



FIRST PAGE LEFT LEFT BLANK

This is my first Mermaid test

BLANK

Mermaid Class Diagram

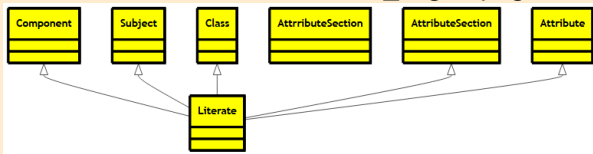


```
classDiagram
class Component
class Literate
class Subject
class Class
class Attribute Section
class Attribute

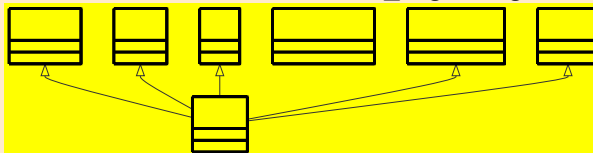
Component <|-- Literate
Subject <|-- Literate
Class <|-- Literate
AttributeSection <|-- Literate
Attribute <|-- Literate

classDef default fill:yellow,stroke:#000, color:black, stroke-
width:4px
```

Mermaid PNG - mermaid_img14.png



Mermaid SVG - mermaid_img14.svg



Mermaid Flowchart



```

%%{init: {
"flowchart": {
"curve": "stepAfter",
"useMaxWidth": true
}
}}%%

flowchart TB
subgraph Component["Component - Base class"]
direction TB

Literate["Literate<br>Core implementation"]

subgraph Subtypes["Component Subtypes"]
direction LR
Subject["Subject<br>Domain entity"]
Class["Class<br>Schema definition"]
AttributeSection["AttributeSection<br>Property group"]
Attribute["Attribute<br>Individual property"]
end

Subject ==> Literate
Class ==> Literate
AttributeSection ==> Literate
Attribute ==> Literate
end

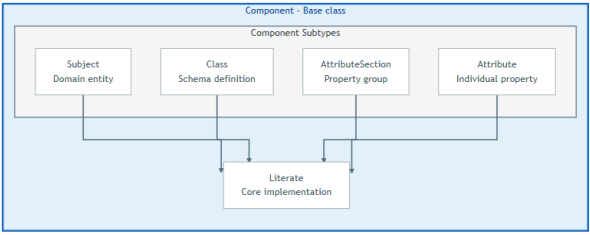
%% Styling with border-radius only
classDef container fill:#e3f2fd,stroke:#1565c0,stroke-
width:3px,color:#0d47a1,border-radius:10px
classDef subcontainer fill:#f5f5f5,stroke:#78909c,stroke-
width:2px,color:#37474f,border-radius:8px
classDef default fill:white,stroke:#90a4ae,stroke-
width:1px,color:#455a64,border-radius:5px

class Component container
class Subtypes subcontainer

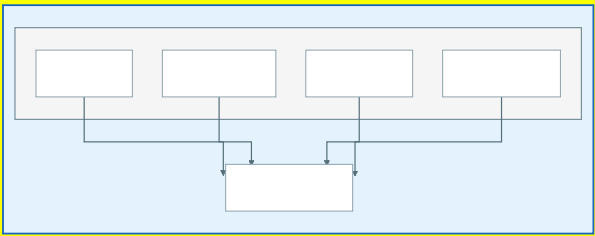
%% Edge styling
linkStyle default stroke:#546e7a,stroke-width:2px, border-radius:
20px

```


Mermaid PNG - mermaid_img15.png

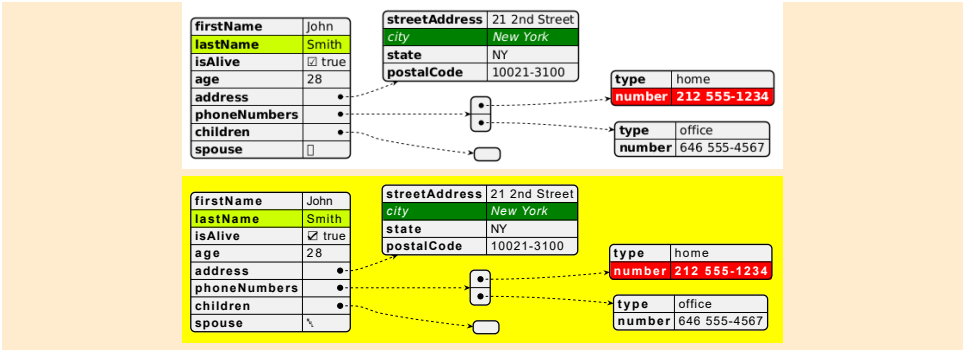


Mermaid SVG - mermaid_img15.svg



BLANK

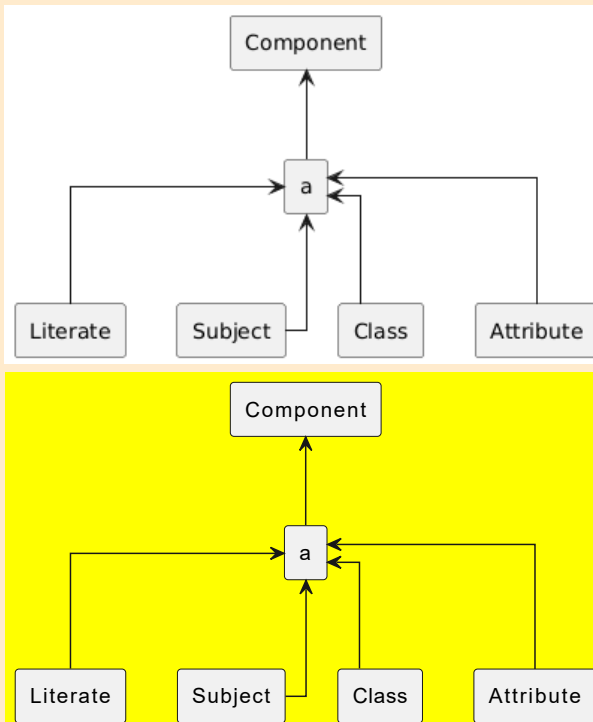

```
@startjson
<style>
.h1 {
  BackGroundColor green
  FontColor white
  FontStyle italic
}
.h2 {
  BackGroundColor red
  FontColor white
  FontStyle bold
}
</style>
#highlight "lastName"
#highlight "address" / "city" <<h1>>
#highlight "phoneNumbers" / "0" / "number" <<h2>>
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 28,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
@endjson
```



BLANK


```
@startuml
    rectangle Component
    rectangle Literate
    rectangle Subject
    rectangle Class
    rectangle Attribute
    rectangle a

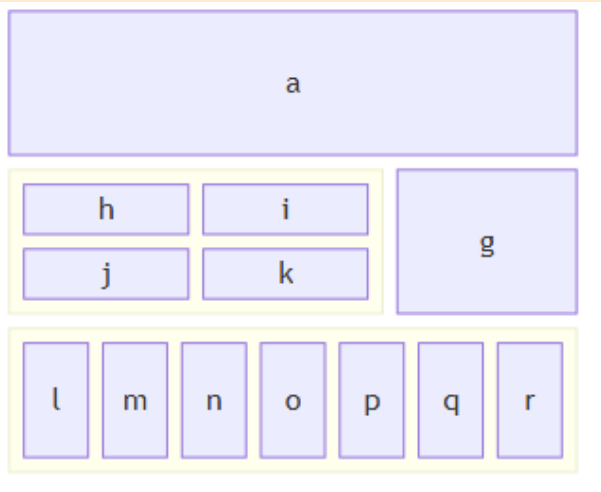
    Literate -u-> a
    Subject -u-> a
    Class -u-> a
    Attribute -u-> a
    a -u-> Component
    skinparam linetype ortho
@enduml
```



Mermaid block diagram

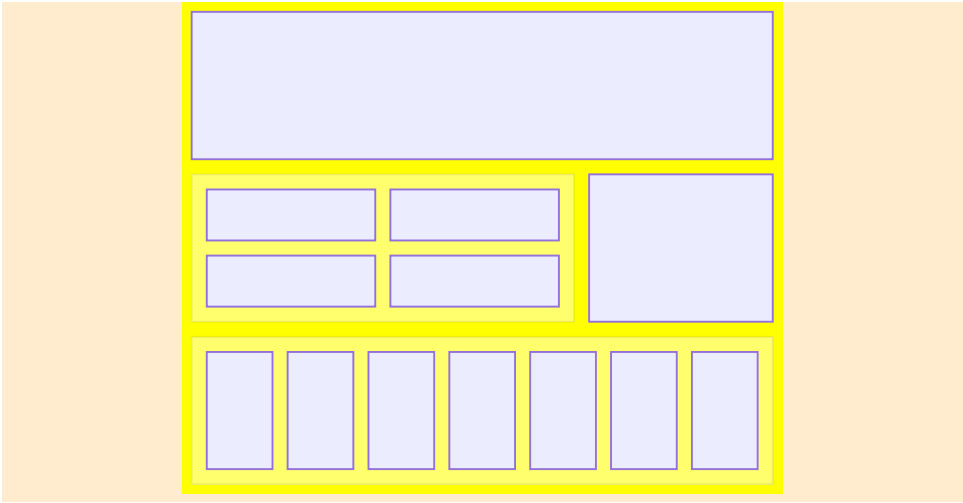
```
block-beta
columns 3
a:3
block:group1:2
columns 2
h i j k
end
g
block:group2:3
%% columns auto (default)
l m n o p q r
end
```

Mermaid PNG - mermaid_img18.png



Mermaid SVG - mermaid_img18.svg

Mermaid block diagram



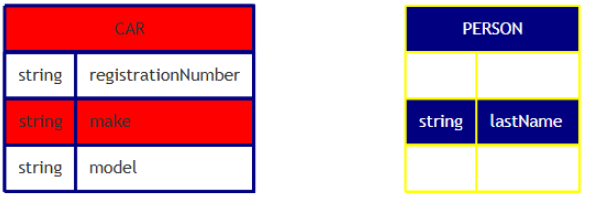
BLANK

Mermaid ER Diagram



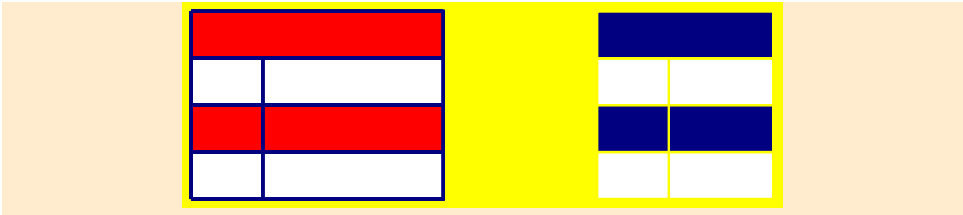
```
erDiagram
    CAR {
        string registrationNumber
        string make
        string model
    }
    PERSON {
        string firstName
        string lastName
        int age
    }
    style CAR fill: red,stroke:navy,stroke-width:3px
    style PERSON color: white, fill: navy,stroke:yellow ,stroke-width:2px
```

Mermaid PNG - mermaid_img19.png



Mermaid SVG - mermaid_img19.svg

Mermaid ER Diagram



BLANK

Mermaid ER Diagram



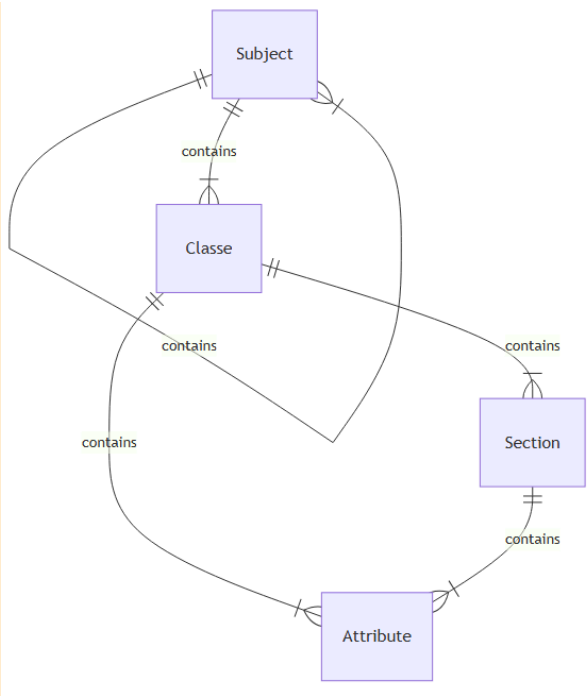
```
erDiagram
    class Subject Component
    class Section Component
    class Attribute Component
    class Classe Component

    Subject ||--|{ Subject : contains
    Subject ||--|{ Classe : contains

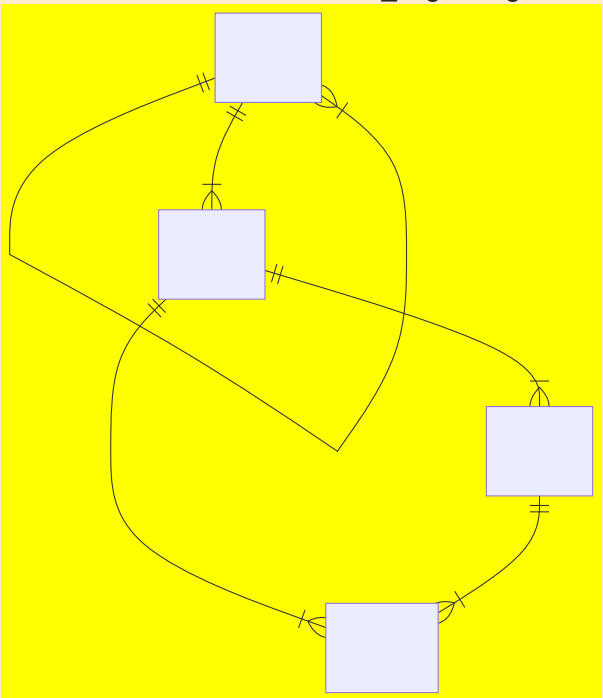
    Classe ||--|{ Section : contains
    Classe ||--|{ Attribute : contains
    Section ||--|{ Attribute : contains
```

Mermaid PNG - mermaid_img20.png

Mermaid ER Diagram



Mermaid SVG - mermaid_img20.svg



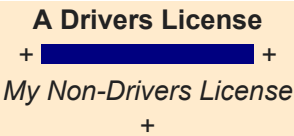
BLANK

Captioned figure





And the same figure with figure/caption markup



List of Codes

eFormat, Description	
E-Book, 'Kindle or Apple books - etc'	
PDF, formatted for printing and direct delivery	
eFormat	Description
0	E-Book 'Kindle or Apple books - etc'
1	PDF formatted for printing and direct delivery


```
@startuml

nwdiag {

network {

Component;

Literate;

Subject;

Attribute;

AttributeSection;

Class;

Component -- Literate;

Component -- Subject;

Component -- Class;

Component -- AttributeSection;

Component -- Attribute;

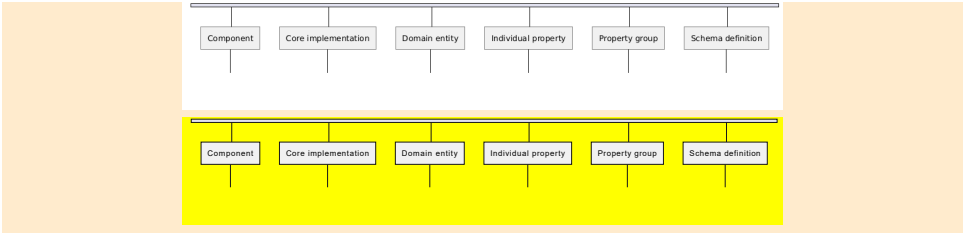

Subject [description = "Domain entity"];

Literate [description = "Core implementation"];

AttributeSection [description = "Property group"];

Attribute [description = "Individual property"];

Class [description = "Schema definition"];
```



BLANK


```

@startuml
hide empty description
!pragma layout elk
skinparam rectangleBorderThickness 1
skinparam defaultTextAlignment center
skinparam lifelineStrategy solid
skinparam monochrome false
skinparam style strictuml
hide empty members
skinparam Linetype ortho

rectangle "Базовые модули" as base {

class "Базовые объекты" as baseobjects
class "Делопроизводство\n4.5" as takeoffice
class "Управление\nпроцессами" as workflow
class "Windows-клиент" as windowsclient

class "Управление\ndocumentами" as documentmanagement
class "Конструктор\nсогласований" as approvaldesigner

class "Платформа" as platform
class "Служба\n фоновых операций" as worker

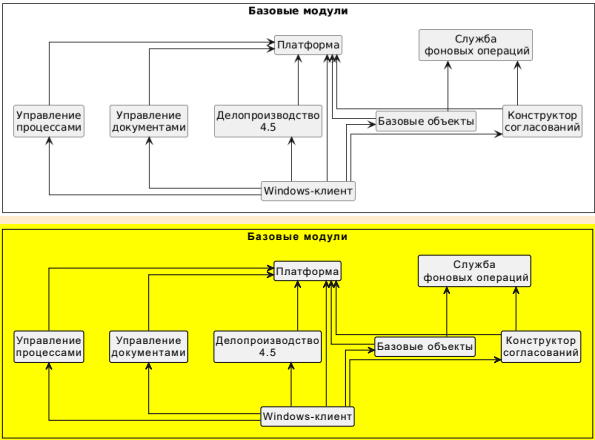
}

platform <-- baseobjects
platform <-- workflow
platform <-- takeoffice
platform <-- windowsclient
platform <-- documentmanagement
platform <-- approvaldesigner

windowsclient -up-> approvaldesigner
windowsclient -up-> documentmanagement
windowsclient -up-> baseobjects
windowsclient -up-> takeoffice
windowsclient -up-> workflow

worker <-- approvaldesigner
worker <-- baseobjects
@enduml

```



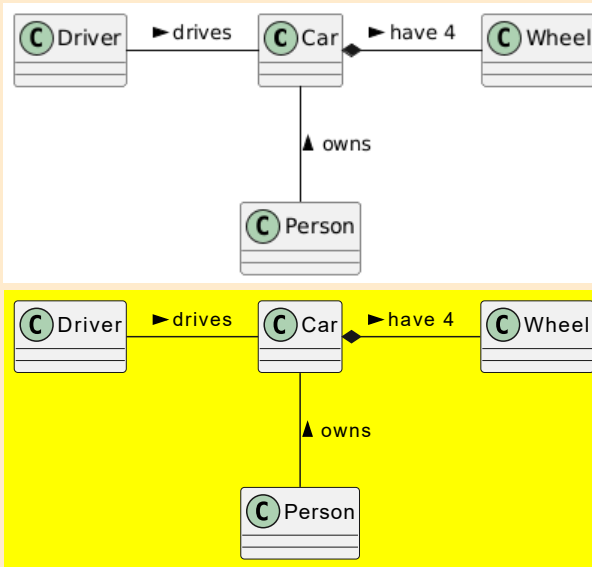
BLANK

Car diagram



```
@startuml
class Car

Driver - Car : drives >
Car *- Wheel : have 4 >
Car -- Person : < owns
@enduml
```




```

@startuml
' Configure the modern style approach with CSS
' Try polyline instead of ortho
skinparam linetype polyline

<style>
/* Global settings */
diagram {
backgroundColor: white;
}

/* Class styling */
class {
BackgroundColor: #FFFFFFEE;
BorderColor: #333333;
BorderThickness: 1;
BorderRadius: 8;
FontColor: #333333;
FontSize: 12;
}

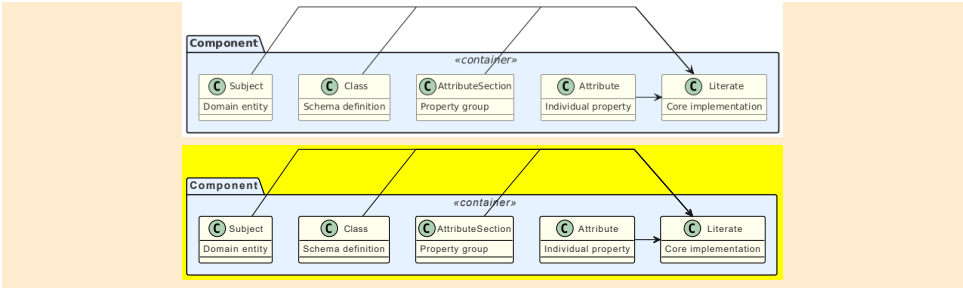
/* Arrow styling */
arrow {
Color: #333333;
Thickness: 1.5;
}

/* Package styling */
package {
BackgroundColor: #E6F2FF;
BorderColor: #336699;
BorderThickness: 3;
FontColor: #333333;
}

/* Custom style for Component */
.container {
BackgroundColor: #E6F2FF;
BorderColor: #336699;
BorderThickness: 3;
BorderRadius: 10;
}
</style>

```

Fancy Plant UML



BLANK

Mind Map

PlanUML

```

@startmindmap
* Component
** Literate
*** Subject
*** Class
*** AttributeSection
*** Attribute
@endmindmap
@startuml
!include <C4/C4_Component>

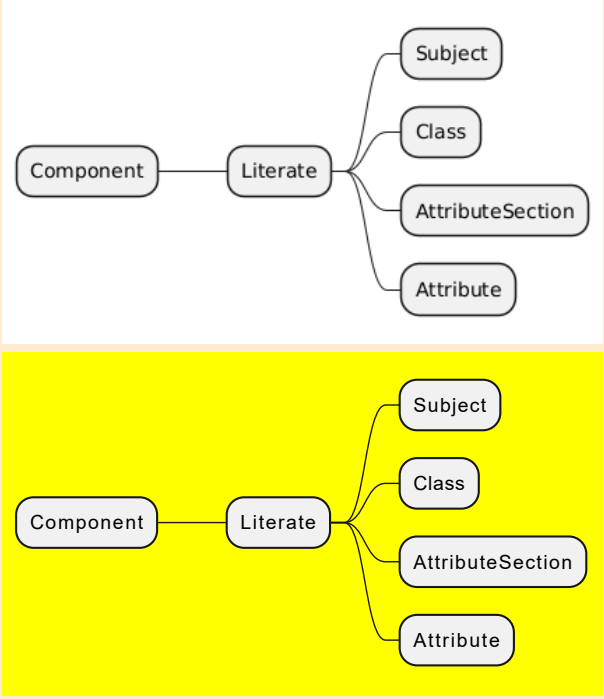
Person(user, "User")
Container_Boundary(component, "Component") {
Component(literate, "Literate", "Core implementation")
Component(subject, "Subject", "Domain entity")
Component(class, "Class", "Schema definition")
Component(attributeSection, "AttributeSection", "Property group")
Component(attribute, "Attribute", "Individual property")
}

Rel(subject, literate, "extends")
Rel(class, literate, "extends")
Rel(attributeSection, literate, "extends")
Rel(attribute, literate, "extends")
@enduml

@startjson
{
"Component": {
"Literate": {"description": "Core implementation"},
"Subject": {"description": "Domain entity", "extends": "Literate"},
"Class": {"description": "Schema definition", "extends": "Literate"},
"AttributeSection": {"description": "Property group", "extends":
"Literate"},
"Attribute": {"description": "Individual property", "extends":
"Literate"}
}
}
@endjson

```


Mind Map



BLANK


```
@startjson
{
  "Component": {
    "Literate": ["description", "Core implementation"],
    "Subject": {"description": "Domain entity", "extends": "Literate"},
    "Class": {"description": "Schema definition", "extends": "Literate"},
    "AttributeSection": {"description": "Property group", "extends":
    "Literate"},
    "Attribute": {"description": "Individual property", "extends":
    "Literate"}
  }
}
@endjson
```

