

```

class FleetManagerFactory
{
    buildFleetManager(String, String, String, String, String, String) FleetManager
}

```

```

class FleetManagerFactoryTest
{
    FleetManagerTest String
    testBuildFleetManager() void
    setUp() void
    buildExpenses() void
}

```

```

class RentalTermFactory
{
    buildRentalTerm(String, String, String, String, String, String) RentalTerm
}

```

```

class RentalTermFactoryTest
{
    RentalTermTest String
    testBuildRentalTerm() void
    setUp() void
    buildRentalTerm() void
}

```

```

class VehiclePurchasePriceFactory
{
    buildVehiclePurchasePrice(String, String, String) VehiclePurchasePrice
}

```

```

class VehiclePurchasePriceFactoryTest
{
    VehiclePurchasePriceTest String
    testBuildVehiclePurchasePrice() void
    setUp() void
    buildVehiclePurchasePrice() void
}

```

```

class VehicleSellingPriceFactory
{
    buildVehicleSellingPrice(String, String, String) VehicleSellingPrice
}

```

```

class VehicleSellingPriceFactoryTest
{
    VehicleSellingPriceTest String
    testBuildVehicleSellingPrice() void
    setUp() void
    buildVehicleSellingPrice() void
}

```

```

class InfrastructureFactory
{
    buildInfrastructure(String, String, String, String, String) Infrastructure
}

```

```

class InfrastructureFactoryTest
{
    InfrastructureTest String
    testBuildInfrastructure() void
    setUp() void
    buildInfrastructure() void
}

```

```

class ExpensesFactory
{
    buildExpenses(String, String, String, String, String, String) Expenses
}

```

```

class ExpensesFactoryTest
{
    ExpensesTest String
    testBuildExpenses() void
    setUp() void
    buildExpenses() void
}

```

```

class ClientFactory
{
    buildClient(String, String, String, String, String, String, String) Client
}

```

```

class ClientFactoryTest
{
    ClientTest String
    testBuildClient() void
    setUp() void
    buildClient() void
}

```

```

class RevenueFactory
{
    buildRevenue(String, String, String, String, String, String) Revenue
}

```

```

class RevenueFactoryTest
{
    RevenueTest String
    testBuildRevenue() void
    setUp() void
    buildRevenue() void
}

```

```

class BranchLocationFactory
{
    buildBranchLocation(String, String, String) BranchLocation
}

```

```

class BranchLocationFactoryTest
{
    BranchLocationTest String
    testBuildBranchLocation() void
    setUp() void
    buildBranchLocation() void
}

```

```

class PerishablesFactory
{
    buildPerishables(String, String, String, String, String) Perishables
}

```

```

class PerishablesFactoryTest
{
    PerishablesTest String
    testBuildPerishables() void
    setUp() void
    buildPerishables() void
}

```

```

class SoftwareFactory
{
    buildSoftware(String, String, String, String, String) Software
}

```

```

class SoftwareFactoryTest
{
    SoftwareTest String
    testBuildSoftware() void
    setUp() void
    buildSoftware() void
}

```

```

class StaffFactory
{
    buildStaff(String, String, String, String, String, String) Staff
}

```

```

class StaffFactoryTest
{
    StaffTest String
    testBuildStaff() void
    setUp() void
    buildStaff() void
}

```

```

class SupplierFactory
{
    buildSupplier(String, String, String, String, String) Supplier
}

```

```

class SupplierFactoryTest
{
    SupplierTest String
    testBuildSupplier() void
    setUp() void
    buildSupplier() void
}

```

```

class RentalValueFactory
{
    buildRentalValue(String, String, String, String) RentalValue
}

```

```

class RentalValueFactoryTest
{
    RentalValueTest String
    testBuildRentalValue() void
    setUp() void
    buildRentalValue() void
}

```

```

class ContactFactory
{
    buildContact(String, String, String, String, String) Contact
}

```

```

class ContactFactoryTest
{
    ContactTest String
    testBuildContact() void
    setUp() void
    buildCar() void
}

```

```

class CarFactory
{
    buildCar(String, String, String, String, String, String) Car
}

```

```

class CarFactoryTest
{
    CarTest String
    testBuildCar() void
    setUp() void
    buildCar() void
}

```

```

class ClaimsFactory
{
    buildClaims(String, String, String, String, String) Claims
}

```

```

class ClaimsFactoryTest
{
    ClaimsTest String
    testBuildClaims() void
    setUp() void
    buildClaims() void
}

```

```

class StaffTypeFactory
{
    buildStaffType(String, String, String, String) StaffType
}

```

```

class StaffTypeFactoryTest
{
    StaffTypeTest String
    testBuildStaffType() void
    setUp() void
    buildStaffType() void
}

```

```

class HardwareFactory
{
    buildHardware(String, String, String, String) Hardware
}

```

```

class HardwareFactoryTest
{
    HardwareTest String
    testBuildHardware() void
    setUp() void
    buildHardware() void
}

```

```

class RegionFactory
{
    buildRegion(String, String, String, String) Region
}

```

```

class RegionFactoryTest
{
    RegionTest String
    testBuildRegion() void
    setUp() void
    buildRegion() void
}

```

```

class AdminFactory
{
    buildAdmin(String, String, String, String) Admin
}

```

```

class AdminFactoryTest
{
    AdminTest String
    testBuildAdmin() void
    setUp() void
    buildAdmin() void
}

```

```

class ProfitFactory
{
    buildProfit(String, String, String, String) Profit
}

```

```

class ProfitFactoryTest
{
    ProfitTest String
    testBuildProfit() void
    setUp() void
    buildProfit() void
}

```

```

class ChannelPartnerFactory
{
    buildChannelPartner(String, String, String, String, String) ChannelPartner
}

```

```

class ChannelPartnerFactoryTest
{
    ChannelPartnerTest String
    setUp() void
    buildCar() void
}

```

```

class AddressFactory
{
    buildAddress(String, String, String, String, String, String, String) Address
}

```

```

class AssetsFactory
{
    buildAssets(String, String, String, String, String) Assets
}

```