



LEVEL UP

© 2022

ClassName

attribute: type

+publicMethod(inputVar) : returnType

-privateMethod

abstractMethod

<<constructor> ClassName(inputs)





Name





Attributes





Methods

classA

```
graph TD; classA[classA] -- Inheritance --> classB[classB];
```

The diagram illustrates a class inheritance relationship. At the top is a box for 'classA' with two empty slots below it. A vertical line connects 'classA' to a central box labeled 'Inheritance'. From the bottom of the 'Inheritance' box, a downward-pointing arrow leads to a box for 'classB', which also has two empty slots below it.

Inheritance

classB

classG

```
classDiagram
    classG --> classH : Association
```

The diagram illustrates a directed association between two classes. At the top is a class box for 'classG', which has a header section containing the name and two empty compartments below it. A vertical line descends from the bottom of 'classG' to a central label 'Association'. From this label, another vertical line descends to a class box for 'classH' at the bottom. This line terminates in a hollow triangular arrowhead pointing towards 'classH'. The 'classH' box also has a header section with the name and two empty compartments below it.

Association

classH

classC

```
classDiagram
    classC "1" *-- "1" classD
```

The diagram illustrates a composition relationship between two classes, classC and classD. classC is positioned at the top, and classD is at the bottom. A vertical line connects the bottom of classC to the top of classD, with a hollow diamond symbol at the junction, indicating a composition relationship. Both class boxes are divided into three horizontal compartments: the top for the class name, and the bottom two for attributes and methods respectively. The relationship is labeled 'Composition' in the center of the diagram.

Composition

classD

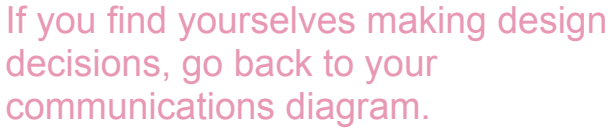
Look at your Communication Diagram, create a Design Class Diagram of all the classes (with methods and attributes) needed for this iteration. Shared understanding is more important than notation

You can draw it, or use Mermaid.

<https://mermaid.live>

A red octagonal stop sign with the word "STOP" in white capital letters in the center.

STOP



If you find yourselves making design decisions, go back to your communications diagram.

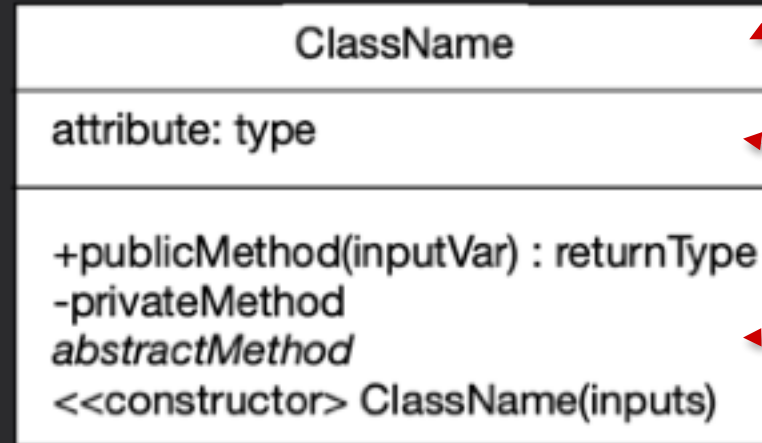
20:00

Design Class Diagrams

Look at your Communication Diagram, create a Design Class Diagram of all the classes (with methods and attributes) needed for this iteration. Shared understanding is more important than notation

You can draw it, or use Mermaid.

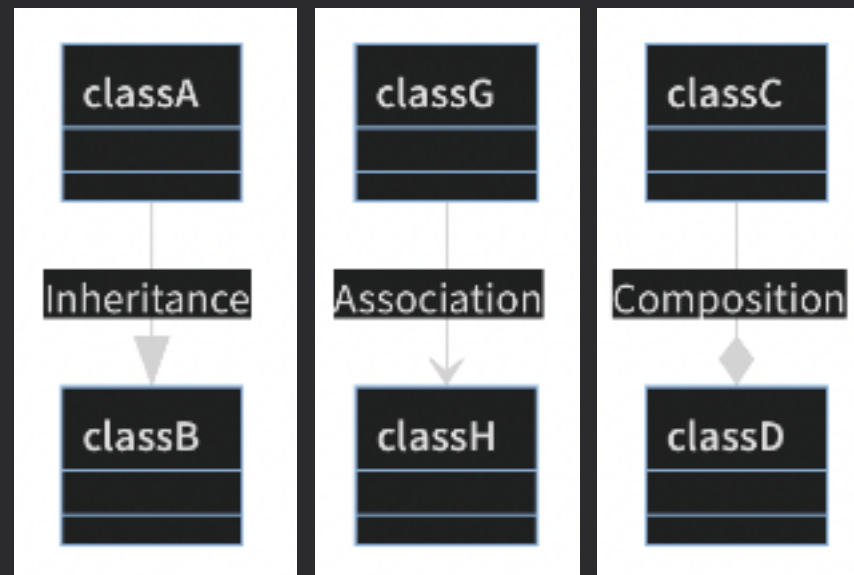
<https://mermaid.live>



Name

Attributes

Methods



20:00

STOP

If you find yourselves making design decisions, go back to your communications diagram.

Continuous Everything