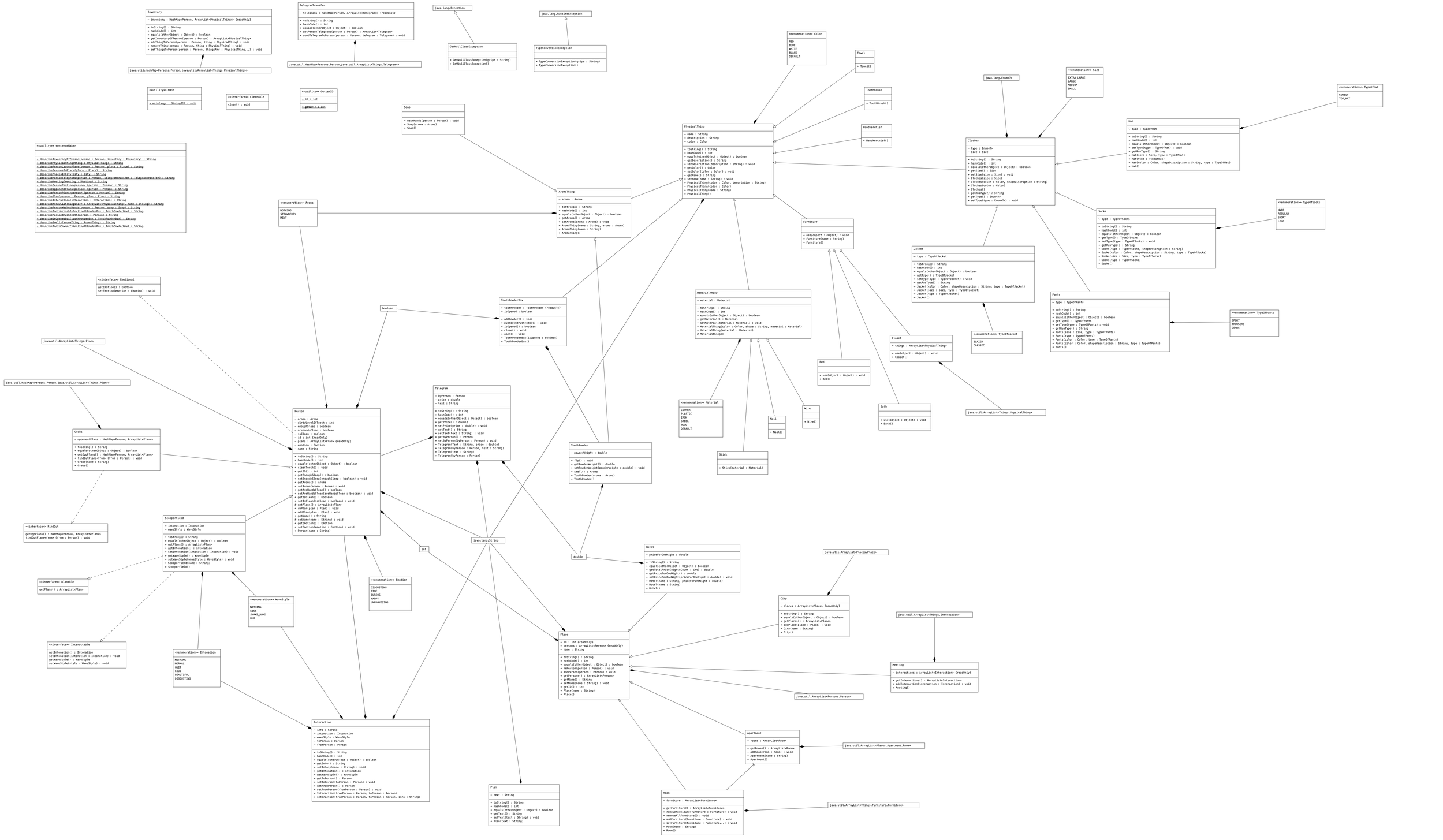
Inventory
- inventory : HashMap<Person, ArrayList<PhysicalThing>> {readOnly}
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getInventoryOfPerson(person : Person) : ArrayList<PhysicalThing>
+ addThingToPerson(person : Person, thing : PhysicalThing) : void
+ removeThing(person : Person, thing : PhysicalThing) : void
+ setThingsToPerson(person : Person, thingsArr : PhysicalThing...) : void
java.util.HashMap<Persons.Person,java.util.ArrayList<Things.PhysicalThing>>
TelegramTransfer
- telegrams : HashMap<Person, ArrayList<Telegram>> {readOnly}
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getPersonTelegrams(person : Person) : ArrayList<Telegram>
+ sendTelegramToPerson(person : Person, telegram : Telegram) : void
java.util.HashMap<Persons.Person,java.util.ArrayList<Things.Telegram>>
java.lang.Exception
GetNullClassException
+ GetNullClassException(gripe : String)
+ GetNullClassException()
java.lang.RuntimeException
TypeConversionException
+ TypeConversionException(gripe : String)
+ TypeConversionException()
<<enumeration>> Color
RED
BLUE
WHITE
BLACK
DEFAULT
Towel
+ Towel()
ToothBrush
+ ToothBrush()
HandkerchiefC
+ HandkerchiefC()
<<utility>> Main
+ main(args : String[]) : void
<<interface>> Cleanable
+ clean() : void
<<utility>> GetterID
- id : int
+ getID() : int
Soap
+ washHands(person : Person) : void
+ Soap(aroma : Aroma)
+ Soap()
PhysicalThing
- name : String
- description : String
- color : Color
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getDescription() : String
+ setDescription(description : String) : void
+ getColor() : Color
+ setColor(color : Color) : void
+ getName() : String
+ setName(name : String) : void
+ PhysicalThing(color : Color, description : String)
+ PhysicalThing(color : Color)
+ PhysicalThing(name : String)
+ PhysicalThing()
java.lang.Enum<?>
<<enumeration>> Size
EXTRA_LARGE
LARGE
MEDIUM
SMALL
<<enumeration>> TypeOfHat
COWBOY
TOP_HAT
Hat
- type : TypeOfHat
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ setType(type : TypeOfHat) : void
+ getRusType() : String
+ HatSize (Size, type : TypeOfHat)
+ Hat(type : TypeOfHat)
+ Hat(color : Color, shapeDiscription : String, type : TypeOfHat)
+ Hat()
Clothes
- type : Enum<?>
- size : Size
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getSize() : Size
+ Clothes(size : Size) : void
+ Clothes(size : Size)
+ Clothes(color : Color, shapeDiscription : String)
+ Clothes(color : Color)
+ Clothes()
+ getRusType() : String
+ getType() : Enum<?>
+ setType(type : Enum<?>) : void
AromaThing
- aroma : Aroma
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getAroma() : Aroma
+ setAroma(aroma : Aroma) : void
+ AromaThing(name : String, aroma : Aroma)
+ AromaThing(name : String)
+ AromaThing()
<<enumeration>> Aroma
NOTHING
STRAWBERRY
MINT
Furniture
+ use(object : Object) : void
+ Furniture(name : String)
+ Furniture()
Socks
- type : TypeOfSocks
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getType() : TypeOfSocks
+ setType(type : TypeOfSocks) : void
+ getRusType() : String
+ Socks(type : TypeOfSocks, shapeDescription : String)
+ Socks(color : Color, shapeDescription : String, type : TypeOfSocks)
+ Socks(size : Size, type : TypeOfSocks)
+ Socks(type : TypeOfSocks)
+ Socks()
<<enumeration>> TypeOfSocks
KNEE
REGULAR
SHORT
LONG
Jacket
- type : TypeOfJacket
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getType() : TypeOfJacket
+ setType(type : TypeOfJacket) : void
+ getRusType() : String
+ Jacket(color : Color, shapeDescription : String, type : TypeOfJacket)
+ Jacket(size : Size, type : TypeOfJacket)
+ Jacket(type : TypeOfJacket)
+ Jacket()
<<interface>> Emotional
getEmotion() : Emotion
setEmotion(emotion : Emotion) : void
ToothPowderBox
- toothPowder : ToothPowder {readOnly}
- isOpened : boolean
+ addPowder() : void
+ putToothBrushToBox() : void
+ isOpened() : boolean
+ close() : void
+ open() : void
+ ToothPowderBox(isOpened : boolean)
+ ToothPowderBox()
boolean
MaterialThing
- material : Material
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getMaterial() : Material
+ setMaterial(material : Material) : void
+ MaterialThing(color : Color, shape : String, material : Material)
+ MaterialThing(material : Material)
# MaterialThing()
Pants
- type : TypeOfPants
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getType() : TypeOfPants
+ setType(type : TypeOfPants) : void
+ getRusType() : String
+ PantsSize (Size, type : TypeOfPants)
+ Pants(type : TypeOfPants)
+ Pants(color : Color, type : TypeOfPants)
+ Pants(color : Color, shapeDescription : String, type : TypeOfPants)
+ Pants()
<<enumeration>> TypeOfPants
SPORT
TROUSERS
JEWS
Closet
- things : ArrayList<PhysicalThing>
+ use(object : Object) : void
+ Closet()
<<enumeration>> TypeOfJacket
BLAZER
CLASSIC
Bed
+ use(object : Object) : void
+ Bed()
java.util.ArrayList<Things.PhysicalThing>
Bath
+ use(object : Object) : void
+ Bath()
<<enumeration>> Material
COPPER
PLASTIC
GLASS
STEEL
WOOD
DEFAULT
Wire
+ Wire()
Nail
+ Nail()
Stick
+ Stick(material : Material)
java.util.ArrayList<Things.Plan>
java.util.HashMap<Persons.Person,java.util.ArrayList<Things.Plan>>
Crabs
- opponentPlans : HashMap<Person, ArrayList<Plan>>
+ toString() : String
+ equals(otherObject : Object) : boolean
+ getOpPlans() : HashMap<Person, ArrayList<Plan>>
+ findOutPlansFrom(-from : Person) : void
+ Crabs(name : String)
+ Crabs()
Person
- aroma : Aroma
- dirtyLevelOfTeeth : int
- enoughSleep : boolean
- arrMendsClean : boolean
- isClean : boolean
- id : int {readOnly}
- plans : ArrayList<Plan> {readOnly}
- emotion : Emotion
- name : String
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ cleanTeeth() : void
+ getID() : int
+ getEnoughSleep() : boolean
+ setEnoughSleep(enoughSleep : boolean) : void
+ getAroma() : Aroma
+ setAroma(aroma : Aroma) : void
+ getArrMendsClean() : boolean
+ setArrMendsClean(arrMendsClean : boolean) : void
+ getIsClean() : boolean
+ setIsClean(isClean : boolean) : void
+ getPlans() : ArrayList<Plan>
+ rmPlan(plan : Plan) : void
+ addPlan(plan : Plan) : void
+ getName() : String
+ setName(name : String) : void
+ getEmotion() : Emotion
+ setEmotion(emotion : Emotion) : void
+ Person(name : String)
Telegram
- byPerson : Person
- price : double
- text : String
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getPrice() : double
+ setPrice(price : double) : void
+ getText() : String
+ setText(text : String) : void
+ getByPerson() : Person
+ setByPerson(byPerson : Person) : void
+ Telegram(Text : String, price : double)
+ Telegram(byPerson : Person, text : String)
+ Telegram(text : String)
+ Telegram(byPerson : Person)
ToothPowder
- powderWeight : double
+ fly() : void
+ getPowderWeight() : double
+ setPowderWeight(powderWeight : double) : void
+ smell() : Aroma
+ ToothPowder(aroma : Aroma)
+ ToothPowder()
<<interface>> FindOut
getOpPlans() : HashMap<Person, ArrayList<Plan>>
findOutPlansFrom(-from : Person) : void
Scooperfield
- intonation : Intonation
- waveStyle : WaveStyle
+ toString() : String
+ equals(otherObject : Object) : boolean
+ getPlans() : ArrayList<Plan>
+ getIntonation() : Intonation
+ setIntonation(intonation : Intonation) : void
+ getWaveStyle() : WaveStyle
+ setWaveStyle(waveStyle : WaveStyle) : void
+ Scooperfield(name : String)
+ Scooperfield()
int
java.lang.String
double
Hotel
- priceForOneNight : double
+ toString() : String
+ equals(otherObject : Object) : boolean
+ getTotalPrice(nightsCount : int) : double
+ getPriceForOneNight() : double
+ setPriceForOneNight(priceForOneNight : double) : void
+ Hotel(name : String, priceForOneNight : double)
+ Hotel(name : String)
+ Hotel()
java.util.ArrayList<Places.Place>
City
- places : ArrayList<Place> {readOnly}
+ toString() : String
+ equals(otherObject : Object) : boolean
+ getPlaces() : ArrayList<Place>
+ addPlace(place : Place) : void
+ City(name : String)
+ City()
<<interface>> Blabable
getPlans() : ArrayList<Plan>
<<enumeration>> WaveStyle
NOTHING
KISS
SHAKE_HAND
HUG
<<enumeration>> Emotion
DISGUSTING
FINE
CURIOS
HAPPY
UNPROMISING
java.util.ArrayList<Things.Interaction>
Place
- id : int {readOnly}
- persons : ArrayList<Person> {readOnly}
- name : String
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ rmPerson(person : Person) : void
+ addPerson(person : Person) : void
+ getPersons() : ArrayList<Person>
+ getName() : String
+ setName(name : String) : void
+ getID() : int
+ Place(name : String)
+ Place()
Meeting
- interactions : ArrayList<Interaction> {readOnly}
+ getInteractions() : ArrayList<Interaction>
+ addInteraction(interaction : Interaction) : void
+ Meeting()
<<interface>> Interactable
getIntonation() : Intonation
setIntonation(intonation : Intonation) : void
getWaveStyle() : WaveStyle
setWaveStyle(style : WaveStyle) : void
<<enumeration>> Intonation
NOTHING
NORMAL
KISS
LOOD
BEAUTIFUL
DISGUSTING
java.util.ArrayList<Persons.Person>
Interaction
- info : String
- intonation : Intonation
- waveStyle : WaveStyle
- toPerson : Person
- fromPerson : Person
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getInfo() : String
+ setInfo(phrase : String) : void
+ getIntonation() : Intonation
+ getWaveStyle() : WaveStyle
+ getToPerson() : Person
+ setToPerson(toPerson : Person) : void
+ getFromPerson() : Person
+ setFromPerson(fromPerson : Person) : void
+ Interaction(fromPerson : Person, toPerson : Person)
+ Interaction(fromPerson : Person, toPerson : Person, info : String)
Apartment
- rooms : ArrayList<Room>
+ getRooms() : ArrayList<Room>
+ addRoom(room : Room) : void
+ Apartment(name : String)
+ Apartment()
java.util.ArrayList<Places.Apartment.Room>
Plan
- text : String
+ toString() : String
+ hashCode() : int
+ equals(otherObject : Object) : boolean
+ getText() : String
+ setText(text : String) : void
+ Plan(text : String)
Room
- furniture : ArrayList<Furniture>
+ getFurniture() : ArrayList<Furniture>
+ removeFurniture(furniture : Furniture) : void
+ removeAllFurniture() : void
+ addFurniture(furniture : Furniture) : void
+ setFurniture(furniture : Furniture...) : void
+ Room(name : String)
+ Room()
java.util.ArrayList<Things.Furniture.Furniture>