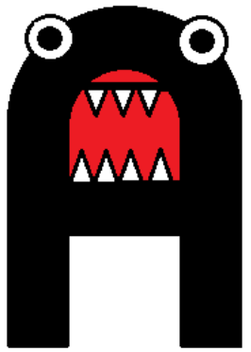
RAGE

Abschlusspräsentation

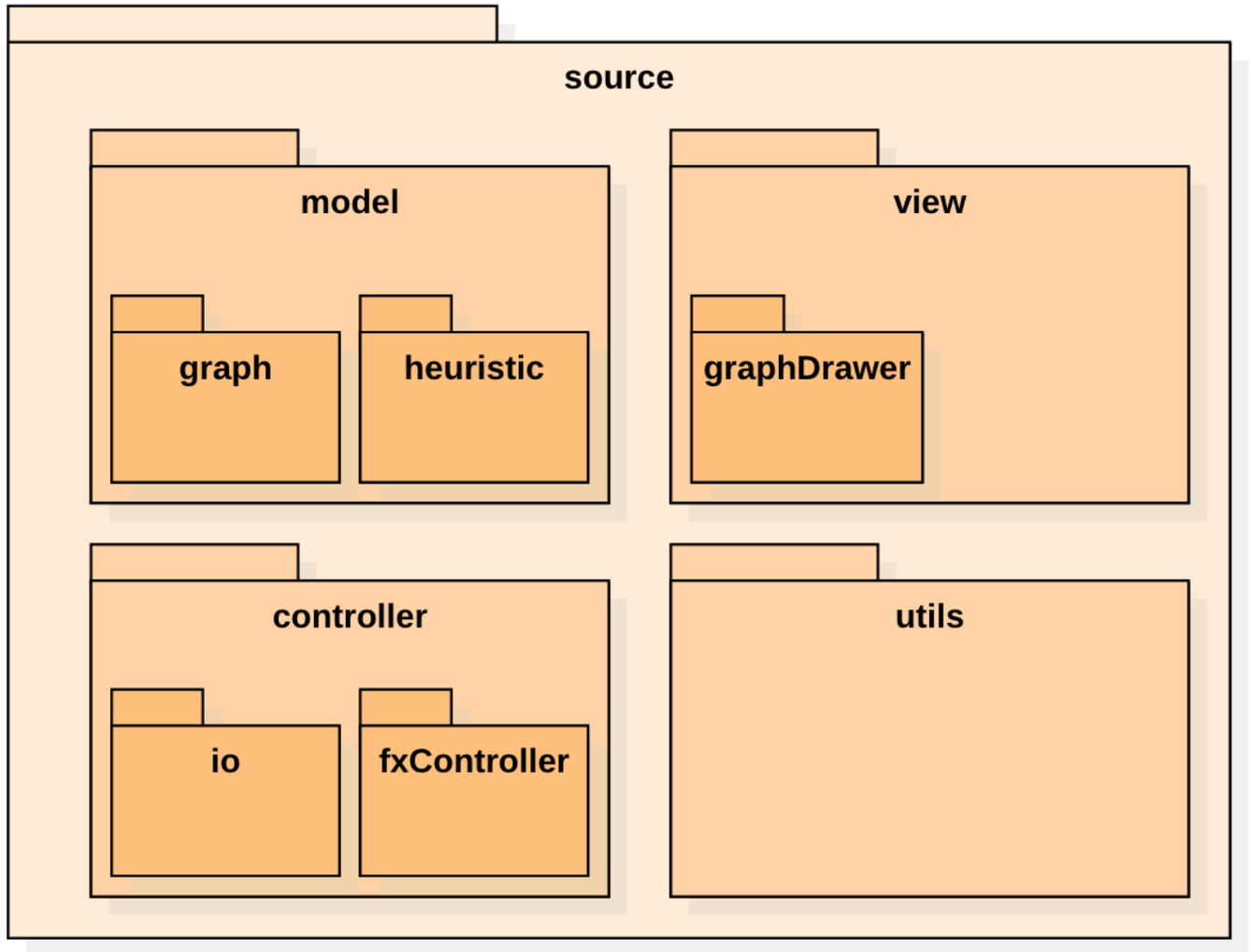
Entwurf

gehalten von Jonas Linßen

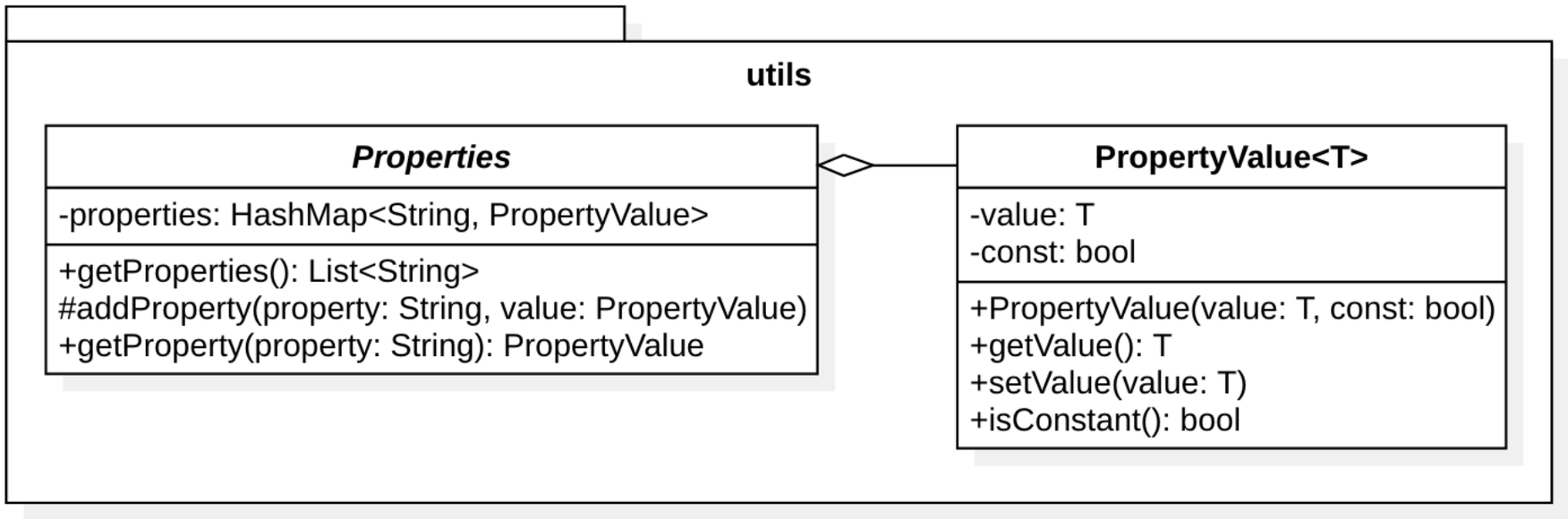
Inhalt



Allgemeiner Entwurf — Paketstruktur



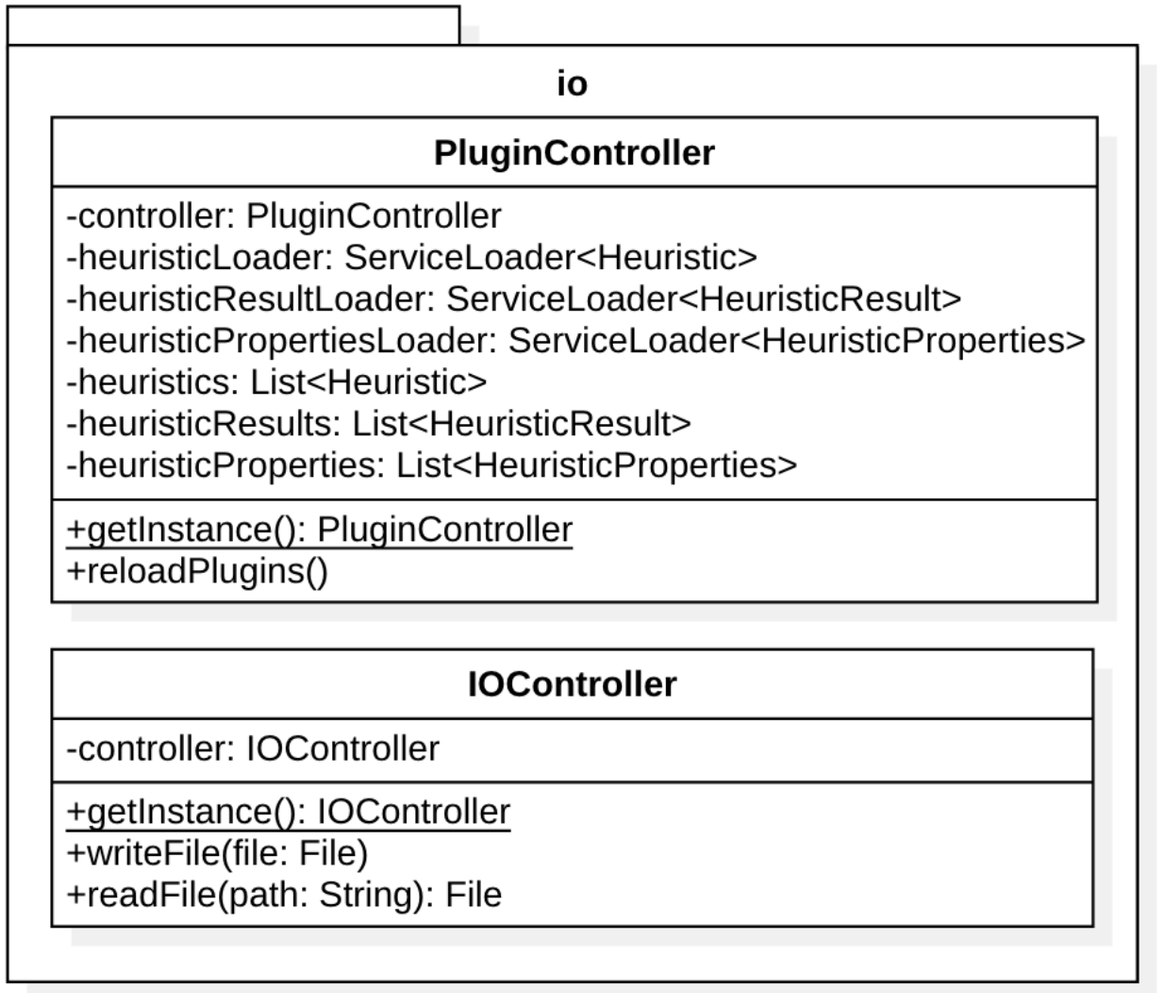
Allgemeiner Entwurf — Properties Klasse



Adaptiver Key-Value Speicher ermöglicht
⟶ Kommunikation über alle Ebenen
⟶ einfache Anpassung der Benutzeroberfläche

Fest spezifizierte Keywords
⟶ ermöglichen einfache Übersetzung der Benutzeroberfläche

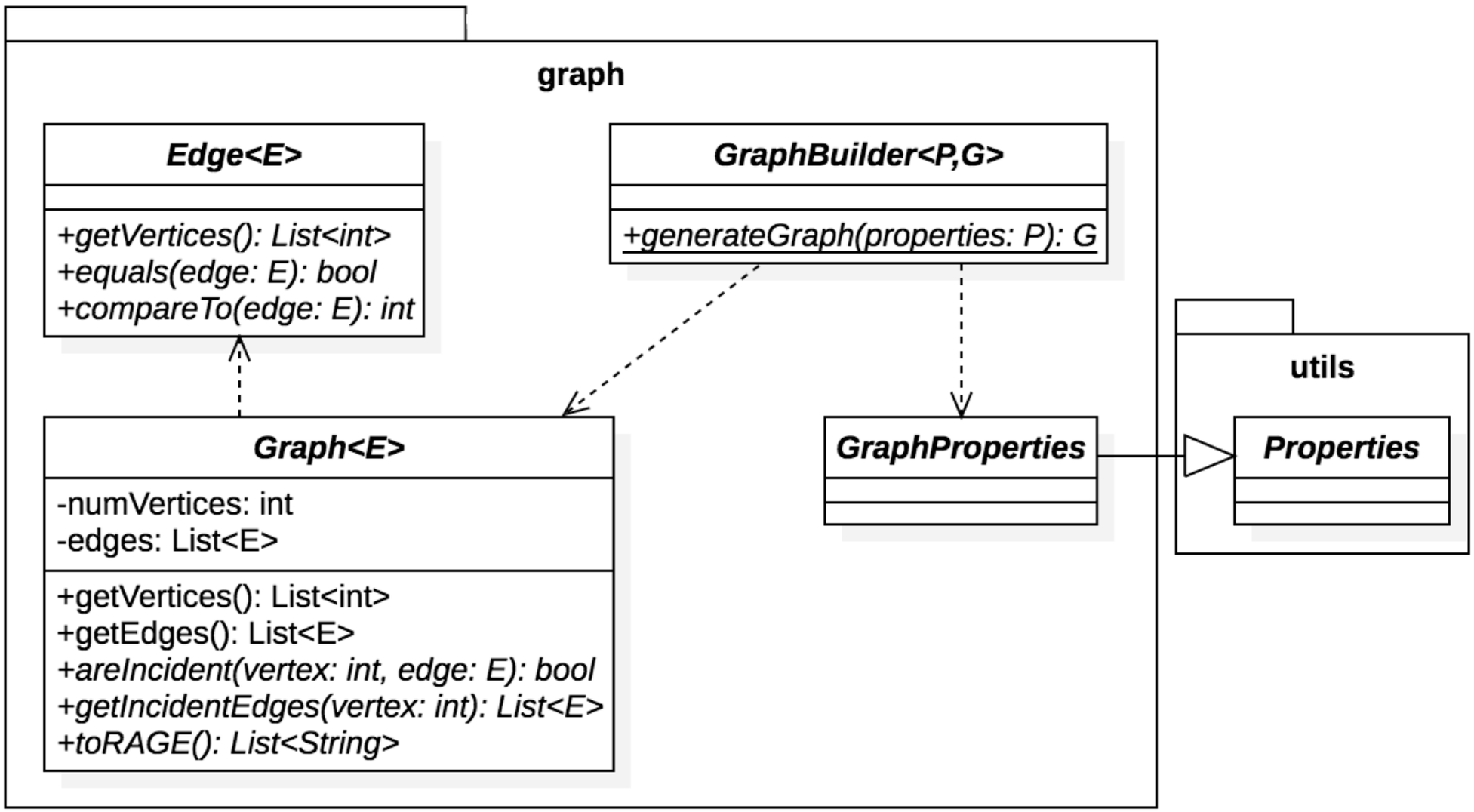
Allgemeiner Entwurf — Speichern und Laden



Singleton - Entwurfsmuster
⟶ vereinfacht Threadsafety

PluginController behält Übersicht über alle geladenen Plugins

Model — Graphen



Kapselt die abstrakte Struktur eines Graphen

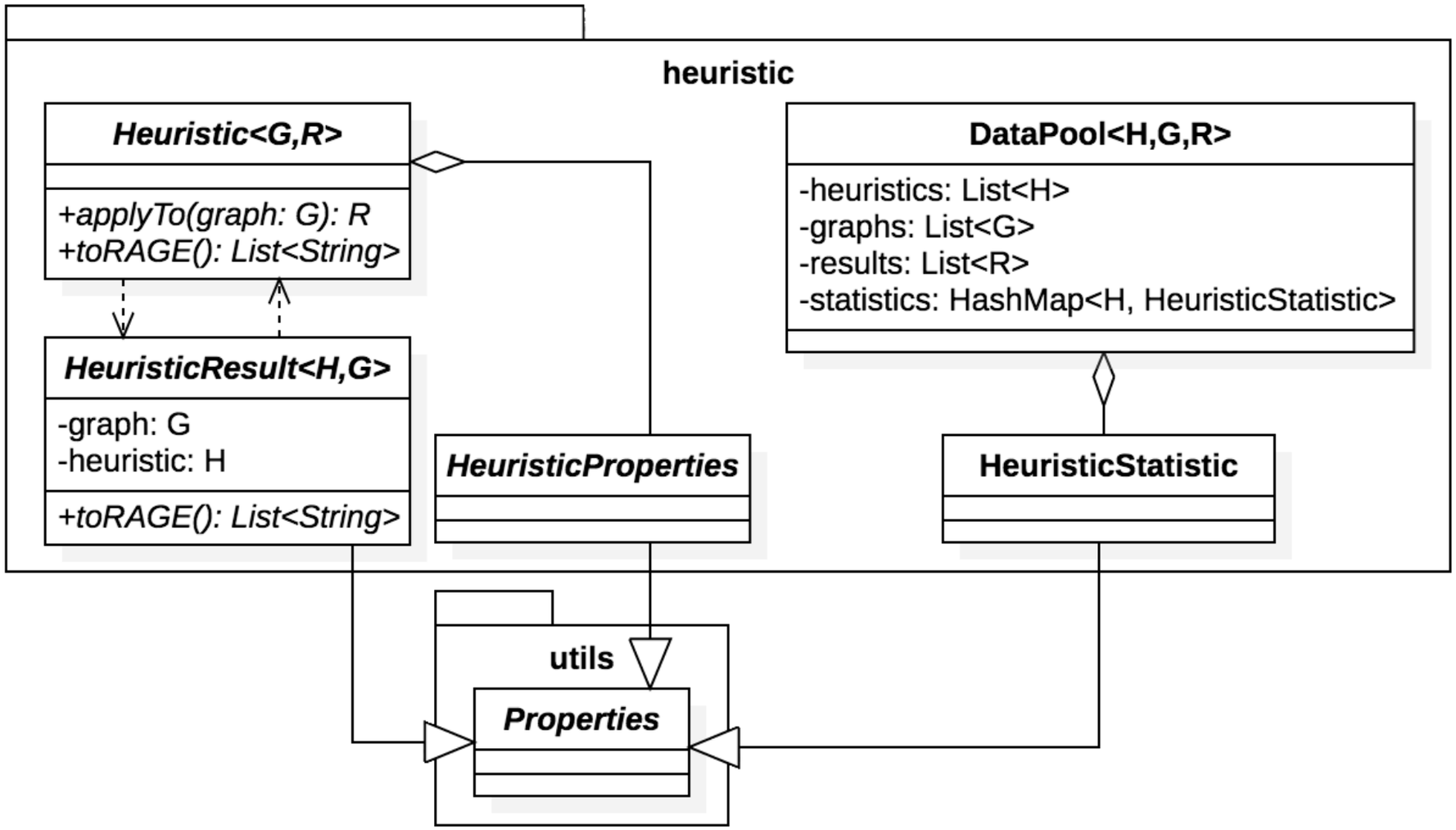
⟶ konkreter Graphentyp über Kantentyp gegeben

⟶ weitere Eigenschaften / Inzidenzrelationen abhängig vom Graph

⟶ Knotenreihenfolge implizit, Kantenreihenfolge explizit

Factory - Entwurfsmuster zur Graphgenerierung

Model — Heuristiken



Heuristik stets abhängig von Graphentyp G und Ergebnistyp R
⟶ Anwendung sollte deterministisch sein

DataPool
⟶ wendet alle Heuristiken auf alle Graphen an
⟶ sammelt Statistiken über die Heuristiken

Model — Heuristiken — EFLGreedyOne

Vermutung:
Jeder einfache Hypergraph mit n Knoten hat eine valide Kantenfärbung mit n Farben

temporäres Speichern der Hyperkanten mit genau einer freien Farbe

```
For any vertex v in order of a breadth first search
   for hyperedge e incident to v
      while there are hyperedges f with exactly one free color
         take that f with minimal index
         color f with the minimally used free color of f
      if e is colored
         continue
      if e cannot be colored
         return incomplete coloring
      color e with the minimally used free color of e
return complete coloring
```

Model — Heuristiken — TCMixedGreedySet

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen der zu einem gemeinsamen Knoten inzidenten und unkolorierten Kanten mit minimaler Flexibilität
falls Knoten ungefärbt: Varianten mit und ohne Knoten

```
while there is such a minimal set
   find set X of minimal flexibility belonging to the vertex v
   with minimal index
   if X has negative flexibility
      return incomplete coloring
   if v is uncolored and in X
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color
return complete coloring
```

Model — Heuristiken — TCMixedGreedyCon

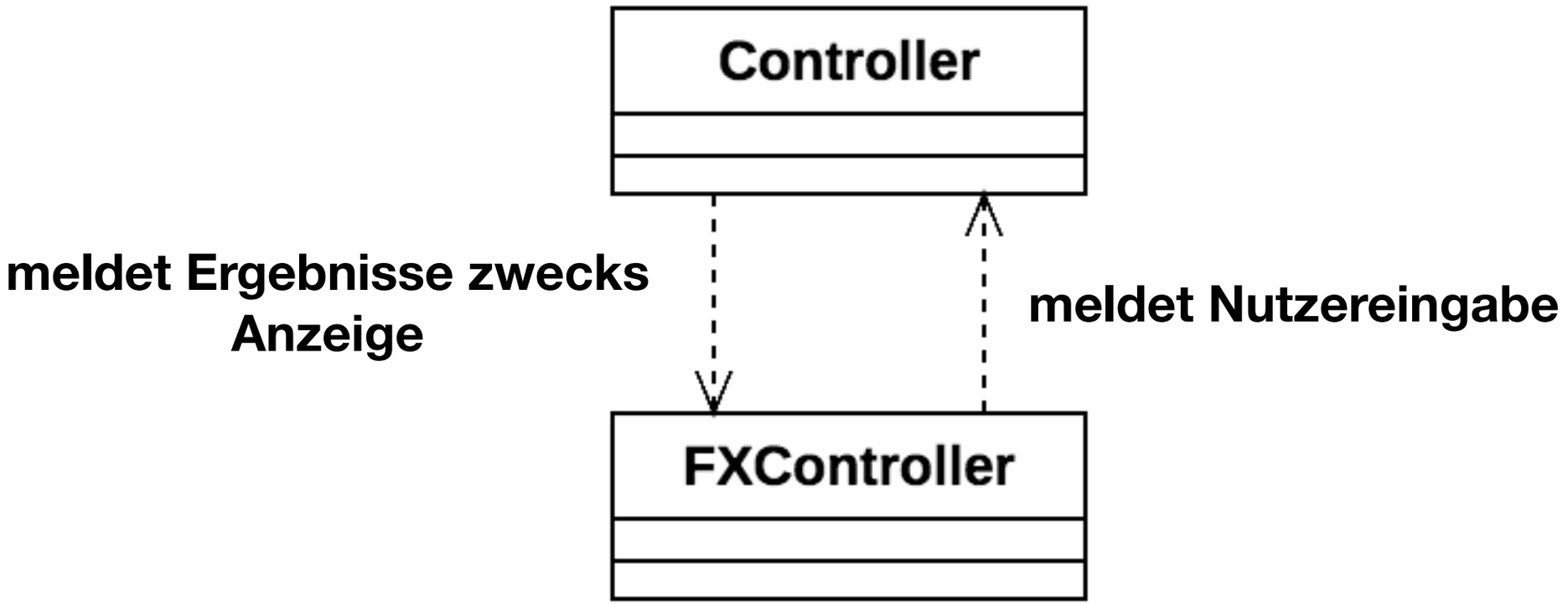
Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen zusammenhängender unkolorierten Knoten und Kanten (bis zu einer gewissen Größe) mit minimaler Flexibilität

```
while there are uncolored vertices or uncolored edges
   find connected set X of minimal flexibility and minimal index
   if X has negative flexibility
      return incomplete coloring
   for any vertex v in X
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color
return complete coloring
```

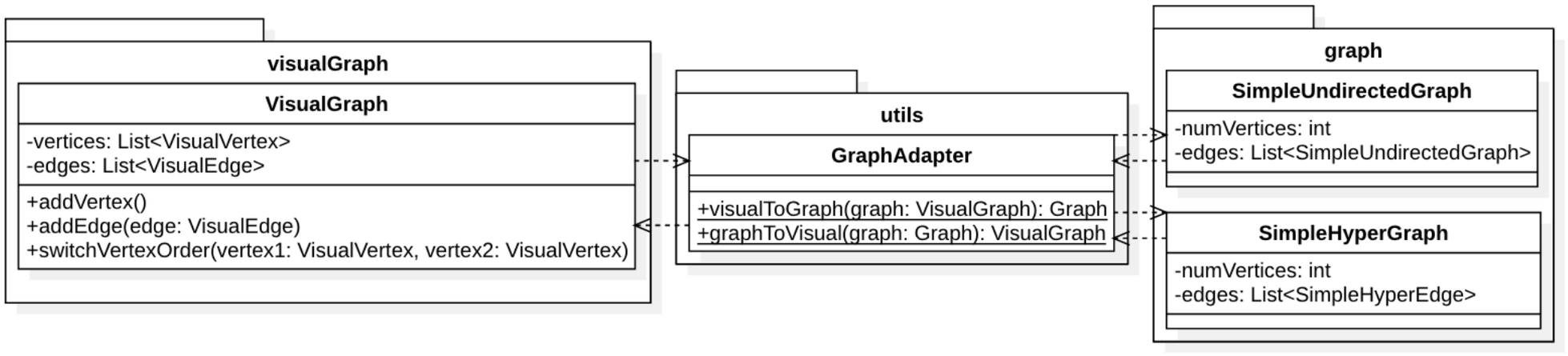
Controller / View — Beschreibung

Kontrolliert Datenverarbeitung



Zuständig für Nutzerinteraktion + Darstellung

Controller / View — Graph-Editor

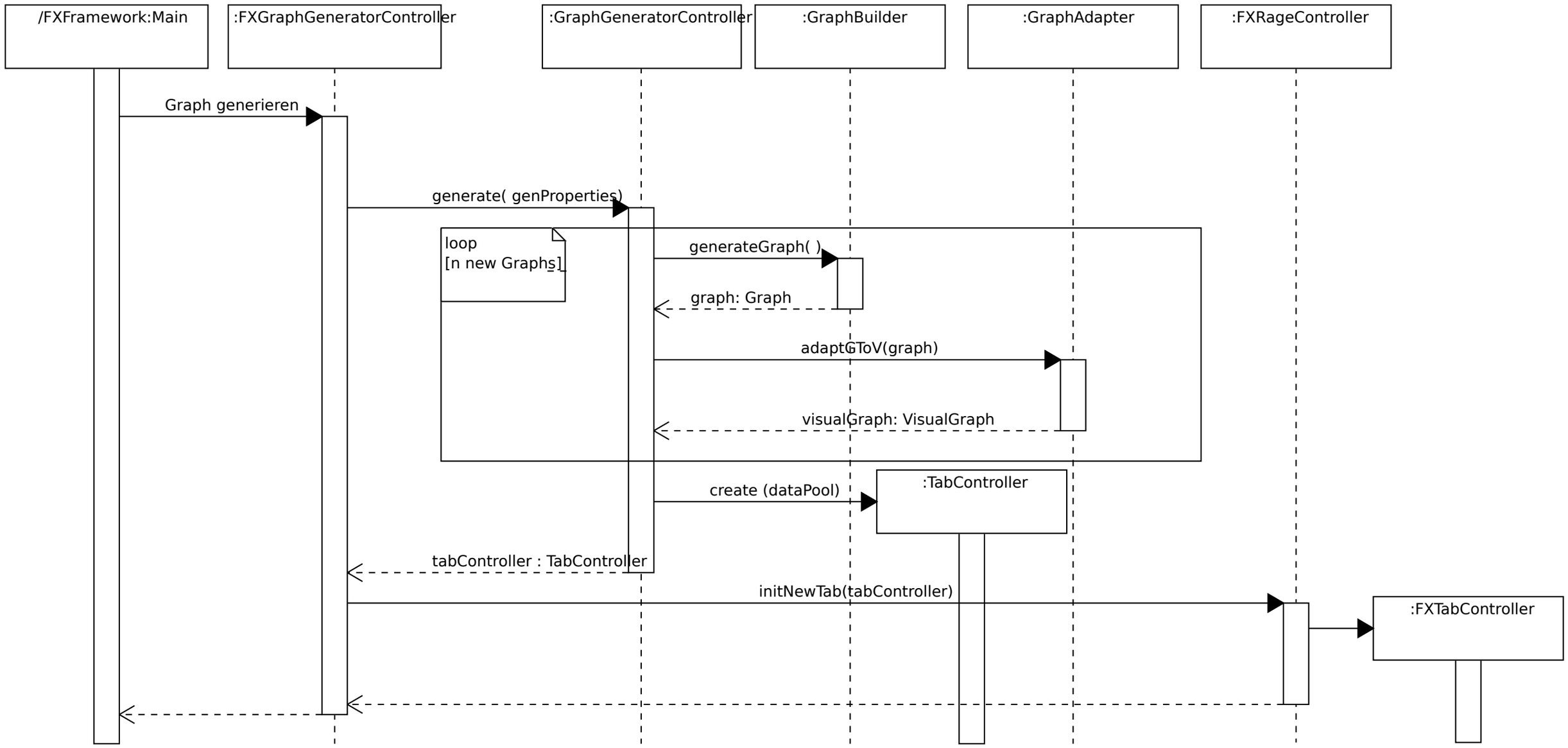


Nutzerinteraktion über VisualGraph

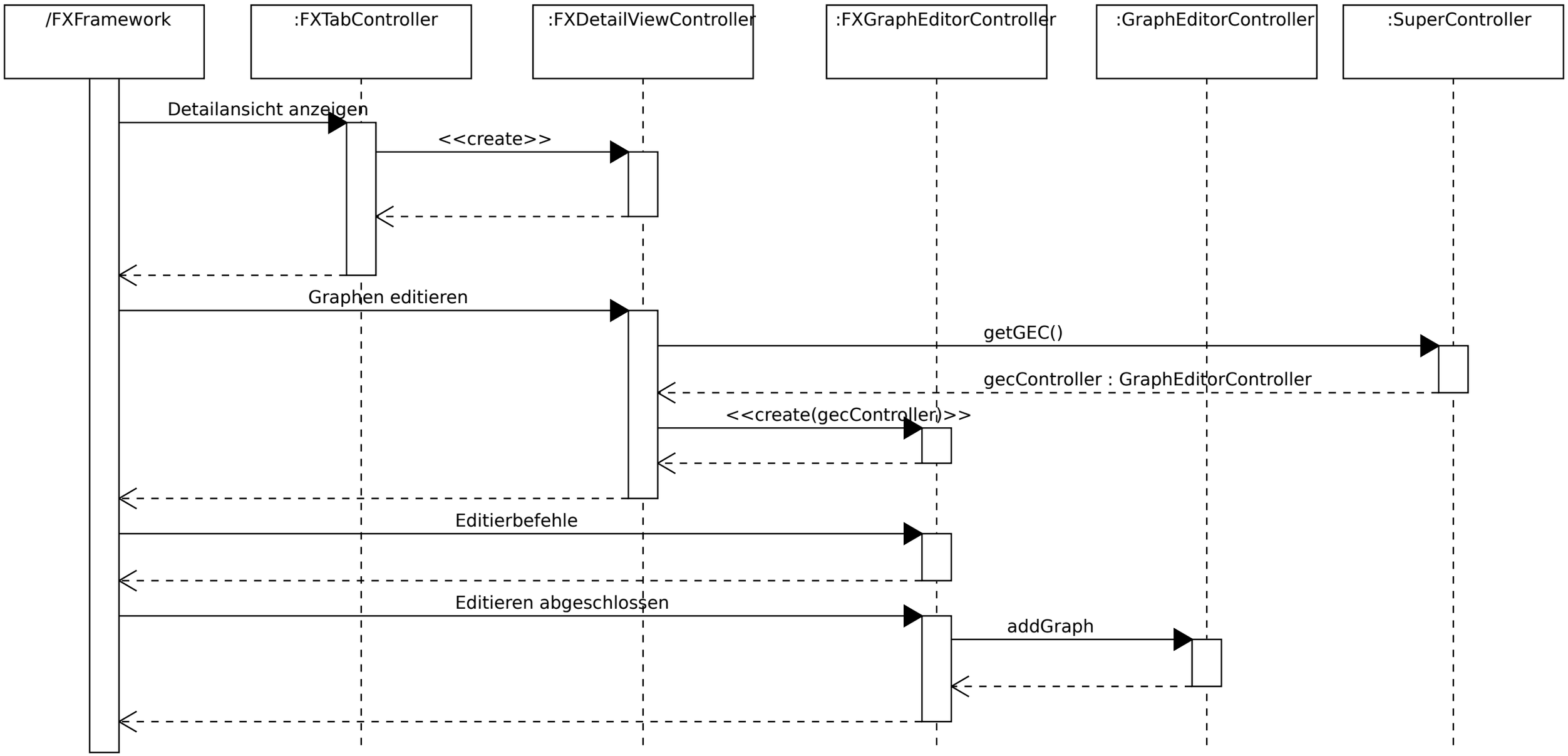
⟶ nach Bestätigung Umwandlung zu passendem Graph aus Model

⟶ Adapter Entwurfsmuster ermöglicht Erweiterbarkeit

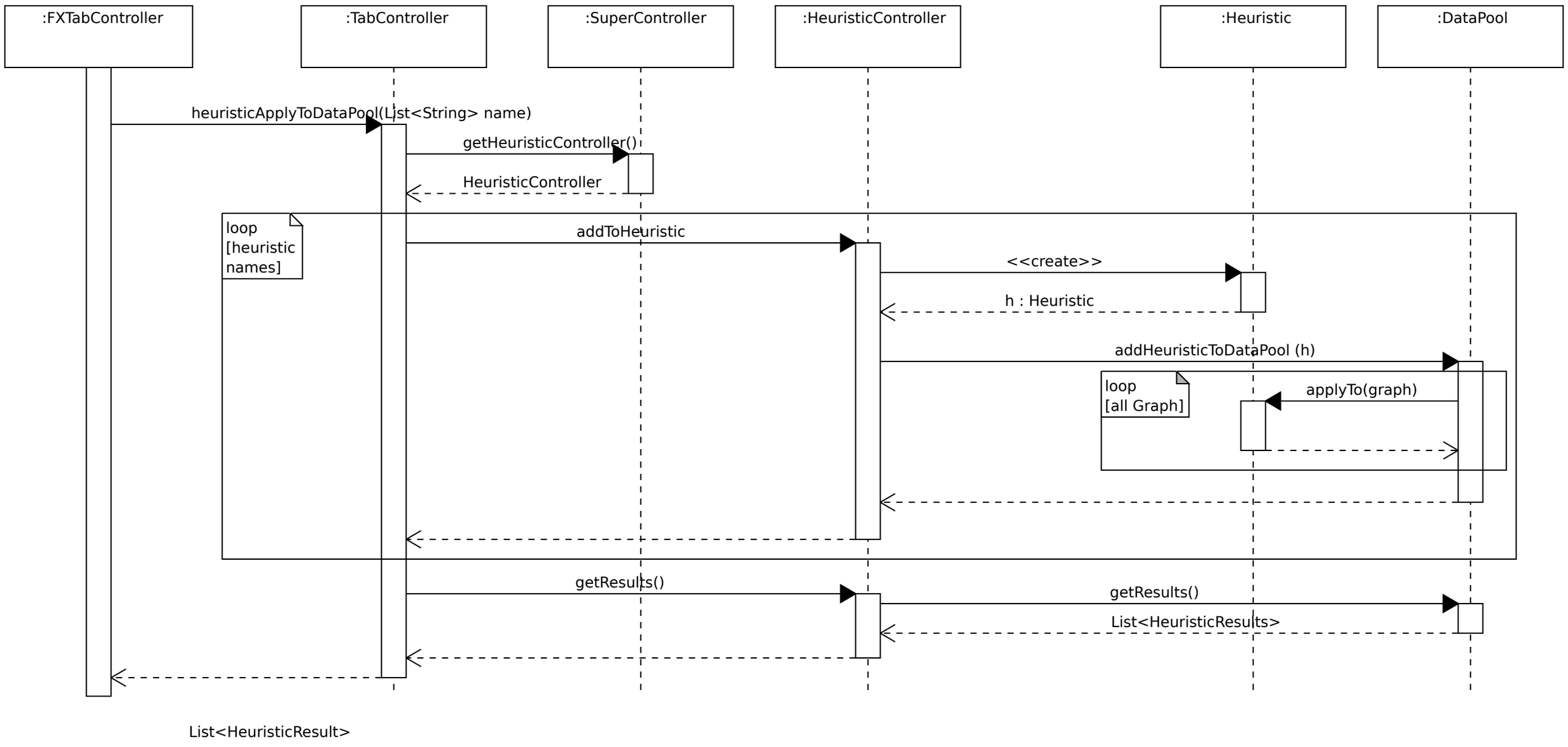
Sequenzdiagramme — Graph generieren



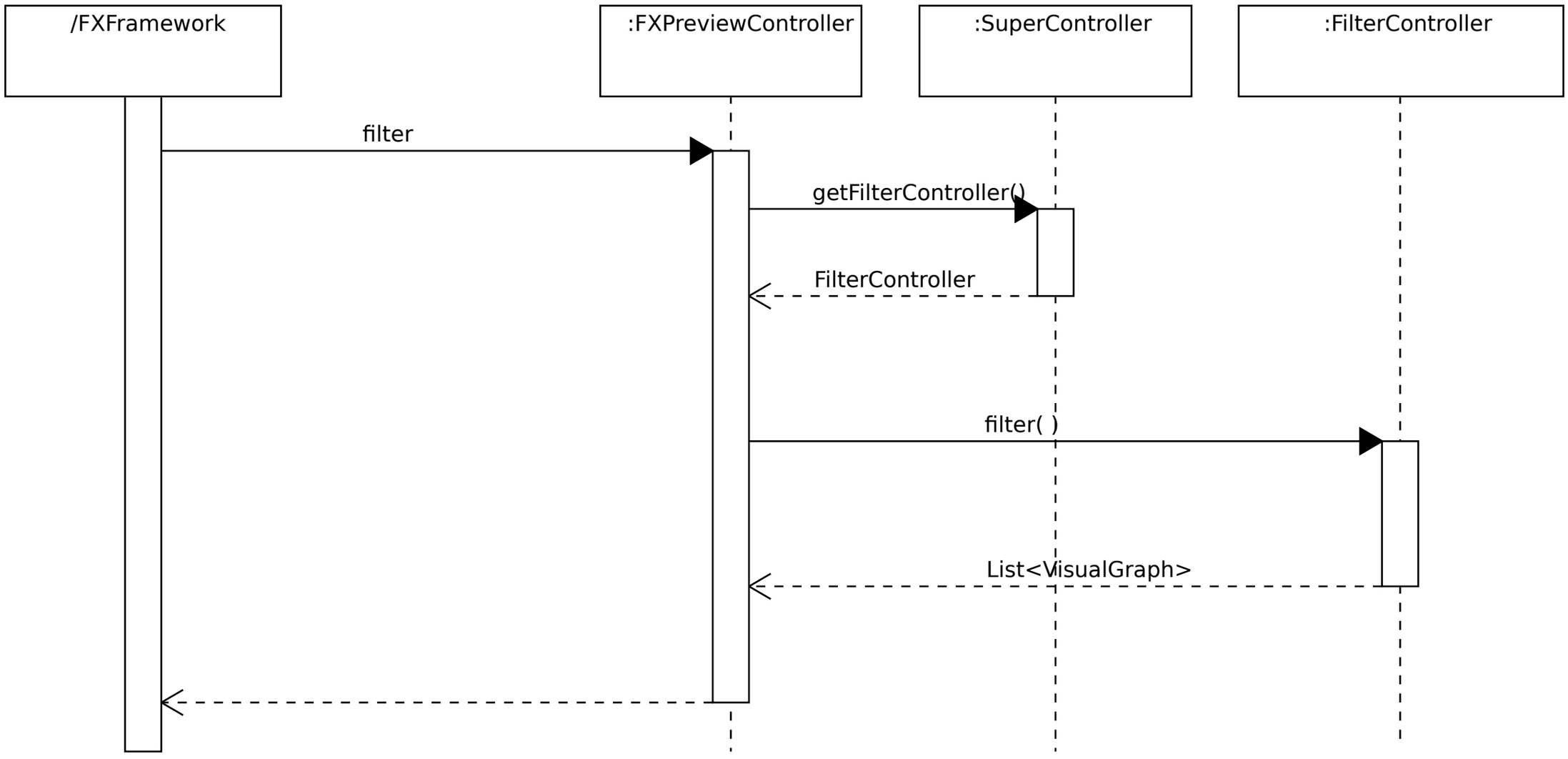
Sequenzdiagramme — Graph editieren



Sequenzdiagramme — Heuristiken anwenden



Sequenzdiagramme — Filtern



Model — Heuristiken — TCGreedy

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color

for every vertex v in order of a breadth first search
   for any uncolored edge e incident to v
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCGreedyOne

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern unkolorierter Kanten mit genau einer freien Farbe

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color
```

```
for every vertex v in order of a breadth first search
   for any uncolored edge e incident to v
      for any uncolored edge f with exactly one free color
         if f cannot be colored
            return incomplete coloring
         color f with minimally used free color
      if e is colored
         continue
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCGreedyFew

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern einer Liste unkolorierter Kanten sortiert nach Anzahl ihrer freien Farben und ihrem Index

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color

for every vertex v in order of a breadth first search
   for any uncolored edge e incident to v
      for any uncolored edge f before e in said list
         if f cannot be colored
            return incomplete coloring
         color f with minimally used free color
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCGreedySet

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen der zu einem gemeinsamen Knoten inzidenten und unkolorierten Kanten mit minimaler Flexibilität

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color

while there is such a minimal set
   find set X of minimal flexibility belonging to the vertex v
   with minimal index
   if X has negative flexibility
      return incomplete coloring
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCGreedyCon

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen zusammenhängender unkolorierten Kanten (bis zu einer gewissen Größe) mit minimaler Flexibilität

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color

while there is such a minimal set
   find set X of minimal flexibility and minimal index
   if X has negative flexibility
      return incomplete coloring
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCMixedGreedy

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

```
for every vertex v in order of a breadth first search
   if v cannot be colored
      return incomplete coloring
   color v with minimally used free color

   for any uncolored edge e incident to v
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCMixedGreedyOne

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von unkolorierten Knoten und Kanten mit genau einer freien Farbe

```
for every vertex v in order of a breadth first search
   while there are objects with exactly one free color
      color them (preferring vertices over edges)

   if v is uncolored
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color

   for any uncolored edge e incident to v
      while there are objects with exactly one free color
         color them (preferring vertices over edges)

      if e is uncolored
         if e cannot be colored
            return incomplete coloring
         color e with minimally used free color

return complete coloring
```

Model — Heuristiken — TCMixedGreedyFew

Vermutung:

Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von unkolorierten Knoten und Kanten sortiert nach Anzahl ihrer freien Farben

```
for every vertex v in order of a breadth first search
   while there is a vertex w with less free colors than v and
   lower index
      if w cannot be colored
         return incomplete coloring
      color w with minimally used free color
   if v is uncolored
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color
   for any uncolored edge e incident to v
      while there is an edge f with less free colors than e and
      lower index
         if f cannot be colored
            return incomplete coloring
         color f with minimally used free color
      if e is uncolored
         if e cannot be colored
            return incomplete coloring
         color e with minimally used free color
return complete coloring
```

Model — Heuristiken — TCMixedGreedySet

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen der zu einem gemeinsamen Knoten inzidenten und unkolorierten Kanten mit minimaler Flexibilität

falls Knoten ungefärbt: Varianten mit und ohne Knoten

```
while there is such a minimal set
   find set X of minimal flexibility belonging to the vertex v
   with minimal index
   if X has negative flexibility
      return incomplete coloring
   if v is uncolored and in X
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color
return complete coloring
```

Model — Heuristiken — TCMixedGreedyCon

Vermutung:
Jeder einfache ungerichtete Graph mit Maximalgrad D hat eine valide Totalfärbung mit D+2 Farben

temporäres Speichern von Teilmengen zusammenhängender unkolorierten Knoten und Kanten (bis zu einer gewissen Größe) mit minimaler Flexibilität

```
while there are uncolored vertices or uncolored edges
   find connected set X of minimal flexibility and minimal index
   if X has negative flexibility
      return incomplete coloring
   for any vertex v in X
      if v cannot be colored
         return incomplete coloring
      color v with minimally used free color
   for edge e in X
      if e cannot be colored
         return incomplete coloring
      color e with minimally used free color
return complete coloring
```

Model — Heuristiken — EFLGreedyOne

Vermutung:
Jeder einfache Hypergraph mit n Knoten hat eine valide Kantenfärbung mit n Farben

temporäres Speichern der Hyperkanten mit genau einer freien Farbe

```
For any vertex v in order of a breadth first search
   for hyperedge e incident to v
      while there are hyperedges f with exactly one free color
         take that f with minimal index
         color f with the minimally used free color of f
      if e is colored
         continue
      if e cannot be colored
         return incomplete coloring
      color e with the minimally used free color of e
return complete coloring
```