context OfferingCatalog::createOffering(type: String, availability: Availability)

pre: self.AvailabilityCatalog.availabilities->includes(availability) and not availability.used

post: self. offerings->exists(o: Offering | o.type=type and o.availability=availability)

context OfferingCatalog::selectOffering(offeringID: integer, ins: Instructor)

pre: self.offerings->exists(o: Offering | o.id=offeringID)

post: self.offerings->exists(o: Offering | o.id=offeringID and o.instructor=ins)

context OfferingCatalog::updateOffering(offeringID: integer, type: String)

pre: self.offerings->exists(o: Offering | o.id=offeringID)

post: self.offerings->exists(o: Offering | o.id=offeringID and o.type=type)

context OfferingCatalog::deleteOffering(offeringID: integer)

pre: self.offerings->exists(o: Offering | o.id=offeringID)

post: not self.offerings->exists(o: Offering | o.id=offeringID)

context BookingCatalog::bookOffering(offeringID: integer, client: Client)

pre: self.OfferingCatalog.offerings->exists(o: Offering | o.id=offeringID) and client.age >= 18

post: self.bookings->exists(b: Booking | b.offering.id=offeringID and b.client = client)

context BookingCatalog::bookOffering(offeringID: integer, client: Client, legalGuardian: Client)

pre: self.OfferingCatalog.offerings->exists(o: Offering | o.id=offeringID) and legalGuardian.age >= 18

post: self.bookings->exists(b: Booking | b.offering.id=offeringID and b.client = client and b.legalGuardian = legalGuardian)

context BookingCatalog::cancelOffering(bookingID: integer)

pre: self.bookings->exists(o: Booking | o.id=bookingID)

post: not self.bookings->exists(o: Booking | o.id=bookingID