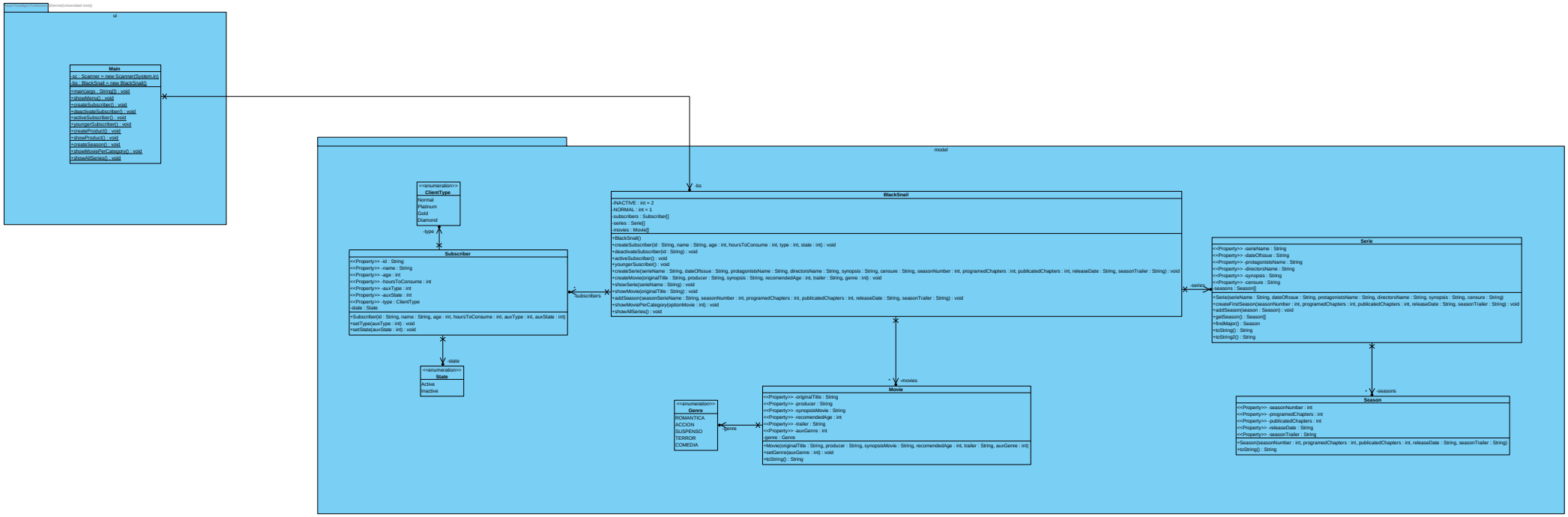
Visual Paradigm Professional(Dennis(Universidad Icesi))
ui
Main
-sc : Scanner = new Scanner(System.in)
-bs : BlackSnail = new BlackSnail()
+main(args : String[]) : void
+showMenu() : void
+createSubscriber() : void
+deactivateSubscriber() : void
+activeSubscriber() : void
+youngerSubscriber() : void
+createProduct() : void
+showProduct() : void
+createSeason() : void
+showMoviePerCategory() : void
+showAllSeries() : void
model
-bs
<<enumeration>>
ClientType
Normal
Platinum
Gold
Diamond
-type
Subscriber
<<Property>> -id : String
<<Property>> -name : String
<<Property>> -age : int
<<Property>> -hoursToConsume : int
<<Property>> -auxType : int
<<Property>> -auxState : int
<<Property>> -type : ClientType
-state : State
+Subscriber(id : String, name : String, age : int, hoursToConsume : int, auxType : int, auxState : int)
+setType(auxType : int) : void
+setState(auxState : int) : void
-state
<<enumeration>>
State
Active
Inactive
BlackSnail
-INACTIVE : int = 2
-NORMAL : int = 1
-subscribers : Subscriber[]
-series : Serie[]
-movies : Movie[]
+BlackSnail()
+createSubscriber(id : String, name : String, age : int, hoursToConsume : int, type : int, state : int) : void
+deactivateSubscriber(id : String) : void
+activeSubscriber() : void
+youngerSuscribe() : void
+createSerie(serieName : String, dateOfIssue : String, protagonistsName : String, directorsName : String, synopsis : String, censure : String, seasonNumber : int, programedChapters : int, publicatedChapters : int, releaseDate : String, seasonTrailer : String) : void
+createMovie(originalTitle : String, producer : String, synopsis : String, recomendedAge : int, trailer : String, genre : int) : void
+showSerie(serieName : String) : void
+showMovie(originalTitle : String) : void
+addSeason(seasonSerieName : String, seasonNumber : int, programedChapters : int, publicatedChapters : int, releaseDate : String, seasonTrailer : String) : void
+showMoviePerCategory(optionMovie : int) : void
+showAllSeries() : void
*
-subscribers
-series
*
-movies
Serie
<<Property>> -serieName : String
<<Property>> -dateOfIssue : String
<<Property>> -protagonistsName : String
<<Property>> -directorsName : String
<<Property>> -synopsis : String
<<Property>> -censure : String
-seasons : Season[]
+Serie(serieName : String, dateOfIssue : String, protagonistsName : String, directorsName : String, synopsis : String, censure : String)
+createFirstSeason(seasonNumber : int, programedChapters : int, publicatedChapters : int, releaseDate : String, seasonTrailer : String) : void
+addSeason(season : Season) : void
+getSeason() : Season[]
+findMajor() : Season
+toString() : String
+toString() : String
*
-seasons
Season
<<Property>> -seasonNumber : int
<<Property>> -programedChapters : int
<<Property>> -publicatedChapters : int
<<Property>> -releaseDate : String
<<Property>> -seasonTrailer : String
+Season(seasonNumber : int, programedChapters : int, publicatedChapters : int, releaseDate : String, seasonTrailer : String)
+toString() : String
<<enumeration>>
Genre
ROMANTICA
ACCION
SUSPENSO
TERROR
COMEDIA
-genre
Movie
<<Property>> -originalTitle : String
<<Property>> -producer : String
<<Property>> -synopsisMovie : String
<<Property>> -recomendedAge : int
<<Property>> -trailer : String
<<Property>> -auxGenre : int
-genre : Genre
+Movie(originalTitle : String, producer : String, synopsisMovie : String, recomendedAge : int, trailer : String, auxGenre : int)
+setGenre(auxGenre : int) : void
+toString() : String