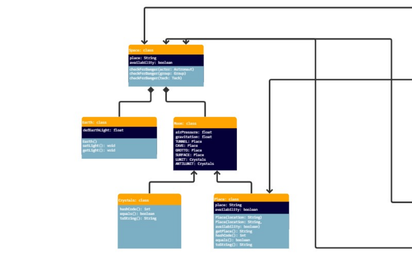
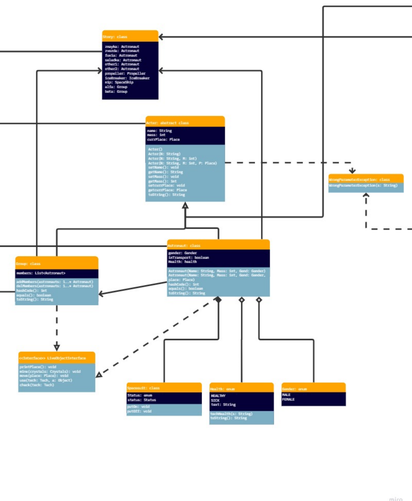
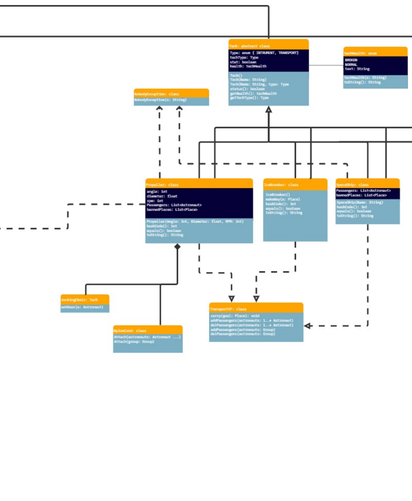

Guide::class
place: String
availability: boolean
checkAvailability(): boolean
checkAttendance(): void
checkAttendanceCheck: Tuple
Walk::class
authorNight: Elief
Walk()
getNight(): void
getLight(): void
Bus::class
arrivesoon: Elief
Bus(): void
NAME: Place
LOC: Place
NAME: Place
LOC: Place
authority: Place
Bus(): void
authorNight: Elevate
Transfer::class
hasDone(): bet
equal(): boolean
toString(): String
Plane::class
place: String
availability: boolean
Plane(location: String)
checkAvailability(): boolean
checkAttendance(): boolean
getTime(): String
getPlace(): Place
equal(): boolean
toString(): String


miro
Group::class
result: Activeset
results: Activeset
Activeset: Activeset
activeSet: Activeset
offset: Activeset
Group(): void
groupClass: PropertyClass
transformer: IsOtherMember
Group(): void
wife: Group
husband: Group
Person::class
gender: Gender
firstname: FirstName
Health: Health
Person(name: String, Mass: int, Gender: Gender): void
Person(name: String, Mass: int, Gender: Gender, Activeset:Person): void
getHealth(): int
hasDone(): bet
equal(): boolean
toString(): String
PropertyClass::class
mathmen: List<ActiveSet>
getAllMembers(): List<ActiveSet>
getMembersFromHusband(): List<ActiveSet>
hasDone(): bet
equal(): boolean
toString(): String
IsOtherMember::class
getAllFamily(): void
getAllFamilyMembers(): void
getMembersFromHusband(): void
hasDone(): bet
equal(): boolean
toString(): String
IsOtherMember::class
Status: enum
Status: Status
Status(): void
getHealth(): int
Gender::class
MALE
FEMALE
FirstNameOfPerson::class
name: int
Mass: int
Gender: Gender
FirstNameOfPerson(name: String, Mass: int, Gender: Gender): void
getMass(): int
getGender(): Gender
equal(): boolean
toString(): String


miro
Type::class
Type: enum { INTEGER, TUNDIFIVE}
typeTypes: Type
typeTypeOfTypes: Type
Type: TuneDeath
Type()
TuneDeath: String
TuneDeath: String, Type: Type
getTypeOfTypes(): Type
getTuneDeath(): TuneDeath
getTheTypeOf(): Type
Type::class
INTEGER
TUNDIFIVE
Type: String
toString(): String
AbstractCalcFor::class
AbstractCalcFor(): void
Person::class
Person(name: String)
LocationOwner::class
LocationOwner(): void
AddMember: Place[]
hasDone(): bet
equal(): boolean
toString(): String
PersonProgram::class
PersonProgram: List<ActiveSet>
SomeFiles: List<File>
PersonProgram(): void
PersonProgram(): void
hasDone(): bet
equal(): boolean
toString(): String
ArrayTest::class
arrayTest: int
setSet(): Activeset[]
MyMethod::class
MyMethod(): void
MyMethod(setSet: Activeset ...)
MyMethod(group: Group)
ArrayTestOfFr::class
arrayTest: Place[]: void
setPassengers(setSet: Activeset ...): void
setPassengers(setSet: Activeset ...): void
setPassengers(setSet: Activeset): void
setPassengers(setSet: Activeset): void
setPassengers(setSet: Group)

miro
WeightCountOverlyn::class
WeightCountOver()
hasDone(): bet
equal(): boolean
toString(): String
hasDone(over: Activeset, group: Group, buffer: bet)
hasDone(over: Activeset, group: Group, buffer: bet)