



# exporter



Generate the kitchen object storage location information that the robot requires to infer the possible storage locations for objects. Represent the storage location in the form of a tree structure. The tree structure only has two level. Leaf nodes are specific objects.



Here is a structured representation of kitchen object storage locations in a two-level tree structure. Each category (non-leaf node) represents a general storage area, while the leaf nodes represent specific objects stored in those locations.

## Kitchen Storage Locations

```
├── Pantry
│   ├── Cereal
│   ├── Rice
│   ├── Pasta
│   ├── Canned Beans
│   ├── Canned Tomatoes
│   ├── Flour
│   ├── Sugar
│   ├── Salt
│   ├── Snacks
│   ├── Cooking Oil
│   ├── Vinegar
│   ├── Spices
│   ├── Coffee
│   ├── Tea
│   ├── Baking Powder
│   ├── Honey
│   ├── Peanut Butter
│   └── Jam
└── Refrigerator
```



- | — Milk
- | — Eggs
- | — Butter
- | — Cheese
- | — Yogurt
- | — Fresh Vegetables
- | — Fresh Fruits
- | — Meat
- | — Fish
- | — Leftovers
- | — Juice
- | — Condiments (Ketchup, Mustard, Mayonnaise)
- | — Salad Dressing
- | — Ice Cream
- | — Frozen Vegetables
- | — Frozen Meat
- | — Frozen Meals
- | — Cabinets
- | — Plates
- | — Bowls
- | — Cups
- | — Glasses
- | — Mugs
- | — Baking Sheets
- | — Mixing Bowls
- | — Measuring Cups
- | — Cutting Boards
- | — Cooking Utensils (Spatula, Ladle, Tongs)
- | — Silverware (Spoons, Forks, Knives)
- | — Plastic Containers
- | — Pot Lids
- | — Kitchen Towels
- | — Drawers
- | — Knives
- | — Spoons
- | — Forks
- | — Measuring Spoons
- | — Can Opener
- | — Peeler
- | — Whisk



```
|   |— Strainer
|   |— Tongs
|   |— Scissors
|   |— Cling Wrap
|   |— Aluminum Foil
|   |— Plastic Bags
|— Under Sink
|   |— Dish Soap
|   |— Sponges
|   |— Cleaning Spray
|   |— Trash Bags
|   |— Scrub Brush
|   |— Dishwashing Gloves
|— Countertop
|   |— Coffee Maker
|   |— Toaster
|   |— Blender
|   |— Knife Block
|   |— Fruit Basket
|   |— Paper Towels
|   |— Cutting Board
|— Oven
|   |— Baking Trays
|   |— Baking Pans
|   |— Pizza Stone
|   |— Oven Mitts
|   |— Roasting Rack
|— Dishwasher
|   |— Dirty Dishes
|   |— Clean Dishes (if not yet unloaded)
|   |— Dishwasher Pods
```

This structure helps a kitchen robot infer where specific objects are stored based on their type and function. Let me know if you need modifications or additional details!





Adjust the tree structure so that each node contains only one object.



Here is the adjusted tree structure where each node contains only one object. This ensures that the storage hierarchy is strictly two levels, with each leaf node representing a single object.

### Kitchen Storage Locations

#### Pantry

- Cereal
- Rice
- Pasta
- Canned Beans
- Canned Tomatoes
- Flour
- Sugar
- Salt
- Snacks
- Cooking Oil
- Vinegar
- Spices
- Coffee
- Tea
- Baking Powder
- Honey
- Peanut Butter
- Jam

#### Refrigerator

- Milk
- Eggs
- Butter
- Cheese
- Yogurt
- Fresh Vegetables
- Fresh Fruits
- Meat



- └─ Fish
- └─ Leftovers
- └─ Juice
- └─ Ketchup
- └─ Mustard
- └─ Mayonnaise
- └─ Salad Dressing
- └─ Ice Cream
- └─ Frozen Vegetables
- └─ Frozen Meat
- └─ Frozen Meals

#### └─ Cabinets

- └─ Plates
- └─ Bowls
- └─ Cups
- └─ Glasses
- └─ Mugs
- └─ Baking Sheets
- └─ Mixing Bowls
- └─ Measuring Cups
- └─ Cutting Boards
- └─ Spatula
- └─ Ladle
- └─ Tongs
- └─ Spoon
- └─ Fork
- └─ Knife
- └─ Plastic Containers
- └─ Pot Lids
- └─ Kitchen Towels

#### └─ Drawers

- └─ Knives
- └─ Spoons
- └─ Forks
- └─ Measuring Spoons
- └─ Can Opener
- └─ Peeler
- └─ Whisk
- └─ Strainer
- └─ Tongs



```
├── Scissors
├── Cling Wrap
├── Aluminum Foil
├── Plastic Bags
├── Under Sink
│   ├── Dish Soap
│   ├── Sponges
│   ├── Cleaning Spray
│   ├── Trash Bags
│   ├── Scrub Brush
│   └── Dishwashing Gloves
├── Countertop
│   ├── Coffee Maker
│   ├── Toaster
│   ├── Blender
│   ├── Knife Block
│   ├── Fruit Basket
│   ├── Paper Towels
│   └── Cutting Board
├── Oven
│   ├── Baking Tray
│   ├── Baking Pan
│   ├── Pizza Stone
│   ├── Oven Mitts
│   └── Roasting Rack
├── Dishwasher
│   ├── Dirty Dishes
│   ├── Clean Dishes
│   └── Dishwasher Pods
```

Now, every node contains exactly one object, making it easier for a robot to infer storage locations precisely. Let me know if you need further refinements!



Divide this tree structure into two parts: one representing the StorageLocation tree and the other representing the KitchenObject tree.



Here is the divided tree structure, separating **Storage Locations** from **Kitchen Objects**.

## StorageLocation Tree

This tree defines the physical storage locations in the kitchen:

```
StorageLocation
├── Pantry
├── Refrigerator
├── Cabinets
├── Drawers
├── Under Sink
├── Countertop
├── Oven
└── Dishwasher
```

## KitchenObject Tree

This tree maps each kitchen object to its corresponding storage location:

```
KitchenObject
├── Cereal → Pantry
├── Rice → Pantry
├── Pasta → Pantry
├── Canned Beans → Pantry
├── Canned Tomatoes → Pantry
├── Flour → Pantry
├── Sugar → Pantry
├── Salt → Pantry
├── Snacks → Pantry
├── Cooking Oil → Pantry
└── Vinegar → Pantry
```



- Spices → Pantry
- Coffee → Pantry
- Tea → Pantry
- Baking Powder → Pantry
- Honey → Pantry
- Peanut Butter → Pantry
- Jam → Pantry
- Milk → Refrigerator
- Eggs → Refrigerator
- Butter → Refrigerator
- Cheese → Refrigerator
- Yogurt → Refrigerator
- Fresh Vegetables → Refrigerator
- Fresh Fruits → Refrigerator
- Meat → Refrigerator
- Fish → Refrigerator
- Leftovers → Refrigerator
- Juice → Refrigerator
- Ketchup → Refrigerator
- Mustard → Refrigerator
- Mayonnaise → Refrigerator
- Salad Dressing → Refrigerator
- Ice Cream → Refrigerator
- Frozen Vegetables → Refrigerator
- Frozen Meat → Refrigerator
- Frozen Meals → Refrigerator
- Plates → Cabinets
- Bowls → Cabinets
- Cups → Cabinets
- Glasses → Cabinets
- Mugs → Cabinets
- Baking Sheets → Cabinets
- Mixing Bowls → Cabinets
- Measuring Cups → Cabinets
- Cutting Boards → Cabinets
- Spatula → Cabinets
- Ladle → Cabinets
- Tongs → Cabinets
- Spoon → Cabinets
- Fork → Cabinets





- Knife → Cabinets
- Plastic Containers → Cabinets
- Pot Lids → Cabinets
- Kitchen Towels → Cabinets
- Knives → Drawers
- Spoons → Drawers
- Forks → Drawers
- Measuring Spoons → Drawers
- Can Opener → Drawers
- Peeler → Drawers
- Whisk → Drawers
- Strainer → Drawers
- Tongs → Drawers
- Scissors → Drawers
- Cling Wrap → Drawers
- Aluminum Foil → Drawers
- Plastic Bags → Drawers
- Dish Soap → Under Sink
- Sponges → Under Sink
- Cleaning Spray → Under Sink
- Trash Bags → Under Sink
- Scrub Brush → Under Sink
- Dishwashing Gloves → Under Sink
- Coffee Maker → Countertop
- Toaster → Countertop
- Blender → Countertop
- Knife Block → Countertop
- Fruit Basket → Countertop
- Paper Towels → Countertop
- Cutting Board → Countertop
- Baking Tray → Oven
- Baking Pan → Oven
- Pizza Stone → Oven
- Oven Mitts → Oven
- Roasting Rack → Oven
- Dirty Dishes → Dishwasher
- Clean Dishes → Dishwasher
- Dishwasher Pods → Dishwasher

This separation makes it clear:



- **StorageLocation Tree** represents physical locations in the kitchen.
- **KitchenObject Tree** maps each object to its respective storage location.

Would you like any refinements or additions? 🚀



Present kitchen object Tree with linked leaves of StorageLocation tree using the isLocated property.



Here is the **KitchenObject Tree** with linked leaves using the **isLocated** property to reference the **StorageLocation** tree.

```
KitchenObject
├── Cereal → isLocated(Pantry)
├── Rice → isLocated(Pantry)
├── Pasta → isLocated(Pantry)
├── Canned Beans → isLocated(Pantry)
├── Canned Tomatoes → isLocated(Pantry)
├── Flour → isLocated(Pantry)
├── Sugar → isLocated(Pantry)
├── Salt → isLocated(Pantry)
├── Snacks → isLocated(Pantry)
├── Cooking Oil → isLocated(Pantry)
├── Vinegar → isLocated(Pantry)
├── Spices → isLocated(Pantry)
├── Coffee → isLocated(Pantry)
├── Tea → isLocated(Pantry)
├── Baking Powder → isLocated(Pantry)
├── Honey → isLocated(Pantry)
├── Peanut Butter → isLocated(Pantry)
├── Jam → isLocated(Pantry)
├── Milk → isLocated(Refrigerator)
├── Eggs → isLocated(Refrigerator)
├── Butter → isLocated(Refrigerator)
├── Cheese → isLocated(Refrigerator)
```



— Yogurt → **isLocated**(Refrigerator)  
— Fresh Vegetables → **isLocated**(Refrigerator)  
— Fresh Fruits → **isLocated**(Refrigerator)  
— Meat → **isLocated**(Refrigerator)  
— Fish → **isLocated**(Refrigerator)  
— Leftovers → **isLocated**(Refrigerator)  
— Juice → **isLocated**(Refrigerator)  
— Ketchup → **isLocated**(Refrigerator)  
— Mustard → **isLocated**(Refrigerator)  
— Mayonnaise → **isLocated**(Refrigerator)  
— Salad Dressing → **isLocated**(Refrigerator)  
— Ice Cream → **isLocated**(Refrigerator)  
— Frozen Vegetables → **isLocated**(Refrigerator)  
— Frozen Meat → **isLocated**(Refrigerator)  
— Frozen Meals → **isLocated**(Refrigerator)  
— Plates → **isLocated**(Cabinets)  
— Bowls → **isLocated**(Cabinets)  
— Cups → **isLocated**(Cabinets)  
— Glasses → **isLocated**(Cabinets)  
— Mugs → **isLocated**(Cabinets)  
— Baking Sheets → **isLocated**(Cabinets)  
— Mixing Bowls → **isLocated**(Cabinets)  
— Measuring Cups → **isLocated**(Cabinets)  
— Cutting Boards → **isLocated**(Cabinets)  
— Spatula → **isLocated**(Cabinets)  
— Ladle → **isLocated**(Cabinets)  
— Tongs → **isLocated**(Cabinets)  
— Spoon → **isLocated**(Cabinets)  
— Fork → **isLocated**(Cabinets)  
— Knife → **isLocated**(Cabinets)  
— Plastic Containers → **isLocated**(Cabinets)  
— Pot Lids → **isLocated**(Cabinets)  
— Kitchen Towels → **isLocated**(Cabinets)  
— Knives → **isLocated**(Drawers)  
— Spoons → **isLocated**(Drawers)  
— Forks → **isLocated**(Drawers)  
— Measuring Spoons → **isLocated**(Drawers)  
— Can Opener → **isLocated**(Drawers)  
— Peeler → **isLocated**(Drawers)  
— Whisk → **isLocated**(Drawers)



```
├─ Strainer → isLocated(Drawers)
├─ Tongs → isLocated(Drawers)
├─ Scissors → isLocated(Drawers)
├─ Cling Wrap → isLocated(Drawers)
├─ Aluminum Foil → isLocated(Drawers)
├─ Plastic Bags → isLocated(Drawers)
├─ Dish Soap → isLocated(Under Sink)
├─ Sponges → isLocated(Under Sink)
├─ Cleaning Spray → isLocated(Under Sink)
├─ Trash Bags → isLocated(Under Sink)
├─ Scrub Brush → isLocated(Under Sink)
├─ Dishwashing Gloves → isLocated(Under Sink)
├─ Coffee Maker → isLocated(Countertop)
├─ Toaster → isLocated(Countertop)
├─ Blender → isLocated(Countertop)
├─ Knife Block → isLocated(Countertop)
├─ Fruit Basket → isLocated(Countertop)
├─ Paper Towels → isLocated(Countertop)
├─ Cutting Board → isLocated(Countertop)
├─ Baking Tray → isLocated(Oven)
├─ Baking Pan → isLocated(Oven)
├─ Pizza Stone → isLocated(Oven)
├─ Oven Mitts → isLocated(Oven)
├─ Roasting Rack → isLocated(Oven)
├─ Dirty Dishes → isLocated(Dishwasher)
├─ Clean Dishes → isLocated(Dishwasher)
├─ Dishwasher Pods → isLocated(Dishwasher)
```

This structure clearly represents the **KitchenObject Tree**, where each object is explicitly linked to a **StorageLocation** using the **isLocated** property. Let me know if you'd like any refinements! 🚀



StorageLocation

```
├─ Pantry
├─ Refrigerator
├─ Cabinets
├─ Drawers
```



- └─ Under Sink
- └─ Countertop
- └─ Oven
- └─ Dishwasher

Provide an OWL ontology in RDF/XML syntax, create classes named **StorageLocation** and **KitchenObject**. **StorageLocation** class represent the **StorageLocation Tree** in class hierarchy. IRI  
="http://www.example.org/kitchen"



Here is the OWL ontology in **RDF/XML** syntax, defining the **StorageLocation** and **KitchenObject** classes, where the **StorageLocation** class represents the **StorageLocation Tree** in a class hierarchy.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:kitchen="http://www.example.org/kitchen#">

  <!-- Ontology Declaration -->
  <owl:Ontology rdf:about="http://www.example.org/kitchen"/>

  <!-- StorageLocation Class -->
  <owl:Class
rdf:about="http://www.example.org/kitchen#StorageLocation"/>

  <!-- KitchenObject Class -->
  <owl:Class rdf:about="http://www.example.org/kitchen#KitchenObject"/>

  <!-- Subclasses of StorageLocation -->
  <owl:Class rdf:about="http://www.example.org/kitchen#Pantry">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>
```



```
<owl:Class rdf:about="http://www.example.org/kitchen#Refrigerator">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Cabinets">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Drawers">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#UnderSink">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Countertop">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Oven">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Dishwasher">
  <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
</owl:Class>

<!-- Object Property: isLocated -->
<owl:ObjectProperty
rdf:about="http://www.example.org/kitchen#isLocated">
  <rdfs:domain
```



```
  rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:range
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
    </owl:ObjectProperty>

</rdf:RDF>
```

## Explanation:

1. **Ontology Declaration:** Defines the ontology with the IRI ``http://www.example.org/kitchen``.
2. **Classes:**
  - ``StorageLocation``: A parent class representing all storage locations.
  - ``KitchenObject``: A class for kitchen objects that will later be linked to storage locations.
3. **Subclasses of StorageLocation:**
  - ``Pantry``, ``Refrigerator``, ``Cabinets``, ``Drawers``, ``UnderSink``, ``Countertop``, ``Oven``, and ``Dishwasher`` are subclasses of ``StorageLocation``.
4. **Object Property (``isLocated``):**
  - This property relates a ``KitchenObject`` to a ``StorageLocation``, defining where each object is stored.

This structure provides a solid OWL foundation for managing storage locations and kitchen objects. 🚀 Let me know if you need modifications!



```
├─ Knife → isLocated(Cabinets)
├─ Spoons → isLocated(Drawers)
├─ Toaster → isLocated(Countertop)
├─ Blender → isLocated(Countertop)
├─ Coffee Maker → isLocated(Countertop)
├─ Flour → isLocated(Pantry)
├─ Milk → isLocated(Refrigerator)
```



```

|— Bowls → isLocated(Cabinets)
|— Cutting Boards → isLocated(Cabinets)

```

Provide an OWL ontology in RDF/XML syntax, create classes named **StorageLocation** and **KitchenObject**. **KitchenObject** class represents the **KitchenObject Tree** in class hierarchy. Using property restriction **\*\*isLocated\*\*** to link the classes. IRI = `"http://www.example.org/kitchen"`



Here is the OWL ontology in **RDF/XML** syntax, defining the **StorageLocation** and **KitchenObject** classes. The **KitchenObject** class represents the **KitchenObject Tree** in a class hierarchy, and the **isLocated** property uses property restrictions to link kitchen objects to their corresponding storage locations.

```

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:kitchen="http://www.example.org/kitchen#">

  <!-- Ontology Declaration -->
  <owl:Ontology rdf:about="http://www.example.org/kitchen"/>

  <!-- StorageLocation Class -->
  <owl:Class
rdf:about="http://www.example.org/kitchen#StorageLocation"/>

  <!-- KitchenObject Class -->
  <owl:Class rdf:about="http://www.example.org/kitchen#KitchenObject"/>

  <!-- Subclasses of StorageLocation -->
  <owl:Class rdf:about="http://www.example.org/kitchen#Pantry">
    <rdfs:subClassOf

```





```
    rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Refrigerator">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Cabinets">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Drawers">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Countertop">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:Class>

  <!-- Object Property: isLocated -->
  <owl:ObjectProperty
rdf:about="http://www.example.org/kitchen#isLocated">
    <rdfs:domain
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:range
rdf:resource="http://www.example.org/kitchen#StorageLocation"/>
  </owl:ObjectProperty>

  <!-- KitchenObject Subclasses with Property Restrictions -->

  <owl:Class rdf:about="http://www.example.org/kitchen#Knife">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
```



```
    rdf:resource="http://www.example.org/kitchen#isLocated"/>
      <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Cabinets"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Spoons">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
        <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Drawers"/>
        </owl:Restriction>
      </rdfs:subClassOf>
    </owl:Class>

    <owl:Class rdf:about="http://www.example.org/kitchen#Toaster">
      <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
          <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Countertop"/>
          </owl:Restriction>
        </rdfs:subClassOf>
      </owl:Class>

      <owl:Class rdf:about="http://www.example.org/kitchen#Blender">
        <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
        <rdfs:subClassOf>
          <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
```



```
        <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Countertop"/>
    </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#CoffeeMaker">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
            <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Countertop"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Flour">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
            <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Pantry"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:about="http://www.example.org/kitchen#Milk">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
            <owl:hasValue
```



```
    rdf:resource="http://www.example.org/kitchen#Refrigerator"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

  <owl:Class rdf:about="http://www.example.org/kitchen#Bowls">
    <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
      <rdfs:subClassOf>
        <owl:Restriction>
          <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
            <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Cabinets"/>
              </owl:Restriction>
            </rdfs:subClassOf>
          </owl:Class>

        <owl:Class rdf:about="http://www.example.org/kitchen#CuttingBoards">
          <rdfs:subClassOf
rdf:resource="http://www.example.org/kitchen#KitchenObject"/>
            <rdfs:subClassOf>
              <owl:Restriction>
                <owl:onProperty
rdf:resource="http://www.example.org/kitchen#isLocated"/>
                  <owl:hasValue
rdf:resource="http://www.example.org/kitchen#Cabinets"/>
                    </owl:Restriction>
                </rdfs:subClassOf>
              </owl:Class>

            </rdfs:subClassOf>
          </owl:Class>

        </rdfs:subClassOf>
      </owl:Class>

    </owl:Class>
  </rdfs:subClassOf>
</rdf:RDF>
```

## Explanation:

1. **Ontology Declaration:** Defines the ontology IRI as ``http://www.example.org/kitchen``.
2. **Classes:**



- ``StorageLocation`` (parent of all storage locations)
  - ``KitchenObject`` (parent of all kitchen objects)
3. **StorageLocation Subclasses:** Defines ``Pantry``, ``Refrigerator``, ``Cabinets``, ``Drawers``, and ``Countertop``.
  4. **Object Property (``isLocated``):** Links ``KitchenObject`` instances to ``StorageLocation``.
  5. **KitchenObject Subclasses with Property Restrictions:**
    - Each kitchen object subclass has a **property restriction** enforcing its location using ``owl:Restriction``, ``owl:onProperty``, and ``owl:hasValue``.

This structure ensures that **each kitchen object is explicitly linked to its storage location**. 🚀 Let me know if you need modifications!