

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto

Tiago

Declaro por minha honra que este diagrama foi realizado apenas pelos elementos que constituem o grupo do projeto.

Rodrigo Baptista

Library

```
- Creators: Collection< Creator >
- Works: Collection< Work >
- Users: Collection< User >
- Requests: Collection< Request >
- CurrentRate: Rate

+ registerCreator(name: string): Creator
+ getAllCreators(): Collection< Creator >
+ getCreator(name: string): Creator
+ registerBook(IDwork: int, available: int, stock: int, title: string, price: int, type: Type,
  Creators: Collection< string >, ISBN: string): Book
+ registerDVD(IDwork: int, available: int, stock: int, title: string, price: int, type: Type,
  Creator: string, I.G.A.C: string): DVD
+ getAllWorks(): Collection< works >
+ getWork(IDwork: int): work
+ registerUser(IDUser: int, name: string, email: string, history: Collection< Books >): User
+ getAllUsers(): Collection< User >
+ getUser(IDUser: int): User
+ requestBook(Douser: int, IDwork: int): Request
+ getRequests(IDUser: int): Collection< Request >
+ getCurrentRate(): Rate
+ advanceRate(days: int): void
```

bci



Digitalizada com CamScanner

Digitalizada com CamScanner

LibraryManager

```
- - Library : Library  
-- filename : String  
  
+ getLibrary() : Library  
+ save() : void  
+ search(filename: string) : void  
+ load(filename: String) : void  
+ importFile(filename: String) : void  
+ advanceDate(days: int) : void  
+ displayDate() : int  
+ registerCreator(name: string) : Creator  
+ getAllCreators() : Collection<Creator>  
+ getCreator(name: string) : Creator  
+ registerBook(IDWork: int, available: int, stock: int, title: String,  
    price: int, type: Type, Creators: Collection<String>, ISBN: String) : Book  
+ registerDVD(IDWork: int, available: int, stock: int, title: String,  
    price: int, type: Type, Creator: String, I_FAC: String) : DVD  
+ getAllWorks() : Collection<Work>  
+ getWork(IDwork: int) : Work  
+ searchWorks(criteria: SearchCriteria) : Collection<Work>  
+ ChangeInventory(IDWork: int, change: int) : void  
+ registerUser(IDUser: int, name: String, email: String,  
    history: Collection<Boolean>) : User  
+ getAllUsers() : Collection<User>  
+ getUser(IDUser: int) : User  
+ requestWork(IDUser: int, IDwork: int) : Request  
+ returnWork(IDwork: int) : void  
+ getAllRequest(IDUser: int) : Collection<Request>  
+ getAllRequests() : Collection<Request>  
+ getErrorCount(IDUser: int) : int  
+ payFine(IDUser: int) : void  
+ getAllNotifications(IDUser: int) : Collection<Notification>  
+ clearNotifications(IDUser: int) : void
```

loci

Creator

- name: String

+ Creator (name: String)

+ removeCreator (): void

<abstraction>
Work

- IDWORK: int

- stock: int

- title: String

- price: int

- type: Type

Work (input: String)

<enumeration>

Type

+ FICTION

+ SCITECH

+ REFERENCE

Book

- creators: Collection<String>

- ISBN: String

+ Book (input: String)

DVD

- creader: String

- IGAC: String

+ DVD (input: String)

bci.work

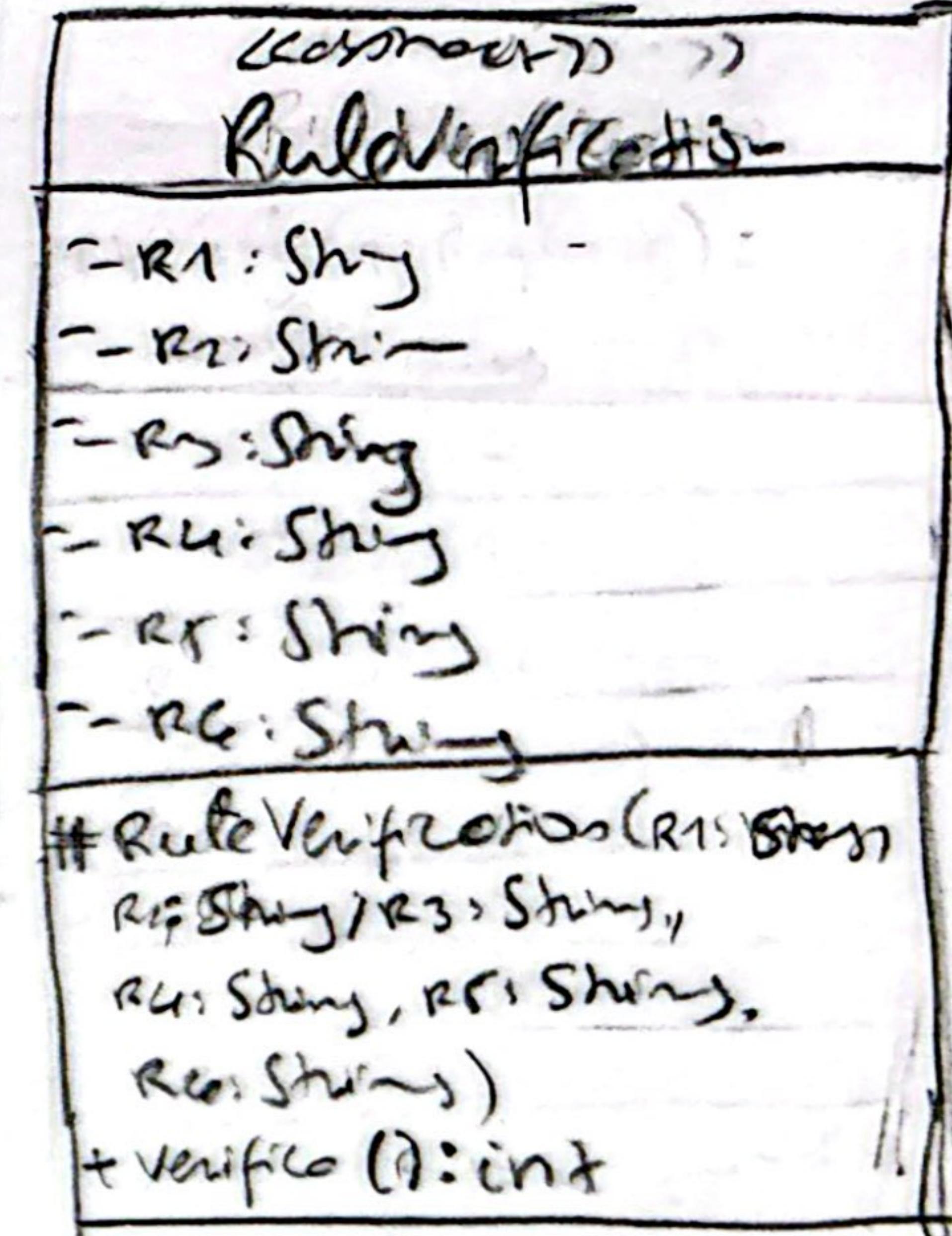
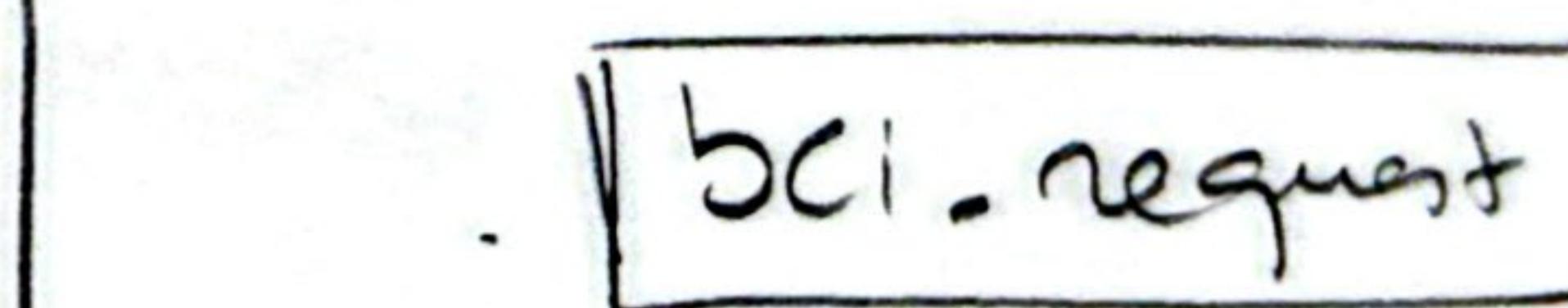
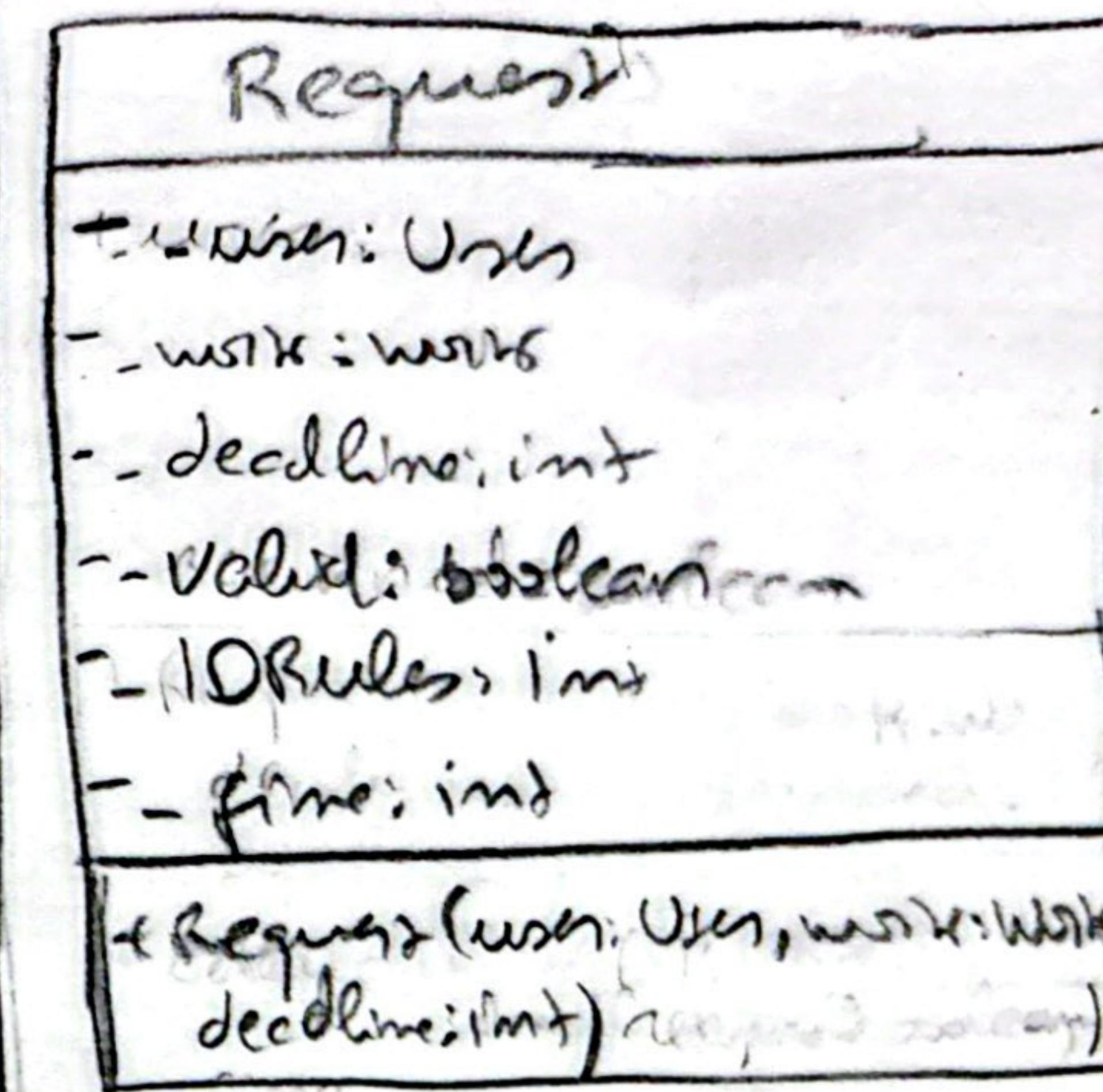
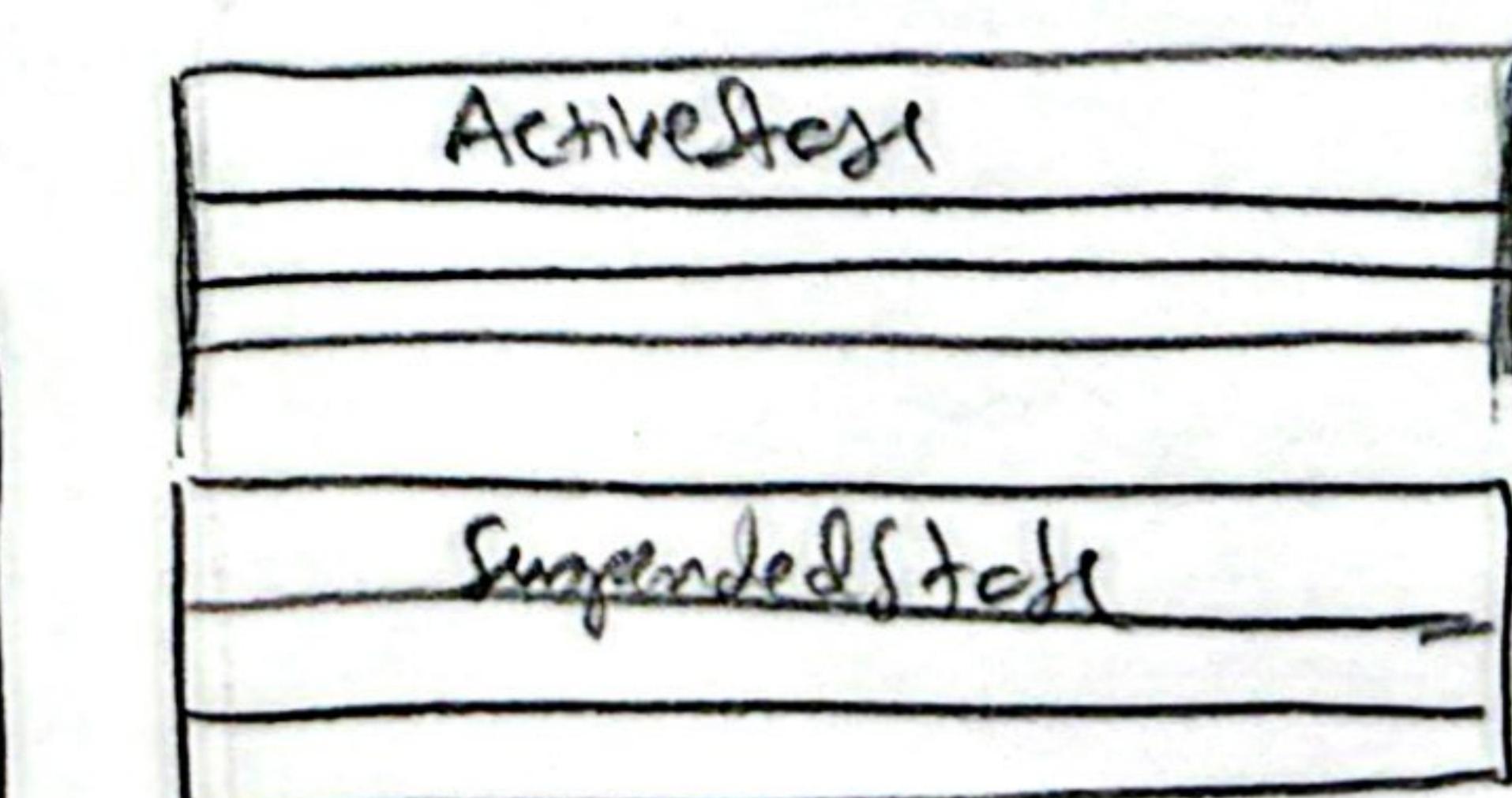
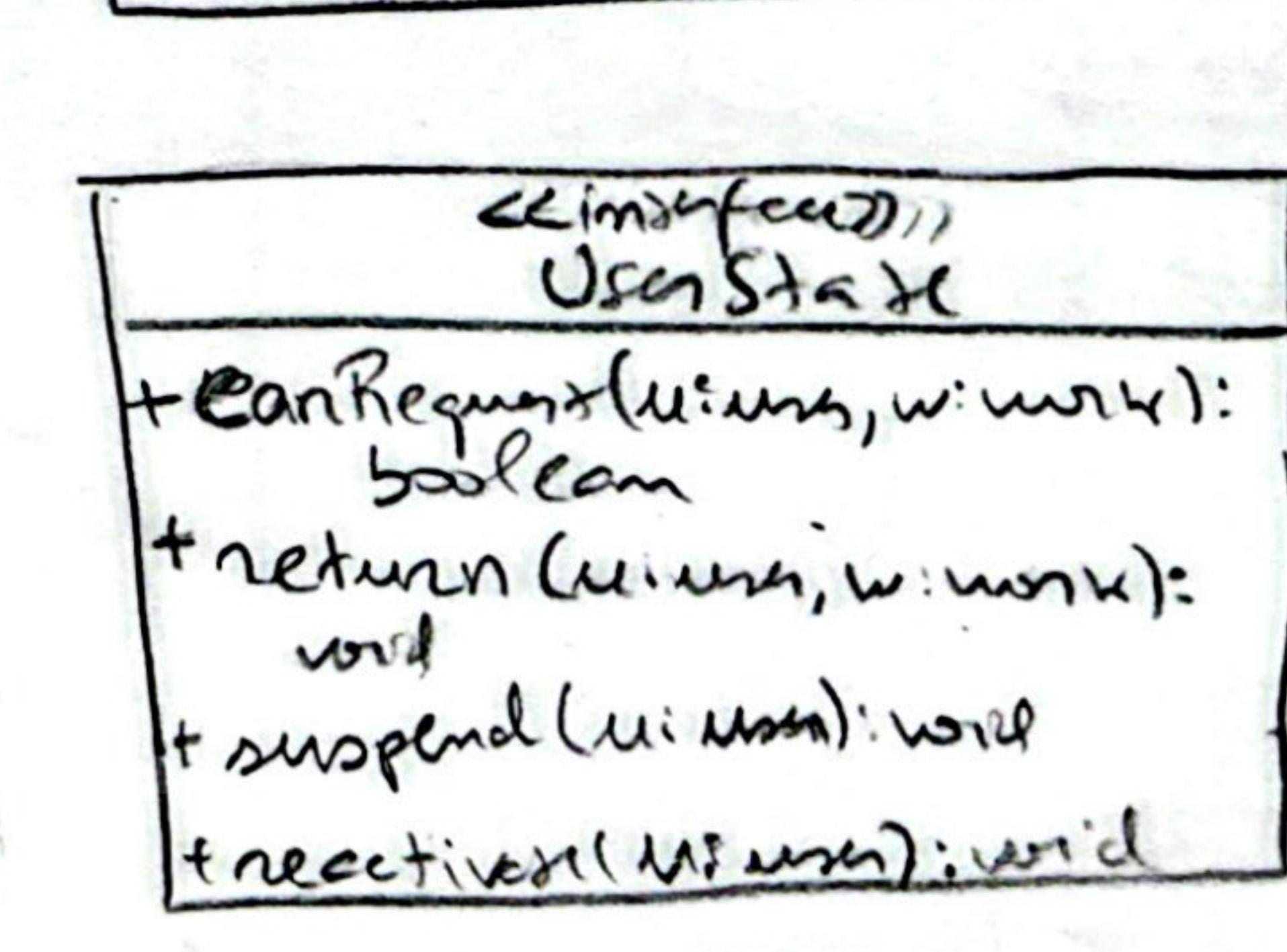
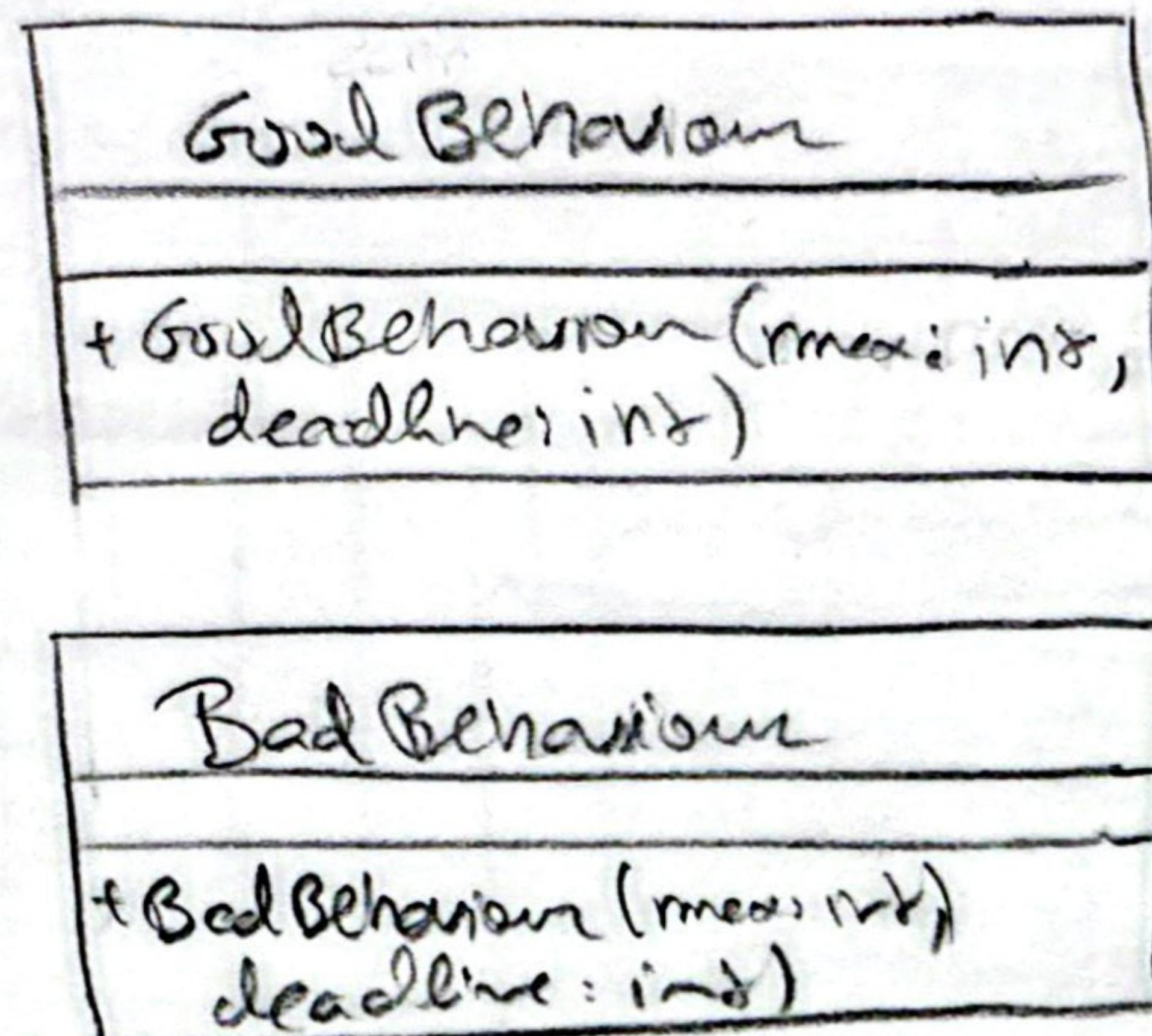
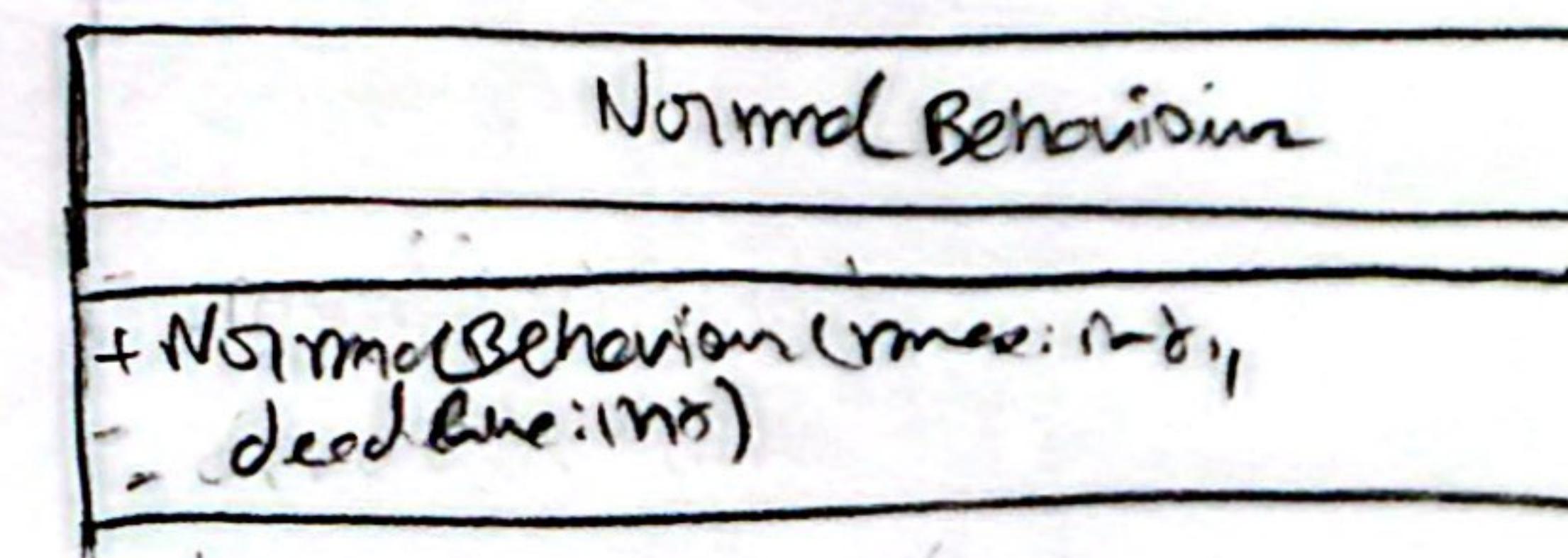
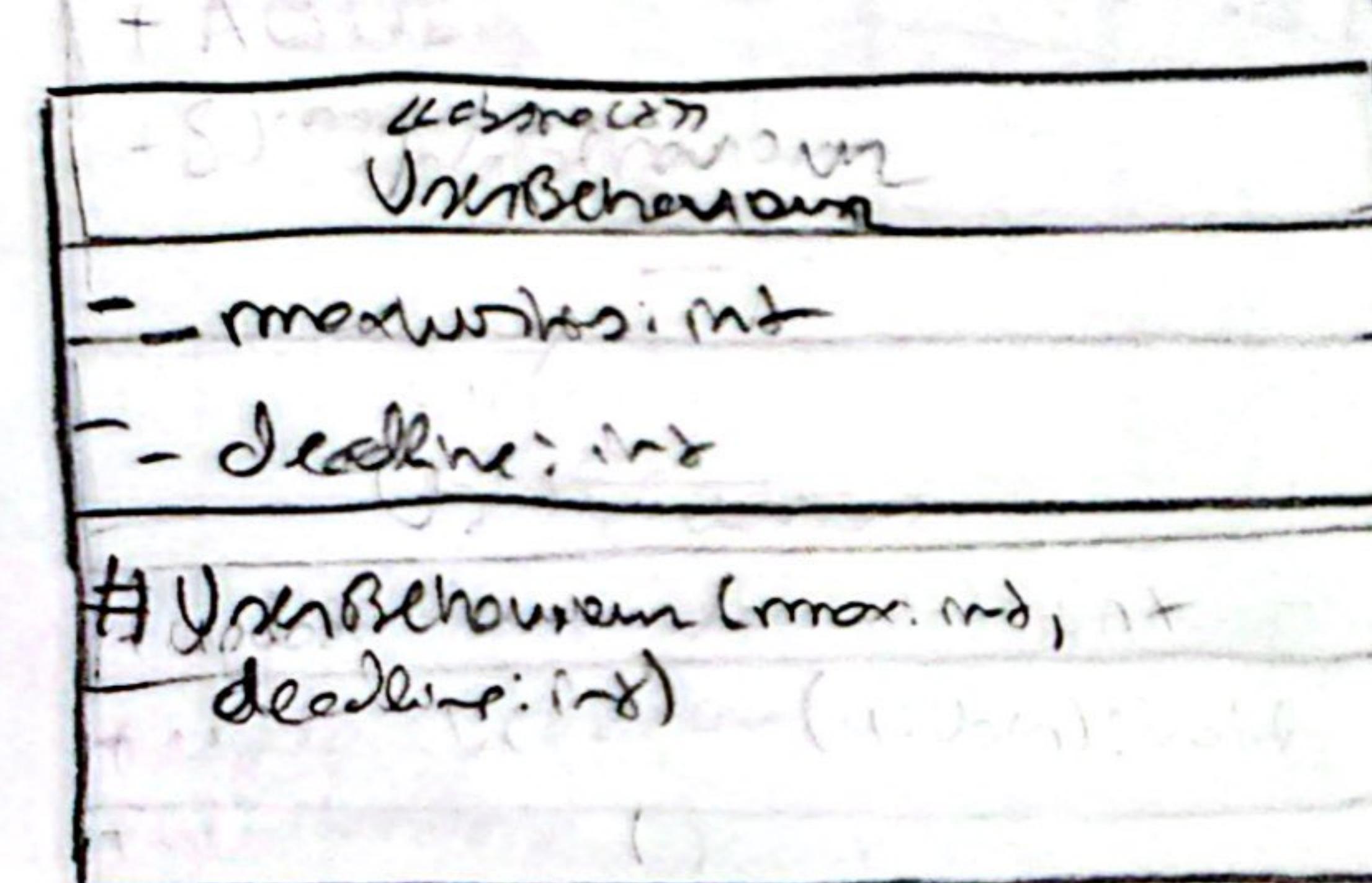
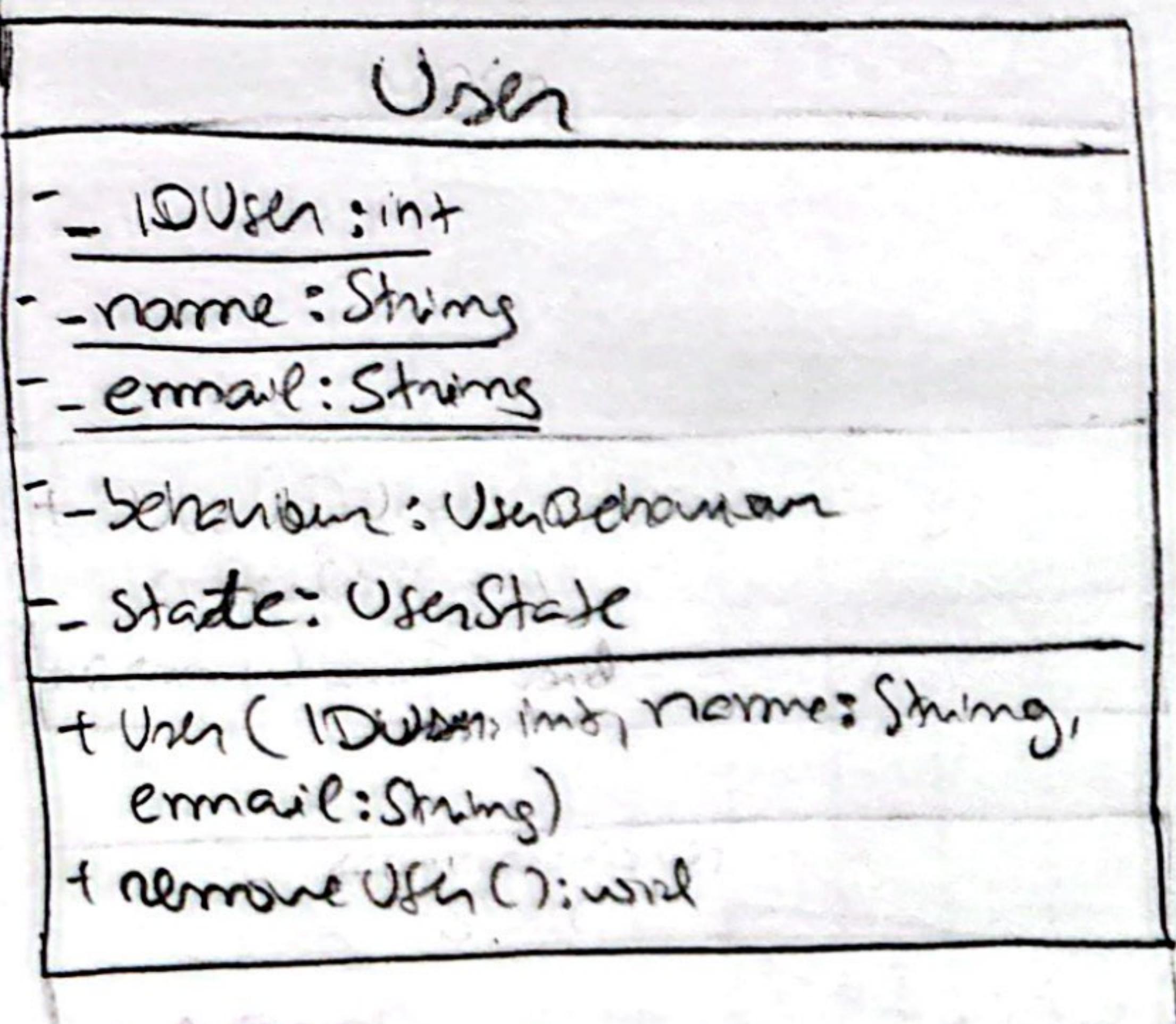
Date

- current-day: int

+ Date (atual: int)

+ advanceDate (days: int): void

bci.date



bci-exception

ImportException

MissingAssociationException

UnavailableException

UnrecognizedFormatException

Notification

-- type: NotificationType
-- msg: String

+ Notification (type: NotificationType,
msg: string, ID user: int)
+ deliverNotification (ID user: int)

<< enumeration >>
NotificationTypes

+ AVAILABLE
+ REQUEST

Search & Receiver

+ search(): Collection<agents>

SearchIndex

+ search(): User

SearchWork

+ search(): Work

bci-search

