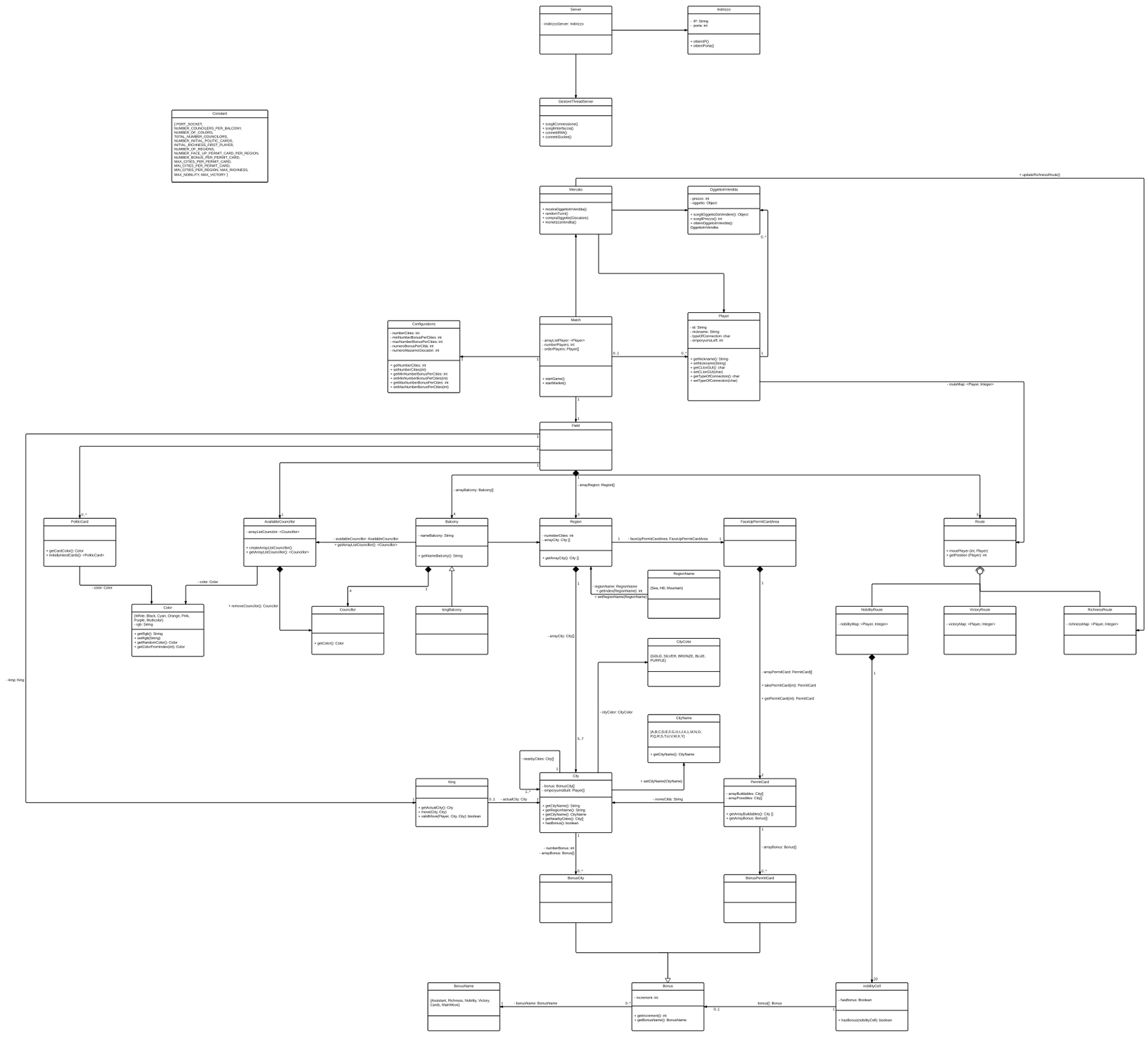
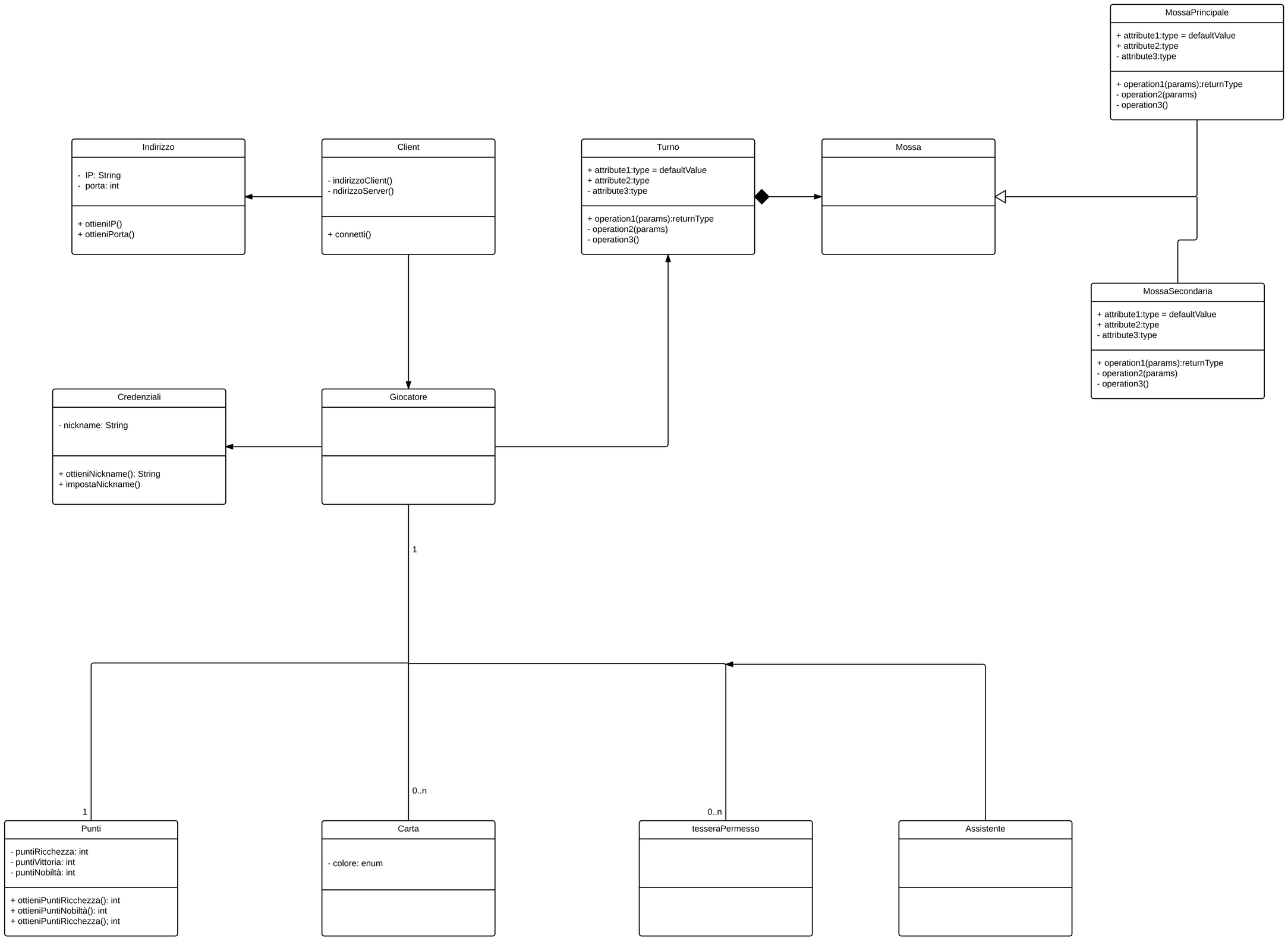

Server
- indirizzoServer: Indirizzo
Indirizzo
- IP: String
- porta: int
+ ottieniIPO
+ ottieniPorta()
GestoreThreadServer
+ scegliConnessione()
+ scegliInterfaccia()
+ connettiRMI()
+ connettiSocket()
Constant
{ PORT_SOCKET,
NUMBER_COUNCILORS_PER_BALCONY,
NUMBER_OF_COLORS,
TOTAL_NUMBER_COUNCILORS,
NUMBER_INITIAL_POLITIC_CARDS,
INITIAL_RICHNESS, FIRST_PLAYER,
NUMBER_OF_REGIONS,
NUMBER_FACE_UP_PERMIT_CARD_PER_REGION,
NUMBER_BONUS_PER_PERMIT_CARD,
MAX_CITIES_PER_PERMIT_CARD,
MIN_CITIES_PER_REGION, MAX_RICHNESS,
MAX_NOBILITY, MAX_VICTORY }
+ updateRichnessRoute()
Mercato
+ mostraOggettiInVendita()
+ randomTurni()
+ compraOggettiVGiocatore()
+ monetizzaVendita()
OggettiInVendita
- prezzo: int
- oggetto: Object
+ scegliOggettoDaVendere(): Object
+ scegliPrezzo(): int
+ ottieniOggettiInVendita(): OggettiInVendita
0..*
Configurations
- numberCities: int
- minNumberBonusPerCities: int
- maxNumberBonusPerCities: int
- numeroBonusPerCitta: int
- numeroMassimoGiocatori: int
+ getNumberCities: int
+ setNumberCities(int)
+ getMinNumberBonusPerCities: int
+ setMinNumberBonusPerCities(int)
+ getMaxNumberBonusPerCities: int
+ setMaxNumberBonusPerCities(int)
Match
- arrayListPlayer: <Player>
- numberPlayers: int
- orderPlayers: Player[]
+ startGame()
+ startMarket()
0..1
0..*
Player
- id: String
- nickname: String
- typeOfConnection: char
- emporiumsLeft: int
+ getNickname(): String
+ setNickname(String)
+ getCLorGUI(): char
+ setCLorGUI(char)
+ getTypeOfConnection(): char
+ setTypeOfConnection(char)
1
- routeMap: <Player, Integer>
1
1
Field
1
1
1
- arrayBalcony: Balcony[]
- arrayRegion: Region[]
PoliticCard
+ getCardColor(): Color
+ initially4HandCards(): <PoliticCard>
0..*
- color: Color
AvailableCouncillor
- arrayListCouncillor: <Councillor>
+ createArrayListCouncillor()
+ getArrayListCouncillor(): <Councillor>
- availableCouncillor: AvailableCouncillor
+ getArrayListCouncillor(): <Councillor>
Balcony
- nameBalcony: String
+ getNameBalcony(): String
Region
- numeberCities: int
- arrayCity: City []
+ getArrayCity(): City []
1
- faceUpPermitCardArea: FaceUpPermitCardArea
1
FaceUpPermitCardArea
Route
+ movePlayer (int, Player)
+ getPosition (Player): int
- color: Color
Color
{White, Black, Cyan, Orange, Pink, Purple, Multicolor}
- rgb: String
+ getRgb(): String
+ setRgb(String)
+ getRandomColor(): Color
+ getColorFromIndex(int): Color
+ removeCouncillor(): Councillor
Councillor
+ getColor(): Color
kingBalcony
- regionName: RegionName
+ getIndex(RegionName): int
+ setRegionName(RegionName)
RegionName
{Sea, Hill, Mountain}
NobilityRoute
- nobilityMap: <Player, Integer>
VictoryRoute
- victoryMap: <Player, Integer>
RichnessRoute
- richnessMap: <Player, Integer>
- arrayCity: City[]
CityColor
{GOLD, SILVER, BRONZE, BLUE, PURPLE}
- arrayPermitCard: PermitCard[]
+ takePermitCard(int): PermitCard
+ getPermitCard(int): PermitCard
- cityColor: CityColor
CityName
{A,B,C,D,E,F,G,H,I,J,K,L,M,N,O, P,Q,R,S,T,U,V,W,X,Y}
+ getCityName(): CityName
- king: King
- nearbyCities: City[]
King
+ getActualCity(): City
+ move(City, City)
+ validMove(Player, City, City): boolean
0..1
- actualCity: City
City
- bonus: BonusCity[]
- emporiumBuilt: Player[]
+ getCityName(): String
+ getRegionName(): String
+ getCityName(): CityName
+ getNearbyCities(): City[]
+ hasBonus(): boolean
+ setCityName(CityName)
- nomeCitta: String
PermitCard
- arrayBuildables: City[]
- arrayPossibles: City[]
+ getArrayBuildables(): City []
+ getArrayBonus: Bonus[]
- arrayBonus: Bonus[]
- numberBonus: int
- arrayBonus: Bonus[]
BonusCity
BonusPermitCard
BonusName
{Assistant, Richness, Nobility, Victory, Cards, MainMove}
- bonusName: BonusName
Bonus
- increment: int
+ getIncrement(): int
+ getBonusName(): BonusName
bonus[]: Bonus
nobilityCell
- hasBonus: Boolean
+ hasBonus(nobilityCell): boolean


MossaPrincipale
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Indirizzo
- IP: String
- porta: int
+ ottieniIP()
+ ottieniPorta()
Client
- indirizzoClient()
- ndirizzoServer()
+ connetti()
Turno
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Mossa
MossaSecondaria
+ attribute1:type = defaultValue
+ attribute2:type
- attribute3:type
+ operation1(params):returnType
- operation2(params)
- operation3()
Credenziali
- nickname: String
+ ottieniNickname(): String
+ impostaNickname()
Giocatore
1
1
0..n
0..n
Punti
- puntiRicchezza: int
- puntiVittoria: int
- puntiNobiltà: int
+ ottieniPuntiRicchezza(): int
+ ottieniPuntiNobiltà(): int
+ ottieniPuntiRicchezza(); int
Carta
- colore: enum
tesseraPermesso
Assistente