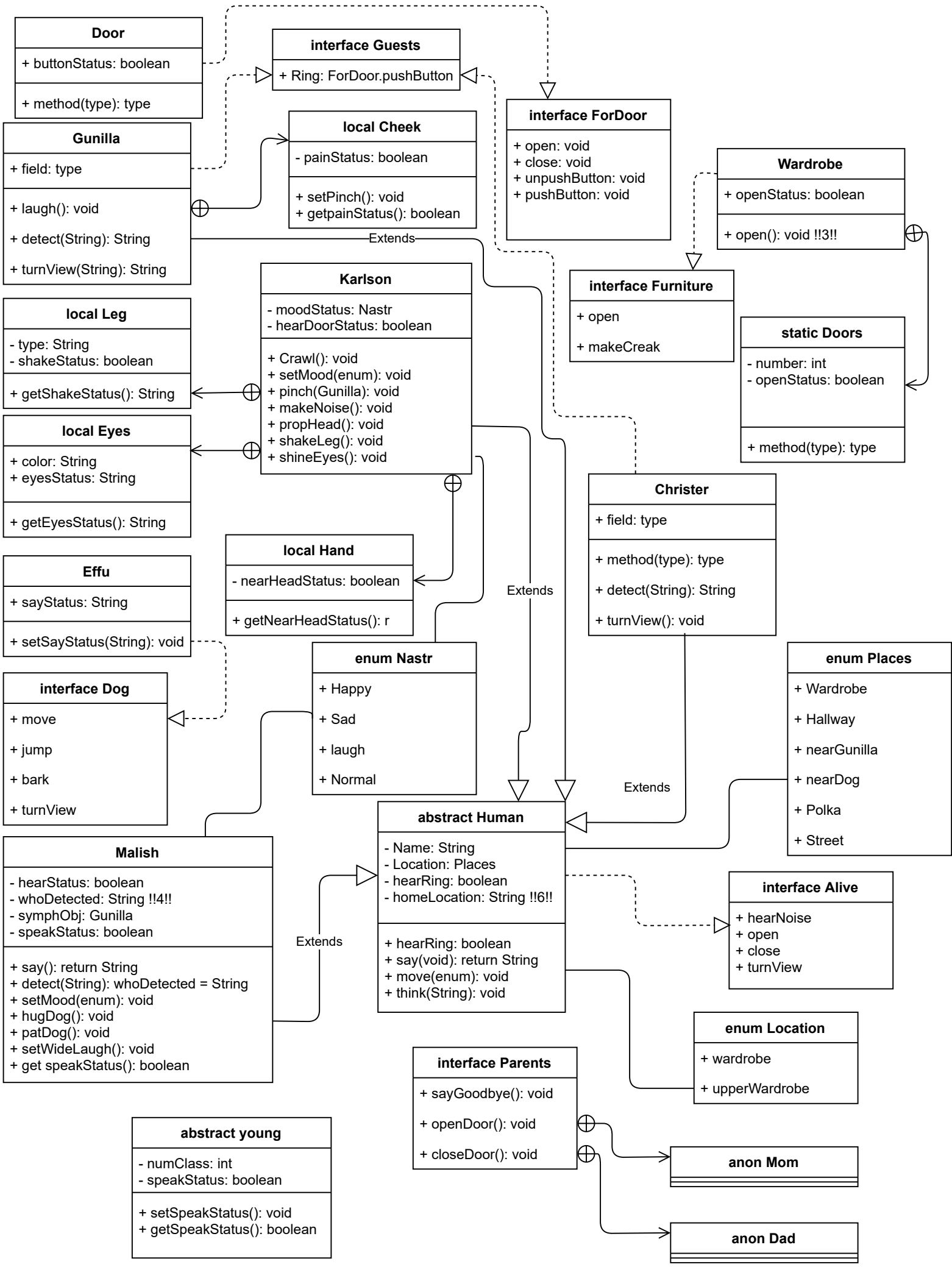
Door
+ buttonStatus: boolean
+ method(type): type
interface Guests
+ Ring: ForDoor.pushButton
interface ForDoor
+ open: void
+ close: void
+ unpushButton: void
+ pushButton: void
Gunilla
+ field: type
+ laugh(): void
+ detect(String): String
+ turnView(String): String
local Cheek
- painStatus: boolean
+ setPinch(): void
+ getpainStatus(): boolean
Extends
Wardrobe
+ openStatus: boolean
+ open(): void !!3!!
interface Furniture
+ open
+ makeCreak
Karlson
- moodStatus: Nastr
- hearDoorStatus: boolean
+ Crawl(): void
+ setMood(enum): void
+ pinch(Gunilla): void
+ makeNoise(): void
+ propHead(): void
+ shakeLeg(): void
+ shineEyes(): void
local Leg
- type: String
- shakeStatus: boolean
+ getShakeStatus(): String
static Doors
- number: int
- openStatus: boolean
+ method(type): type
local Eyes
+ color: String
+ eyesStatus: String
+ getEyesStatus(): String
Christer
+ field: type
+ method(type): type
+ detect(String): String
+ turnView(): void
local Hand
- nearHeadStatus: boolean
+ getNearHeadStatus(): r
Effu
+ sayStatus: String
+ setSayStatus(String): void
Extends
enum Nastr
+ Happy
+ Sad
+ laugh
+ Normal
enum Places
+ Wardrobe
+ Hallway
+ nearGunilla
+ nearDog
+ Polka
+ Street
interface Dog
+ move
+ jump
+ bark
+ turnView
Extends
abstract Human
- Name: String
- Location: Places
- hearRing: boolean
- homeLocation: String !!6!!
+ hearRing: boolean
+ say(void): return String
+ move(enum): void
+ think(String): void
Malish
- hearStatus: boolean
- whoDetected: String !!4!!
- symphObj: Gunilla
- speakStatus: boolean
+ say(): return String
+ detect(String): whoDetected = String
+ setMood(enum): void
+ hugDog(): void
+ patDog(): void
+ setWideLaugh(): void
+ get speakStatus(): boolean
Extends
interface Alive
+ hearNoise
+ open
+ close
+ turnView
enum Location
+ wardrobe
+ upperWardrobe
interface Parents
+ sayGoodbye(): void
+ openDoor(): void
+ closeDoor(): void
abstract young
- numClass: int
- speakStatus: boolean
+ setSpeakStatus(): void
+ getSpeakStatus(): boolean
anon Mom
anon Dad