

# Taskly TMS

Presented by Group 2

# Project Overview

Developing a robust, command-line Task Management System (TMS) that streamlines task management for users.

## Key Functionalities:

1. Task Definition
2. Task Querying
3. Task Modification



# Task Types

## Primitive Task



Primitive tasks are standalone work units with a unique name, description, duration (in hours), and potential prerequisites.

Example:  
task1 boil-water 0.3 ,  
task2 boil-egg 0.5 task1

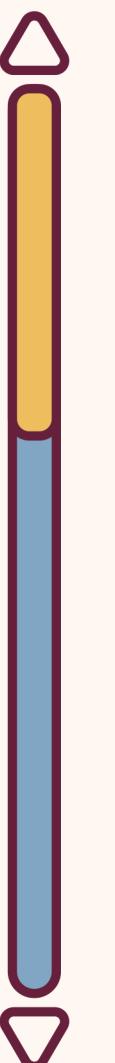


## Composite Task



Composite tasks are collections of other tasks (simple or composite), also featuring a unique name, description, and subtasks.

Example:  
comtask1 cook-egg task1,task2

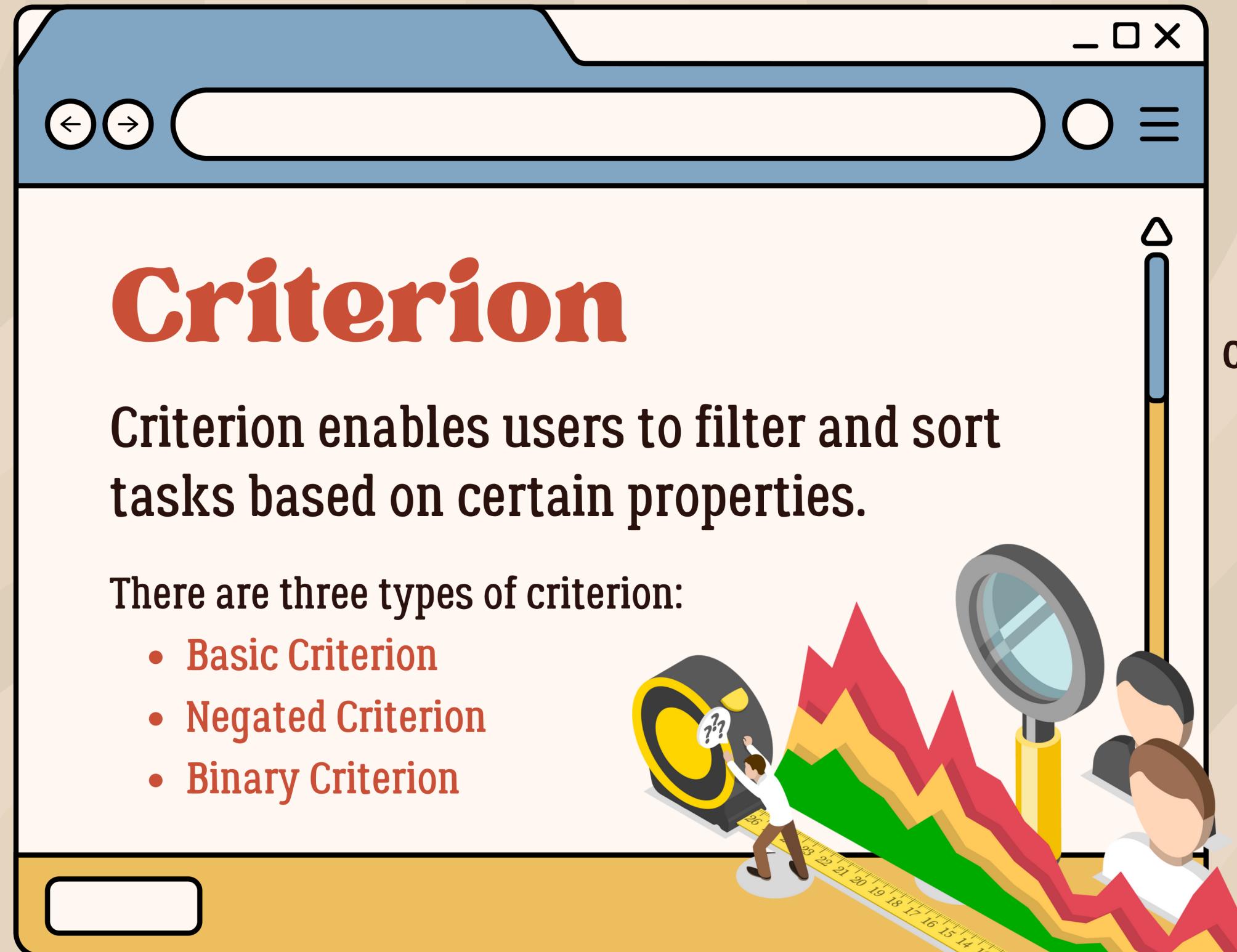


# Criterion

Criterion enables users to filter and sort tasks based on certain properties.

There are three types of criterion:

- Basic Criterion
- Negated Criterion
- Binary Criterion

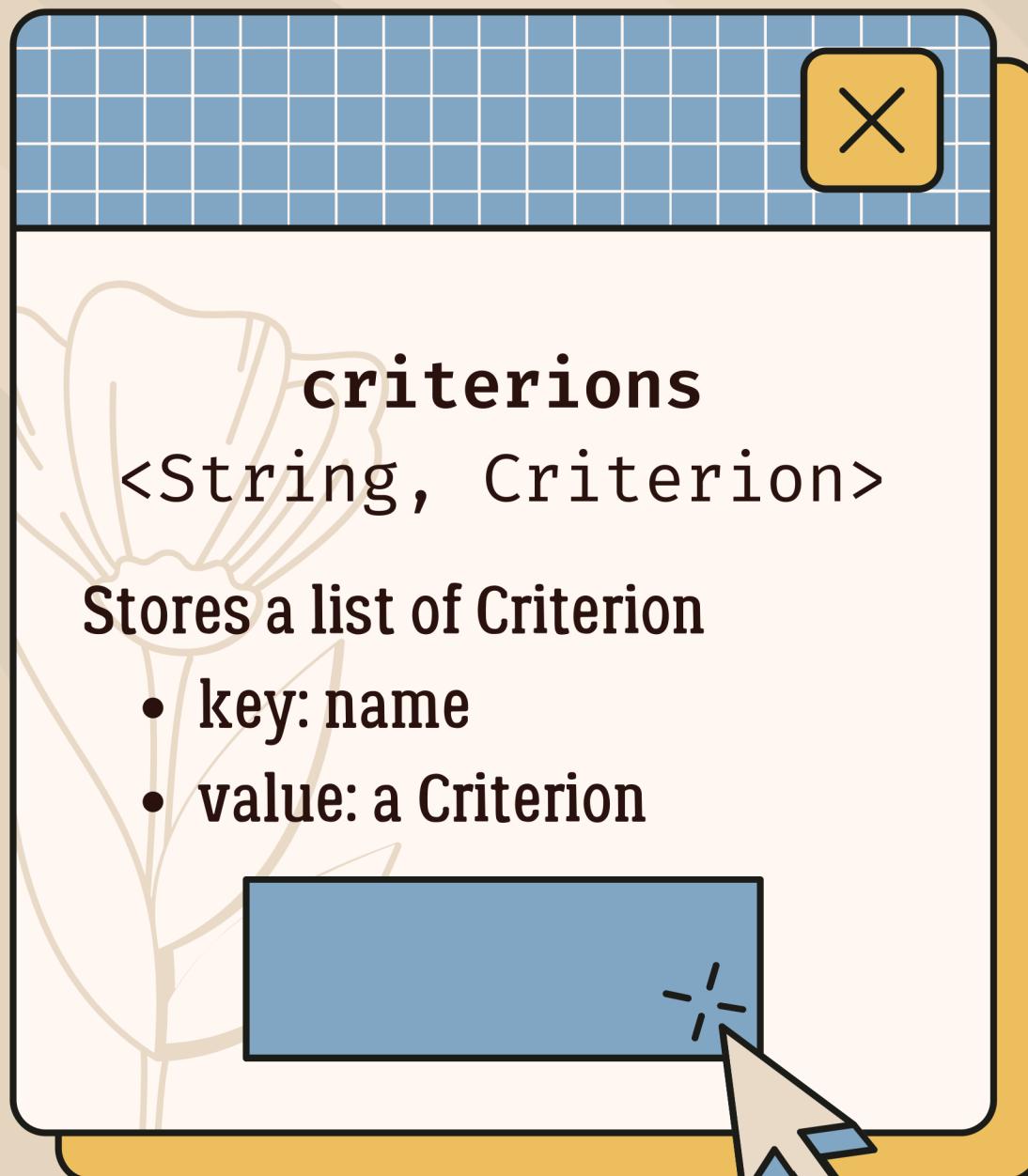
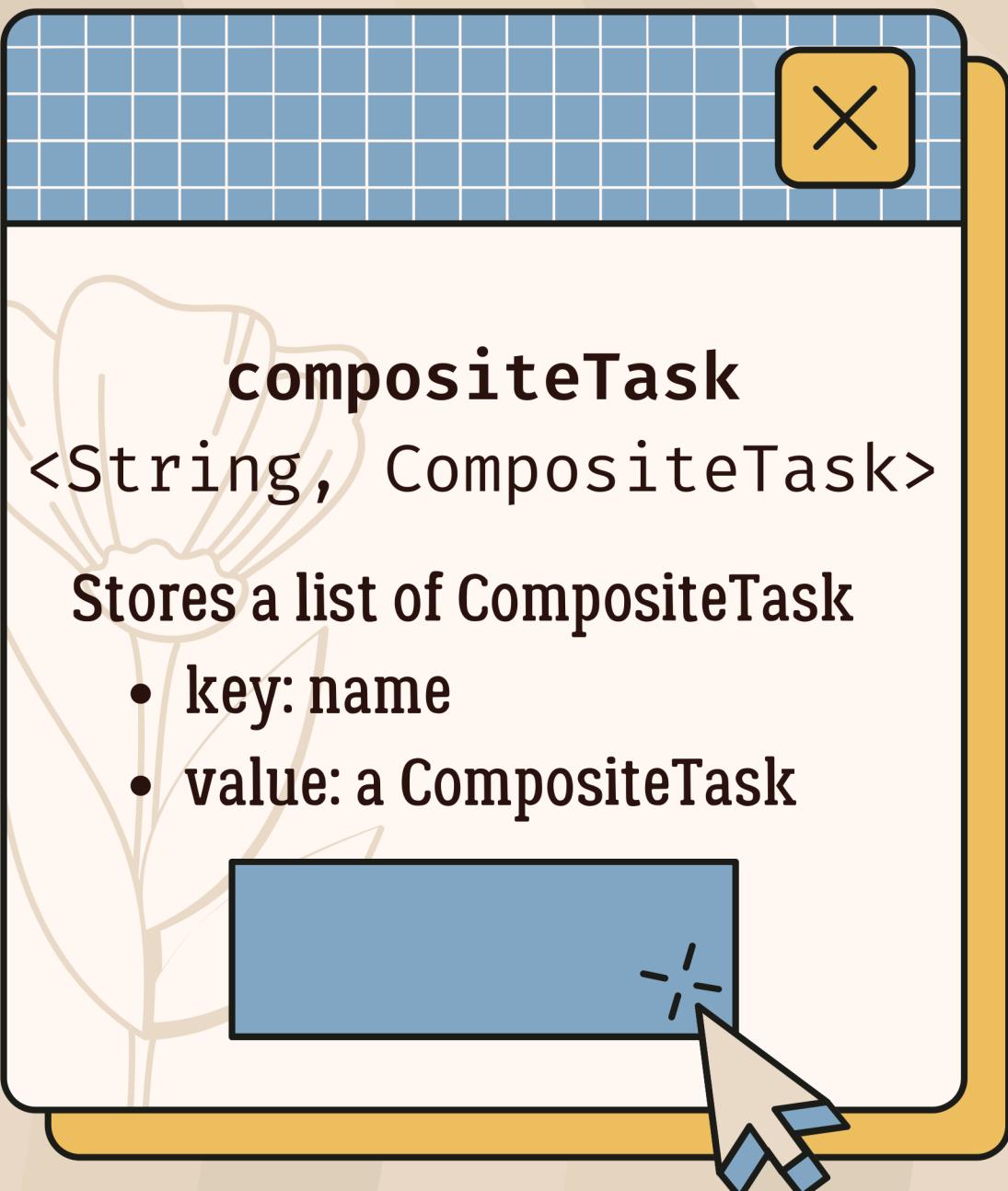
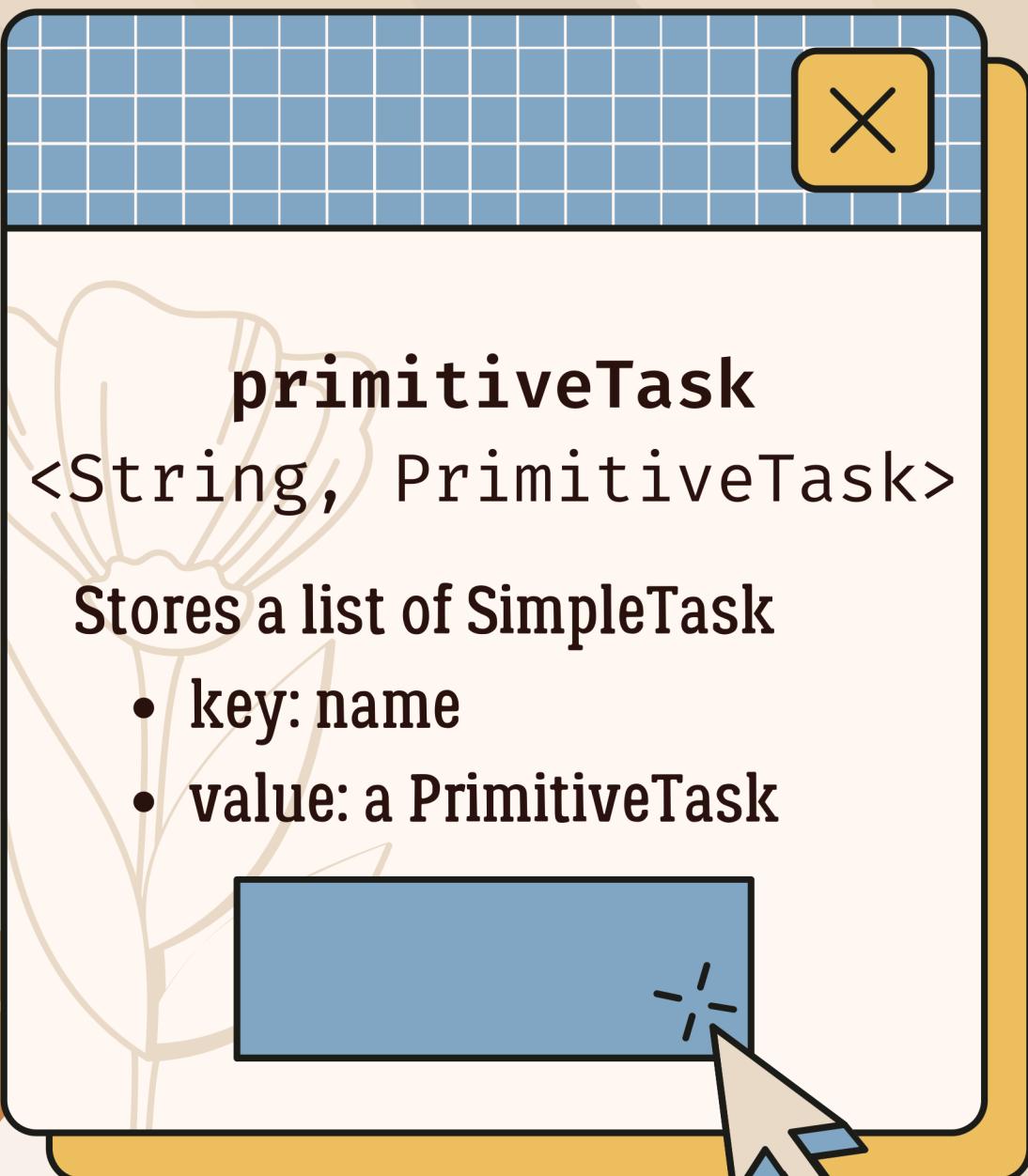


A pie chart on the right side of the slide illustrates the distribution of tasks. The chart is divided into three segments: a large pink segment labeled 'comtask1' at 40%, a blue segment labeled 'task1' at 20%, and a purple segment labeled 'task2' at 20%. A third segment, labeled 'task3', is also present but its percentage is not explicitly labeled on the chart.

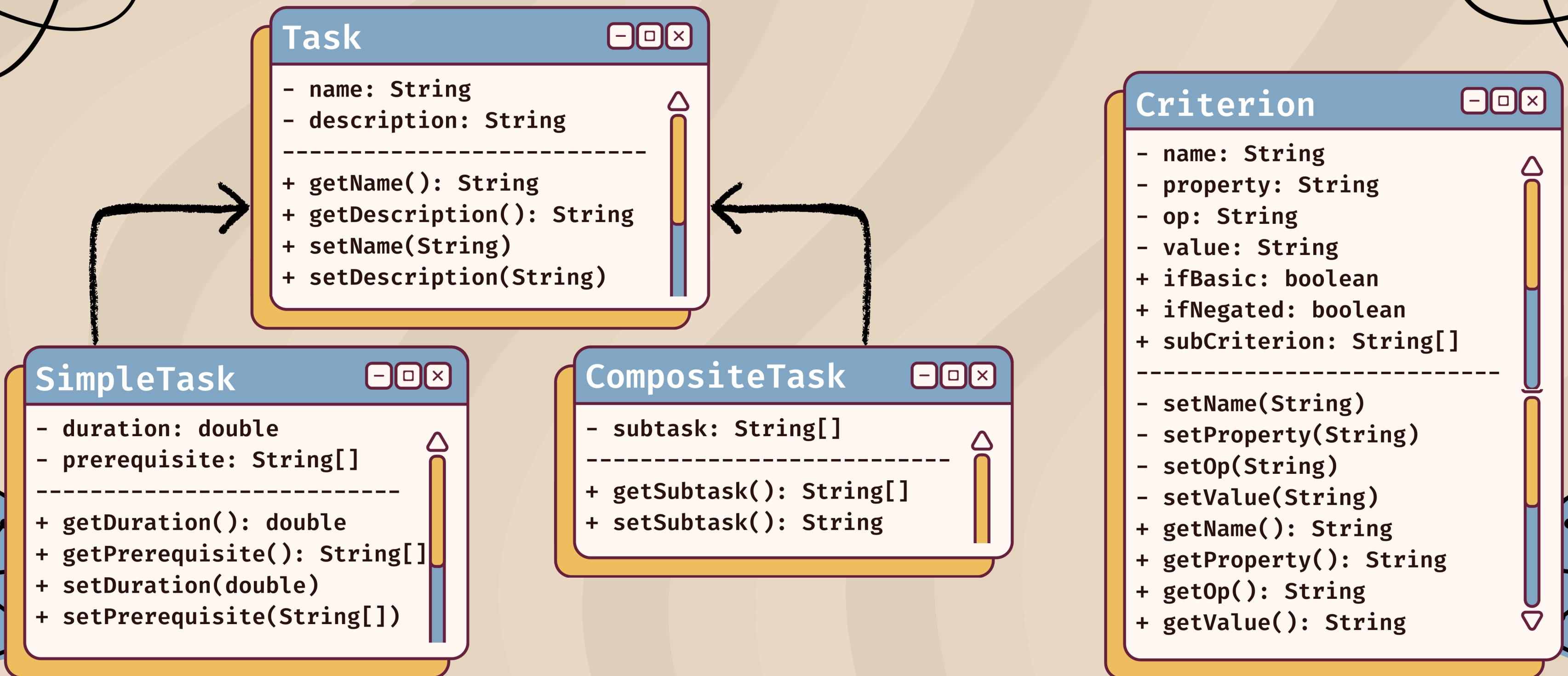
Task Category	Percentage
comtask1	40%
task1	20%
task2	20%
task3	20%

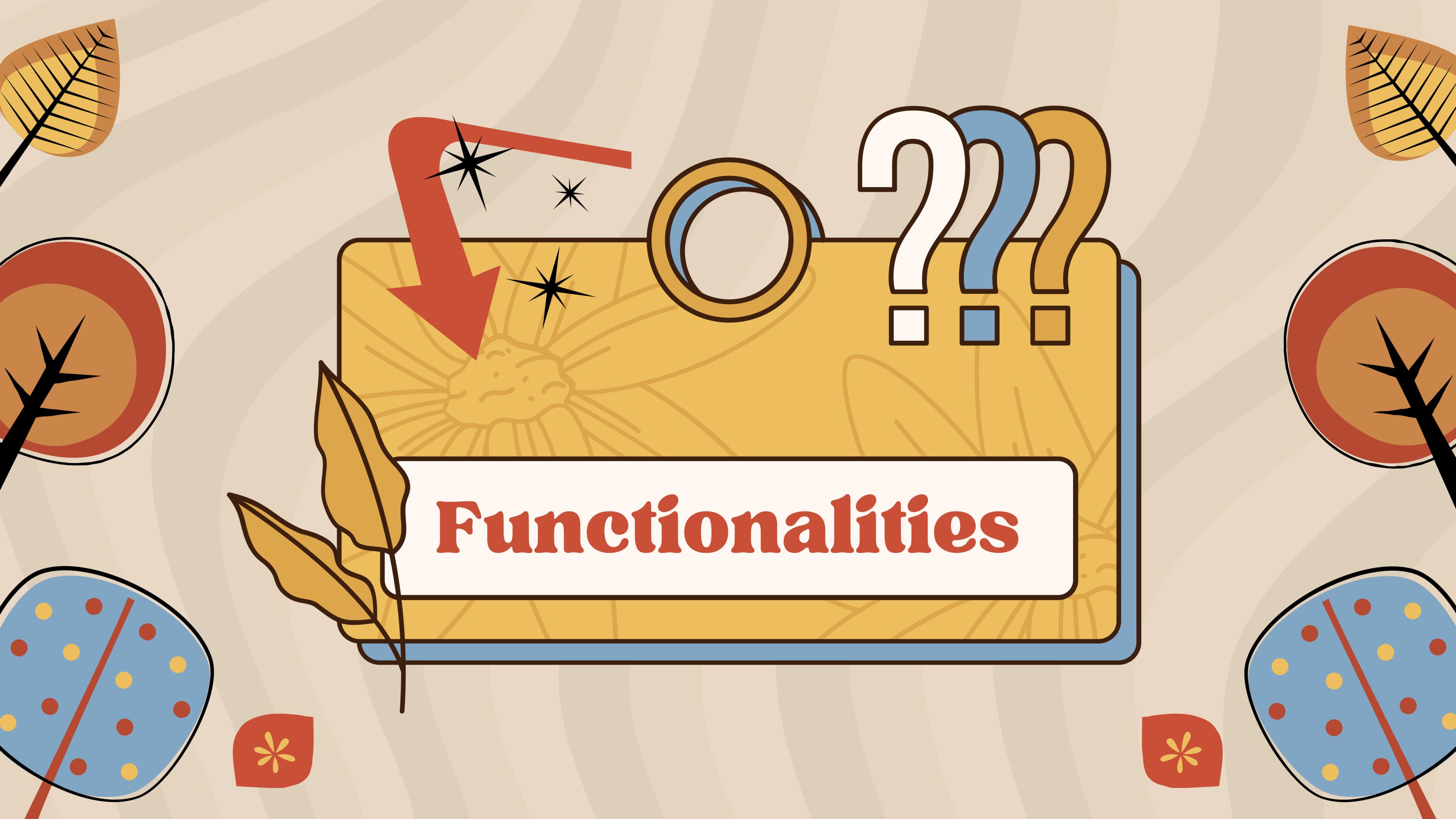


# Architecture: HashMap



# Design Choice





# Functionalities



## CreatePrimitiveTask



The system supports creating primitive tasks.

**Command:** CreatePrimitiveTask name description duration prerequisites

**Effect:** Creates a new simple task

**Example:** CreatePrimitiveTask task1 boil-water 0.3 ,

```
TaskManagementSystem@COMP2021 ~ % CreatePrimitiveTask task1 boil-water 0.3 ,
```

```
task1 has been successfully created!
```

```
TaskManagementSystem@COMP2021 ~ % CreatePrimitiveTask task2 boil-egg 0.7 task1
```

```
task2 has been successfully created!
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# CreateCompositeTask

The system supports creating primitive tasks.

**Command:** CreateCompositeTask name0 description name1,name2,...,namek

**Effect:** Creates a new simple task

**Example:** CreateCompositeTask comtask1 cook-egg task1,task2

```
TaskManagementSystem@COMP2021 ~ % CreateCompositeTask comtask1 cook-egg task1,task2
```

```
comtask1 has been successfully created!
```

```
TaskManagementSystem@COMP2021 ~ %
```



## DeleteTask



The system supports deleting tasks.

**Command:** DeleteTask name

**Effect:** Deletes an existing task

**Example:** DeleteTask comtask1

```
TaskManagementSystem@COMP2021 ~ % DeleteTask task2
```

```
task2 has been successfully deleted.
```

```
TaskManagementSystem@COMP2021 ~ %
```

# ChangeTask

The system supports changing the properties of an existing task.

**Command:** `ChangeTask name property newValue`

**Effect:** Sets the property of a specific task to the new value

**Example:** `ChangeTask task1 duration 0.5`

```
TaskManagementSystem@COMP2021 ~ % ChangeTask task1 duration 0.5
task1's duration has been successfully changed!
TaskManagementSystem@COMP2021 ~ % |
```



## PrintTask



The system supports printing the information of a task.

**Command:** PrintTask name

**Effect:** Prints the information of a task

**Example:** PrintTask task1

```
TaskManagementSystem@COMP2021 ~ % PrintTask task1
```

```
task1 -> description: boil-water, duration: 0.5, prerequisite(s): null
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# PrintAllTasks

The system supports printing the information of all tasks.

**Command:** PrintAllTasks

**Effect:** Prints the information of all existing tasks

**Example:** PrintAllTasks

```
TaskManagementSystem@COMP2021 ~ % PrintAllTasks
Here is the list of Simple Task(s)
task1 -> description: boil-water, duration: 0.5, prerequisite(s): null
```

```
Here is the list of Composite Task(s)
comtask1 -> description: cook-egg, subtasks: task1
TaskManagementSystem@COMP2021 ~ % |
```



## ReportDuration



The system supports reporting the duration of a task.

**Command:** ReportDuration name

**Effect:** Reports the duration of a task

**Example:** ReportDuration task1

```
TaskManagementSystem@COMP2021 ~ % ReportDuration task1
```

```
0.5
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# ReportEarliestFinishTime



The system supports reporting the earliest finish time of a task.

**Command:** ReportEarliestFinishTime name

**Effect:** Reports the earliest finish time of a task

**Example:** ReportEarliestFinishTime comtask1

```
TaskManagementSystem@COMP2021 ~ % ReportEarliestFinishTime comtask1
```

```
0.5
```

```
TaskManagementSystem@COMP2021 ~ %
```



## DefineBasicCriterion



The system supports defining basic task selection criteria.

**Command:** DefineBasicCriterion name1 property op value

**Effect:** Defines a basic task selection criterion

**Example:** DefineBasicCriterion cri1 description contains boil

```
TaskManagementSystem@COMP2021 ~ % DefineBasicCriterion cri1 description contains boil
```

```
cri1 has been successfully defined.
```

```
TaskManagementSystem@COMP2021 ~ %
```

# DefineNegatedCriterion



The system supports defining negated task selection criteria.

**Command:** DefineNegatedCriterion name1 name2

**Effect:** Defines the negation of an existing criterion

**Example:** DefineNegatedCriterion cri2 cri1

```
TaskManagementSystem@COMP2021 ~ % DefineNegatedCriterion cri2 cri1
```

```
cri2 has been successfully defined.
```

```
TaskManagementSystem@COMP2021 ~ %
```



## DefineBinaryCriterion



The system supports defining logical test selection criteria.

**Command:** DefineBinaryCriterion name1 name2 logicOp name3

**Effect:** Defines the logical combination (&& or ||) of two existing criteria

**Example:** DefineBinaryCriterion cri3 cri1 || cri2

```
TaskManagementSystem@COMP2021 ~ % DefineBinaryCriterion cri3 cri1 || cri2
```

```
cri3 has been successfully defined.
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# PrintAllCriteria



The system supports printing all defined criteria.

**Command:** PrintAllCriteria

**Effect:** Prints out all the criteria defined.

**Example:** PrintAllCriteria

```
TaskManagementSystem@COMP2021 ~ % PrintAllCriteria
Here is the list of Criteria(s)
IsPrimitive
cri2 -> property: description, value: boil, op: not-contains
cri3 -> property: description || description, value: boil || boil, op: contains || not-contains
cri1 -> property: description, value: boil, op: contains
TaskManagementSystem@COMP2021 ~ % |
```



## Search



The system supports searching for tasks based on an existing criterion.

**Command:** `Search name`

**Effect:** Lists all tasks that satisfy the criterion with the given name

**Example:** `Search cri3`

```
TaskManagementSystem@COMP2021 ~ % Search cri3
```

```
Here is the list of task(s) of cri3
```

```
task1
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# Store



The system supports storing the defined tasks and criteria into a file.

**Command:** `Store path`

**Effect:** Stores the defined tasks and criteria into the file

**Example:** `Store d:\task.data`

```
TaskManagementSystem@COMP2021 ~ % Store d:\tasks.data
```

```
Tasks and criteria stored successfully.
```

```
TaskManagementSystem@COMP2021 ~ % |
```



# Load



The system supports loading tasks and criteria from the file.

**Command:** `Load path`

**Effect:** Lists all tasks that satisfy the criterion with the given name

**Example:** `Load d:\tasks.data`

```
TaskManagementSystem@COMP2021 ~ % Load d:\tasks.data
```

```
Tasks and criteria read successfully.
```

```
TaskManagementSystem@COMP2021 ~ % |
```

# Quit

The system supports termination of the current execution.

Command: `Quit`

Effect: Terminates the execution of the system

Example: `Quit`

```
TaskManagementSystem@COMP2021 ~ % Quit
Thank you for using the TMS. See you again!
Process finished with exit code 0
```



**Thank  
You**

