculerRendement(Taux: float): void CompteCourant plafond : float - \$ soldeMin : int=10 +CompteLdd(plafond:float,soldeMin: int)

+ deposer(som: float, soldeMin: int): void

Main