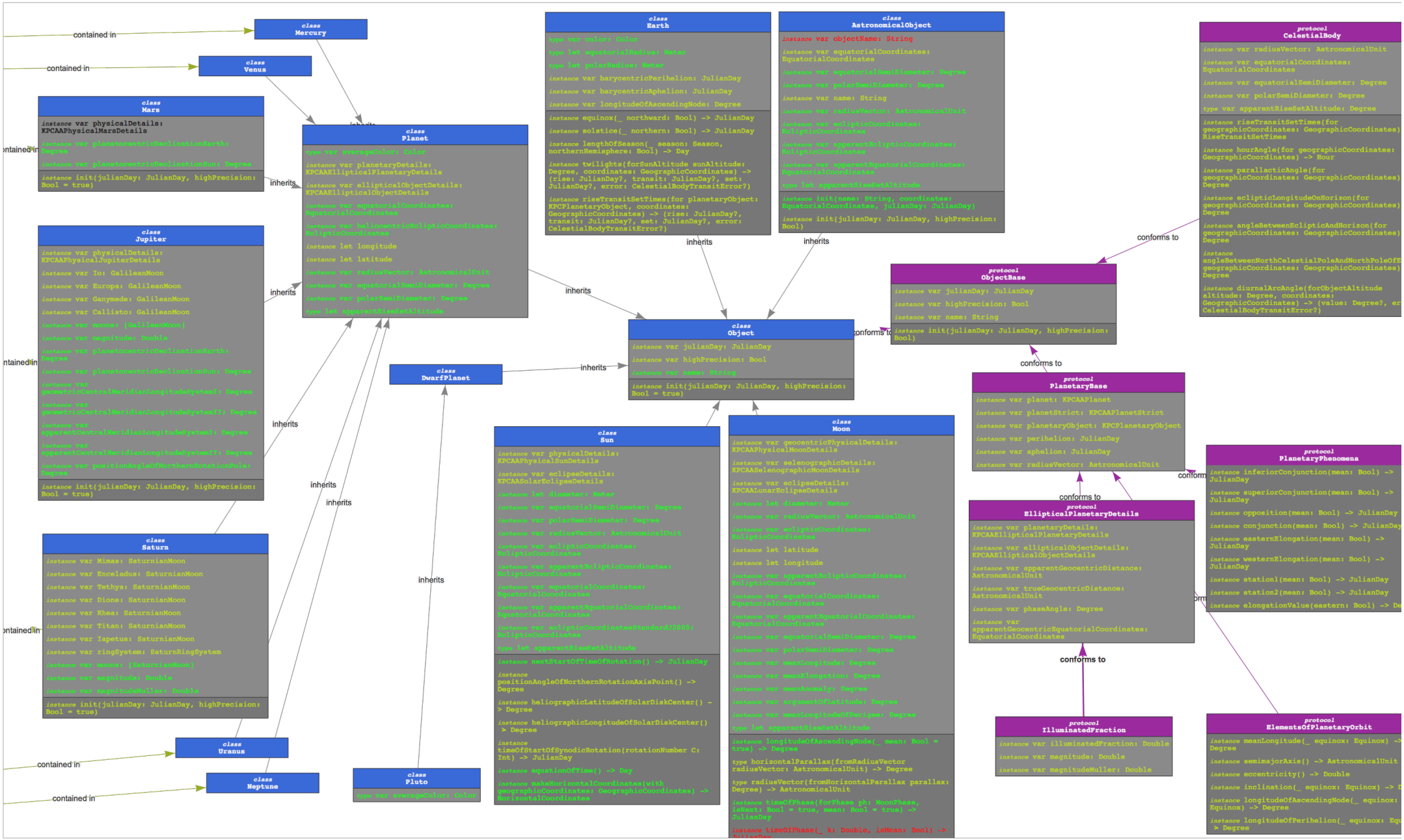
class Mercury
class Venus
class Mars
instance var physicalDetails: KPCAAPhysicalMarsDetails
instance var planetocentricDeclinationEarth: Degree
instance var planetocentricDeclinationSun: Degree
instance init(julianDay: JulianDay, highPrecision: Bool = true)
class Jupiter
instance var physicalDetails: KPCAAPhysicalJupiterDetails
instance var Io: GalileanMoon
instance var Europa: GalileanMoon
instance var Ganymede: GalileanMoon
instance var Callisto: GalileanMoon
instance var moons: [GalileanMoon]
instance var magnitude: Double
instance var planetocentricDeclinationEarth: Degree
instance var planetocentricDeclinationSun: Degree
instance var geocentricCentralMeridianLongitudeSystemI: Degree
instance var geocentricCentralMeridianLongitudeSystemII: Degree
instance var apparentCentralMeridianLongitudeSystemI: Degree
instance var apparentCentralMeridianLongitudeSystemII: Degree
instance var positionAngleOfNorthernRotationPole: Degree
instance init(julianDay: JulianDay, highPrecision: Bool = true)
class Saturn
instance var Mimas: SaturnianMoon
instance var Enceladus: SaturnianMoon
instance var Tethys: SaturnianMoon
instance var Dione: SaturnianMoon
instance var Rhea: SaturnianMoon
instance var Titan: SaturnianMoon
instance var Iapetus: SaturnianMoon
instance var ringSystem: SaturnRingSystem
instance var moons: [SaturnianMoon]
instance var magnitude: Double
instance var magnitudeMuller: Double
instance init(julianDay: JulianDay, highPrecision: Bool = true)
class Uranus
class Neptune
class Planet
type var averageColor: Color
instance var planetaryDetails: KPCAAEllipticalPlanetaryDetails
instance var ellipticalObjectDetails: KPCAAEllipticalObjectDetails
instance var equatorialCoordinates: EquatorialCoordinates
instance var heliocentricEclipticCoordinates: EclipticCoordinates
instance let longitude
instance let latitude
instance var radiusVector: AstronomicalUnit
instance var equatorialSemiDiameter: Degree
instance var polarSemiDiameter: Degree
type let apparentRiseSetAltitude
class DwarfPlanet
class Pluto
type var averageColor: Color
class Earth
type var color: Color
type let equatorialRadius: Meter
type let polarRadius: Meter
instance var barycentricPerihelion: JulianDay
instance var barycentricAphelion: JulianDay
instance var longitudeOfAscendingNode: Degree
instance equinox(_ northward: Bool) -> JulianDay
instance solstice(_ northern: Bool) -> JulianDay
instance lengthOfSeason(_ season: Season, northernHemisphere: Bool) -> Day
instance twilights(forSunAltitude sunAltitude: Degree, coordinates: GeographicCoordinates) -> (rise: JulianDay?, transit: JulianDay?, set: JulianDay?, error: CelestialBodyTransitError?)
instance riseTransitSetTimes(for planetaryObject: KPCPlanetaryObject, coordinates: GeographicCoordinates) -> (rise: JulianDay?, transit: JulianDay?, set: JulianDay?, error: CelestialBodyTransitError?)
class AstronomicalObject
instance var objectName: String
instance var equatorialCoordinates: EquatorialCoordinates
instance var equatorialSemiDiameter: Degree
instance var polarSemiDiameter: Degree
instance var name: String
instance var radiusVector: AstronomicalUnit
instance var eclipticCoordinates: EclipticCoordinates
instance var apparentEclipticCoordinates: EclipticCoordinates
instance var apparentEquatorialCoordinates: EquatorialCoordinates
type let apparentRiseSetAltitude
instance init(name: String, coordinates: EquatorialCoordinates, julianDay: JulianDay)
instance init(julianDay: JulianDay, highPrecision: Bool)
class Object
instance var julianDay: JulianDay
instance var highPrecision: Bool
instance var name: String
instance init(julianDay: JulianDay, highPrecision: Bool = true)
class Sun
instance var physicalDetails: KPCAAPhysicalSunDetails
instance var eclipseDetails: KPCAASolarEclipseDetails
instance let diameter: Meter
instance var equatorialSemiDiameter: Degree
instance var polarSemiDiameter: Degree
instance var radiusVector: AstronomicalUnit
instance var eclipticCoordinates: EclipticCoordinates
instance var apparentEclipticCoordinates: EclipticCoordinates
instance var equatorialCoordinates: EquatorialCoordinates
instance var apparentEquatorialCoordinates: EquatorialCoordinates
instance var eclipticCoordinatesStandardE2000: EclipticCoordinates
type let apparentRiseSetAltitude
instance meanStartOfTimeOfRotation() -> JulianDay
instance positionAngleOfNorthernRotationAxisPoint() -> Degree
instance heliographicLatitudeOfSolarDiskCenter() -> Degree
instance heliographicLongitudeOfSolarDiskCenter() -> Degree
instance timeOfStartOfSynodicRotation(rotationNumber C: Int) -> JulianDay
instance equationOfTime() -> Day
instance makeHorizontalCoordinates(with geographicCoordinates: GeographicCoordinates) -> HorizontalCoordinates
class Moon
instance var geocentricPhysicalDetails: KPCAAPhysicalMoonDetails
instance var selenographicDetails: KPCAASelenographicMoonDetails
instance var eclipseDetails: KPCAALunarEclipseDetails
instance let diameter: Meter
instance var radiusVector: AstronomicalUnit
instance var eclipticCoordinates: EclipticCoordinates
instance let latitude
instance let longitude
instance var apparentEclipticCoordinates: EclipticCoordinates
instance var equatorialCoordinates: EquatorialCoordinates
instance var apparentEquatorialCoordinates: EquatorialCoordinates
instance var equatorialSemiDiameter: Degree
instance var polarSemiDiameter: Degree
instance var meanLongitude: Degree
instance var meanElongation: Degree
instance var meanAnomaly: Degree
instance var argumentOfLatitude: Degree
instance var meanLongitudeOfPerigee: Degree
type let apparentRiseSetAltitude
instance longitudeOfAscendingNode(_ mean: Bool = true) -> Degree
type horizontalParallax(fromRadiusVector radiusVector: AstronomicalUnit) -> Degree
type radiusVector(fromHorizontalParallax parallax: Degree) -> AstronomicalUnit
instance timeOfPhase(forPhase pk: MoonPhase, isNext: Bool = true, mean: Bool = true) -> JulianDay
instance timeOfPhase(_ k: Double, isMean: Bool) -> JulianDay
protocol CelestialBody
instance var radiusVector: AstronomicalUnit
instance var equatorialCoordinates: EquatorialCoordinates
instance var equatorialSemiDiameter: Degree
instance var polarSemiDiameter: Degree
type var apparentRiseSetAltitude: Degree
instance riseTransitSetTimes(for geographicCoordinates: GeographicCoordinates) RiseTransitSetTimes
instance hourAngle(for geographicCoordinates: GeographicCoordinates) -> Hour
instance parallacticAngle(for geographicCoordinates: GeographicCoordinates) Degree
instance eclipticLongitudeOnHorizon(for geographicCoordinates: GeographicCoordinates) Degree
instance angleBetweenEclipticAndHorizon(for geographicCoordinates: GeographicCoordinates) Degree
instance angleBetweenNorthCelestialPoleAndNorthPoleOfE geographicCoordinates: GeographicCoordinates) Degree
instance diurnalArcAngle(forObjectAltitude altitude: Degree, coordinates: GeographicCoordinates) -> (value: Degree?, er CelestialBodyTransitError?)
protocol ObjectBase
instance var julianDay: JulianDay
instance var highPrecision: Bool
instance var name: String
instance init(julianDay: JulianDay, highPrecision: Bool)
protocol PlanetaryBase
instance var planet: KPCAAPlanet
instance var planetStrict: KPCAAPlanetStrict
instance var planetaryObject: KPCPlanetaryObject
instance var perihelion: JulianDay
instance var aphelion: JulianDay
instance var radiusVector: AstronomicalUnit
protocol PlanetaryPhenomena
instance inferiorConjunction(mean: Bool) -> JulianDay
instance superiorConjunction(mean: Bool) -> JulianDay
instance opposition(mean: Bool) -> JulianDay
instance conjunction(mean: Bool) -> JulianDay
instance easternElongation(mean: Bool) -> JulianDay
instance westernElongation(mean: Bool) -> JulianDay
instance station1(mean: Bool) -> JulianDay
instance station2(mean: Bool) -> JulianDay
instance elongationValue(eastern: Bool) -> De
protocol EllipticalPlanetaryDetails
instance var planetaryDetails: KPCAAEllipticalPlanetaryDetails
instance var ellipticalObjectDetails: KPCAAEllipticalObjectDetails
instance var apparentGeocentricDistance: AstronomicalUnit
instance var trueGeocentricDistance: AstronomicalUnit
instance var phaseAngle: Degree
instance var apparentGeocentricEquatorialCoordinates: EquatorialCoordinates
protocol IlluminatedFraction
instance var illuminatedFraction: Double
instance var magnitude: Double
instance var magnitudeMuller: Double
protocol ElementsOfPlanetaryOrbit
instance meanLongitude(_ equinox: Equinox) -> Degree
instance semimajorAxis() -> AstronomicalUnit
instance eccentricity() -> Double
instance inclination(_ equinox: Equinox) -> D
instance longitudeOfAscendingNode(_ equinox: Equinox) -> Degree
instance longitudeOfPerihelion(_ equinox: Equ -> Degree
contained in
inherits
conforms to