# Table of Contents

GTTG.Model.ViewModel.Infrastructure.Railways
  IRailwayViewFactory<TRailwayView, TStationView, TTrackView>
  RailwayView<TStationView, TTrackView>
  StrategyRailwayView<TStationView, TTrackView>
GTTG.Model.ViewModel.Infrastructure.Stations
  IStationViewFactory<TStationView, TTrackView>
  StationView<TTrackView>
  StrategyStationView<TTrackView>
GTTG.Model.ViewModel.Infrastructure.Tracks
  ITrackViewFactory<TTrackView>
  TrackView
GTTG.Model.ViewModel.Traffic
  ITrafficViewFactory<TTrafficView, TTrainView, TTrain>
  ITrainViewFactory<TTrainView, TTrain>
  StrategyTrainView<TStrategy, TTrain>
  TrafficView<TTrainView, TTrain>
  TrainView<TTrain>

# Namespace GTTG.Model.Lines

## Classes

### LinePaint

Line with modifiable stroke width.

### TrainPath

Creates train path from TrainEvent of updated schedule.

## Interfaces

### ITrainPath

Contract for train path created from schedule.

# Interface ITrainPath

Contract for train path created from schedule.

Namespace: GTTG.Model.Lines

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITrainPath
```

## Properties

### Item[Int32]

Points of train path.

Declaration

```
SKPoint this[int index] { get; }
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Int32 | index | Index in train path with number of points equal to PointCount. |

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKPoint | Point at specified index. |

Exceptions

| TYPE | CONDITION |
| --- | --- |
| System.ArgumentOutOfRangeException | Lower than 0 or higher or equal than PointCount. |

### LinePaint

Line to create and draw train path from.

Declaration

```
LinePaint LinePaint { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| LinePaint | |

### PathColor

Color of path.

Declaration

```
SKColor PathColor { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKColor | |

### PointCount

Number of points in path.

Declaration

```
int PointCount { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Int32 | |

### PointsByTrainPathEvents

Maps TrainEvent of updated schedule to index of point in path and the point itself.

Declaration

```
IReadOnlyDictionary<TrainEvent, (int Index, SKPoint PathPoint)> PointsByTrainPathEvents { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IReadOnlyDictionary<TrainEvent, System.ValueTuple<System.Int32, SKPoint>> | |

### TrainPathEvents

Provides all mapped movement events of updated schedule.

Declaration

```
IReadOnlyList<TrainEvent> TrainPathEvents { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IReadOnlyList<TrainEvent> | |

## Methods

### Arrange()

Arranges points in path.

Declaration

```
void Arrange()
```

### Clear()

Reset train path and removes points. Needs to create new with Update(ImmutableArray<TrainEvent>).

Declaration

```
void Clear()
```

### DistanceFromPoint(SKPoint)

Measures closest distance of train path to provided point.

Declaration

```
float DistanceFromPoint(SKPoint point)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---|---|---|
| SKPoint | point | Provided point to determine distance from. |

Returns

| TYPE | DESCRIPTION |
|---|---|
| System.Single | Closest distance of path to provided point. |

### Draw(DrawingCanvas)

Draws train path on canvas.

Declaration

```
void Draw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---|---|---|
| DrawingCanvas | drawingCanvas | Drawing canvas to draw onto. |

### MeasurePathStrokeWidth()

Measures maximal path stroke width with ornaments included.

Declaration

```
float MeasurePathStrokeWidth()
```

Returns

| TYPE | DESCRIPTION |
|---|---|
| System.Single | Measured stroke width. |

### Update(ImmutableArray<TrainEvent>)

Updates value from which train path is created.

Declaration

```
void Update(ImmutableArray<TrainEvent> schedule)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ImmutableArray<TrainEvent> | schedule | Schedule of events converted path. |

# Class LinePaint

Line with modifiable stroke width.

Inheritance

System.Object
LinePaint

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Model.Lines

Assembly: cs.temp.dll.dll

Syntax

```
public class LinePaint
```

## Constructors

### LinePaint(Single, SKColor)

Creates line and paint with desired size from color.

Declaration

```
public LinePaint(float desiredStrokeWidth, SKColor color)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Single | desiredStrokeWidth | Desired height of line. |
| SKColor | color | Color set to paint. |

### LinePaint(Single, SKPaint)

Creates line with desired size from paint.

Declaration

```
public LinePaint(float desiredStrokeWidth, SKPaint paint)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Single | desiredStrokeWidth | Desired height of line. |

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPaint | paint | Paint with stroke width to modify. |

## Properties

### ArrangedStrokeWidth

Actual stroke width.

Declaration

```
public float ArrangedStrokeWidth { get; protected set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | |

### DesiredStrokeWidth

Default stroke width.

Declaration

```
public float DesiredStrokeWidth { get; protected set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | |

### Paint

Wrapped paint with modified stroke width.

Declaration

```
public SKPaint Paint { get; protected set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKPaint | |

## Methods

### Arrange(Single)

Assigns new value to actual stroke width ArrangedStrokeWidth.

Declaration

```
public void Arrange(float height)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Single | height | Arranged stroke width to use. Modifies Paint stroke width. |

### Clone()

Creates new LinePaint.

Declaration

```
public LinePaint Clone()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| LinePaint | LinePaint with cloned value and instance of Paint |

### Measure()

Measures desired stroke width.

Declaration

```
public float Measure()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | Value of DesiredStrokeWidth. |

# Class TrainPath

Creates train path from TrainEvent of updated schedule.

Inheritance

System.Object
TrainPath

Implements

ITrainPath

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: GTTG.Model.Lines

Assembly: cs.temp.dll.dll

Syntax

```
public class TrainPath : ITrainPath
```

## Constructors

### TrainPath(IViewProvider, ISegmentRegistry<LineType, MeasureableSegment>, LinePaint)

Creates empty train path.

Declaration

```
public TrainPath(IViewProvider viewProvider, ISegmentRegistry<LineType, MeasureableSegment> segmentRegistry,
LinePaint linePaint)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| IViewProvider | viewProvider | Converter of date time event values to horizontal positions. |
| ISegmentRegistry<LineType, MeasureableSegment> | segmentRegistry | Registry of lines providing it's vertical position. |
| LinePaint | linePaint | Line to create path from. |

## Fields

### _PointsByTrainPathEvents

Maps TrainEvent to index in SkTrainPath and it's point.

Declaration

```
protected readonly Dictionary<TrainEvent, (int Index, SKPoint PathPoint)> _PointsByTrainPathEvents
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.Dictionary<TrainEvent, System.ValueTuple<System.Int32, SKPoint>> | |

### _trainPathEvents

TrainEvent as schedule to be arranged to form SkTrainPath.

Declaration

```
protected readonly List<TrainEvent> _trainPathEvents
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.List<TrainEvent> | |

### SegmentRegistrations

Keeps track of segments where line was registered for measure.

Declaration

```
protected readonly List<MeasureableSegment> SegmentRegistrations
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.List<MeasureableSegment> | |

### SegmentRegistry

Provides segments of position of horizontal lines where points SkTrainPath are placed.

Declaration

```
protected readonly ISegmentRegistry<LineType, MeasureableSegment> SegmentRegistry
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| ISegmentRegistry<LineType, MeasureableSegment> | |

### SkTrainPath

of points that forms path of train in .

Declaration

```
protected readonly SKPath SkTrainPath
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKPath | |

### TrainLinePaint

LinePaint to draw SkTrainPath.

Declaration

```
protected readonly LinePaint TrainLinePaint
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| LinePaint | |

### ViewProvider

View provider to converts System.DateTime values to horizontal values.

Declaration

```
protected readonly IViewProvider ViewProvider
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| IViewProvider | |

## Properties

### Item[Int32]

Points of train path.

Declaration

```
public SKPoint this[int index] { get; }
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Int32 | index | Index in train path with number of points equal to PointCount. |

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKPoint | Point at specified index. |

Exceptions

| TYPE | CONDITION |
| --- | --- |
| System.ArgumentOutOfRangeException | Lower than 0 or higher or equal than PointCount. |

### LinePaint

Line to create and draw train path from.

Declaration

```
public LinePaint LinePaint { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| LinePaint | |

### PathColor

Color of path.

Declaration

```
public SKColor PathColor { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKColor | |

### PointCount

Number of points in path.

Declaration

```
public int PointCount { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Int32 | |

### PointsByTrainPathEvents

Maps TrainEvent of updated schedule to index of point in path and the point itself.

Declaration

```
public IReadOnlyDictionary<TrainEvent, (int Index, SKPoint PathPoint)> PointsByTrainPathEvents { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IReadOnlyDictionary<TrainEvent, System.ValueTuple<System.Int32, SKPoint>> | |

### TrainPathEvents

Provides all mapped movement events of updated schedule.

Declaration

```
public IReadOnlyList<TrainEvent> TrainPathEvents { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IReadOnlyList<TrainEvent> | |

## Methods

### Arrange()

Arranges points in path.

Declaration

```
public void Arrange()
```

### Clear()

Reset train path and removes points. Needs to create new with Update(ImmutableArray<TrainEvent>).

Declaration

```
public void Clear()
```

### DistanceFromPoint(SKPoint)

Measures closest distance of train path to provided point.

Declaration

```
public float DistanceFromPoint(SKPoint point)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | point | Provided point to determine distance from. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | Closest distance of path to provided point. |

### Draw(DrawingCanvas)

Draws train path on canvas.

Declaration

```
public virtual void Draw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | Drawing canvas to draw onto. |

### MeasurePathStrokeWidth()

Measures maximal path stroke width with ornaments included.

Declaration

```
public float MeasurePathStrokeWidth()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | Measured stroke width. |

### Update(ImmutableArray<TrainEvent>)

Updates value from which train path is created.

Declaration

```
public void Update(ImmutableArray<TrainEvent> schedule)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ImmutableArray<TrainEvent> | schedule | Schedule of events converted path. |

### Implements

ITrainPath

# Namespace GTTG.Model.Model.Events

### Classes

TrainEvent

Base class for train events.

### Enums

TrainEventType

Type of train movement event in station.

# Class TrainEvent

Base class for train events.

Inheritance

System.Object
TrainEvent

Namespace: GTTG.Model.Model.Events

Assembly: cs.temp.dll.dll

Syntax

```
public class TrainEvent : ObservableObject
```

## Constructors

### TrainEvent(DateTime, Station, Track, TrainEventType)

Creates a new event.

Declaration

```
public TrainEvent(DateTime dateTime, Station station, Track track, TrainEventType trainEventType)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---|---|---|
| System.DateTime | dateTime | Time value of event. |
| Station | station | Station where event occurs. |
| Track | track | Track of the station where event occurs. |
| TrainEventType | trainEventType | Type of the event. |

Exceptions

| TYPE | CONDITION |
|---|---|
| System.ArgumentException | `station` does not contain `track`. |

## Properties

### DateTime

System.DateTime when event occurs.

Declaration

```
public DateTime DateTime { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.DateTime | |

### IsArrival

Determines whether event is arrival to the station.

Declaration

```
public bool IsArrival { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### IsDeparture

Determines whether the event is departure from the station.

Declaration

```
public bool IsDeparture { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### IsPassage

Determines whether the event is passage through the station.

Declaration

```
public bool IsPassage { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### Station

Station where event occurs.

Declaration

```
public Station Station { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Station | |

### Track

Track where event occurs.

Declaration

```
public Track Track { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Track | |

### TrainEventType

Movement type of event in station.

Declaration

```
public TrainEventType TrainEventType { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| TrainEventType | |

# Enum TrainEventType

Type of train movement event in station.

Namespace: GTTG.Model.Model.Events

Assembly: cs.temp.dll.dll

Syntax

```
public enum TrainEventType
```

## Fields

| NAME | DESCRIPTION |
| --- | --- |
| Arrival | Event of train arriving to a station. |
| Departure | Event of train leaving a station. |
| Passage | Event of train passing through a station. |

# Namespace GTTG.Model.Model.Infrastructure

Classes

### Railway

Represents railway with stations which contains tracks.

### Station

Represents station which contains tracks.

### Track

Represents track in station.

# Class Railway

Represents railway with stations which contains tracks.

Inheritance

System.Object
Railway

Namespace: GTTG.Model.Model.Infrastructure

Assembly: cs.temp.dll.dll

Syntax

```
public class Railway : ObservableObject
```

## Constructors

### Railway(IEnumerable<Station>)

Initializes a new instance of the Railway with stations.

Declaration

```
public Railway(IEnumerable<Station> stations)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Collections.Generic.IEnumerable<Station> | stations | Stations placed in Stations. |

## Properties

### Stations

Stations in railway.

Declaration

```
public ImmutableArray<Station> Stations { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableArray<Station> | |

# Class Station

Represents station which contains tracks.

Inheritance

System.Object
Station

Namespace: GTTG.Model.Model.Infrastructure

Assembly: cs.temp.dll.dll

Syntax

```
public class Station : ObservableObject
```

## Constructors

### Station(IEnumerable<Track>)

Initializes a new instance of the Station with tracks.

Declaration

```
public Station(IEnumerable<Track> tracks)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Collections.Generic.IEnumerable<Track> | tracks | Tracks placed in Tracks. |

## Properties

### Tracks

Tracks in the station.

Declaration

```
public ImmutableArray<Track> Tracks { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableArray<Track> | |

# Class Track

Represents track in station.

Inheritance

System.Object

Track

Namespace: GTTG.Model.Model.Infrastructure

Assembly: cs.temp.dll.dll

Syntax

```
public class Track : ObservableObject
```

# Namespace GTTG.Model.Model.Traffic

Classes

Traffic<TTrain>

Trains as traffic in Railway.

Train

Train in railway.

# Class Traffic<TTrain>

Trains as traffic in Railway.

Inheritance

System.Object

Traffic<TTrain>

Namespace: GTTG.Model.Model.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public class Traffic<TTrain> : ObservableObject where TTrain : Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrain | |

## Constructors

### Traffic(IEnumerable<TTrain>)

Initializes a new instance of the Traffic<TTrain> with trains.

Declaration

```
public Traffic(IEnumerable<TTrain> trains)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Collections.Generic.IEnumerable<TTrain> | trains | Trains placed in Trains. |

## Properties

### Trains

Trains in traffic.

Declaration

```
public ImmutableArray<TTrain> Trains { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableArray<TTrain> | |

# Class Train

Train in railway.

Inheritance

System.Object
Train

Namespace: GTTG.Model.Model.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public class Train : ObservableObject
```

## Constructors

### Train(IEnumerable<TrainEvent>)

Initializes a new instance of the Train with it's schedule.

Declaration

```
public Train(IEnumerable<TrainEvent> schedule)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Collections.Generic.IEnumerable<TrainEvent> | schedule | Collection of TrainEvent as actual schedule of the train. |

## Properties

### Schedule

Gets or sets current schedule of train.

Declaration

```
public ImmutableArray<TrainEvent> Schedule { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableArray<TrainEvent> | |

# Namespace GTTG.Model.Strategies

### Classes

#### Container

Groups multiple into one container to positioned in strategy.

#### Strategy

Represents strategies applicable to .

### Interfaces

#### IStrategy

Represents contract for strategy implementation used by .

### Enums

#### ElementsOrder

Flag for Container to determine order of added elements.

# Class Container

Groups multiple into one container to positioned in strategy.

Inheritance

System.Object
Container

Namespace: GTTG.Model.Strategies

Assembly: cs.temp.dll.dll

Syntax

```
public class Container : ViewElement
```

## Constructors

### Container(ElementsOrder)

Creates empty container with determined order for adding elements.

Declaration

```
public Container(ElementsOrder elementsOrder)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ElementsOrder | elementsOrder | |

## Fields

### Components

Elements in container.

Declaration

```
protected readonly List<ViewElement> Components
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.List<ViewElement> | |

## Properties

### ElementsOrder

Determines from which side are elements added.

Declaration

```
public ElementsOrder ElementsOrder { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ElementsOrder | |

## Methods

### AddComponent(ViewElement)

Adds element to container.

Declaration

```
public virtual void AddComponent(ViewElement element)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ViewElement | element | |

### ArrangeOverride(SKSize)

Arranges elements in container. As managed by strategy, expects same size as DesiredSize.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### MeasureOverride(SKSize)

Measure width as sum of widths of all elements in container. Height is equal to maximal height from elements.

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

## ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

# Enum ElementsOrder

Flag for Container to determine order of added elements.

Namespace: GTTG.Model.Strategies

Assembly: cs.temp.dll.dll

Syntax

```
public enum ElementsOrder
```

## Fields

| NAME | DESCRIPTION |
| --- | --- |
| FirstFromLeft | Adds elements from the left. |
| FirstFromRight | Adds elements from the right. |

# Interface IStrategy

Represents contract for strategy implementation used by .

Namespace: GTTG.Model.Strategies

Assembly: cs.temp.dll.dll

Syntax

```
public interface IStrategy : IVisual
```

## Methods

### Clear()

Removes all visuals from strategy.

Declaration

```
void Clear()
```

### Dock()

Rearranges visuals in strategy.

Declaration

```
void Dock()
```

# Class Strategy

Represents strategies applicable to .

Inheritance

System.Object

Strategy

Implements

IStrategy

IVisual

Namespace: GTTG.Model.Strategies

Assembly: cs.temp.dll.dll

Syntax

```
public class Strategy : Visual, IStrategy, IVisual
```

## Constructors

### Strategy(IStrategyDocker, IStrategyDocker, StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Track>, MeasureableSegment>, StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Station>, MeasureableSegment>)

Creates instance with strategies applicable to particular train by using it's .

Declaration

```
public Strategy(IStrategyDocker trackStrategyDocker, IStrategyDocker stationStrategyDocker,
StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Track>, MeasureableSegment>
trackStrategyManager, StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Station>,
MeasureableSegment> stationStrategyManager)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---|---|---|
| IStrategyDocker | trackStrategyDocker | Docker for strategy to place elements in segments above or below horizontal line of tracks. |
| IStrategyDocker | stationStrategyDocker | Docker for strategy to place elements in segments between stations. |
| StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Track>, MeasureableSegment> | trackStrategyManager | Manager to which elements are added to be placed below or above horizontal line of tracks in angles intersecting train's path. |
| StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Station>, MeasureableSegment> | stationStrategyManager | Manager to which elements are added to be placed in segments between stations on train's path. |

### Strategy(ITrainPath, ISegmentRegistry<SegmentType<Track>, MeasureableSegment>, ISegmentRegistry<SegmentType<Station>, MeasureableSegment>)

Creates instance with strategies applicable to particular train by using it's .

Declaration

```
public Strategy(ITrainPath trainPath, ISegmentRegistry<SegmentType<Track>, MeasureableSegment>
trackSegmentRegistry, ISegmentRegistry<SegmentType<Station>, MeasureableSegment> stationSegmentRegistry)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---|---|---|
| ITrainPath | trainPath | Path of train to which strategies are applied. |
| ISegmentRegistry<SegmentType<Track>, MeasureableSegment> | trackSegmentRegistry | TrackSegments above or below horizontal line of tracks. |
| ISegmentRegistry<SegmentType<Station>, MeasureableSegment> | stationSegmentRegistry | TrackSegments between stations. |

## Properties

### StationStrategyDocker

Docker for strategy to place elements in segments between stations.

Declaration

```
protected IStrategyDocker StationStrategyDocker { get; }
```

Property Value

| TYPE | DESCRIPTION |
|---|---|
| IStrategyDocker | |

### StationStrategyManager

Manager to which elements are added to be placed in segments between stations on train's path.

Declaration

```
public StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Station>, MeasureableSegment>
StationStrategyManager { get; }
```

Property Value

| TYPE | DESCRIPTION |
|---|---|
| StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Station>, MeasureableSegment> | |

### TrackStrategyDocker

Docker for strategy to place elements in segments above or below horizontal line of tracks.

Declaration

```
protected IStrategyDocker TrackStrategyDocker { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| IStrategyDocker | |

### TrackStrategyManager

Manager to which elements are added to be placed below or above horizontal line of tracks in angles intersecting train's path.

Declaration

```
public StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Track>, MeasureableSegment>
TrackStrategyManager { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| StrategyManager<TrainEventPlacement, ViewElement, SegmentType<Track>, MeasureableSegment> | |

## Methods

### Clear()

Removes all visuals from strategy.

Declaration

```
public virtual void Clear()
```

### Dock()

Rearranges visuals in strategy.

Declaration

```
public virtual void Dock()
```

### HasHit(SKPoint)

Declaration

```
public override bool HasHit(SKPoint contentPoint)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | contentPoint | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

## ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

## Implements

IStrategy
IVisual

# Namespace GTTG.Model.Strategies.Converters

Classes

TrainEventPlacementConverter

Converts to and provides vector from TrainEvent point to neighbor TrainEvent in provided train path.

# Class TrainEventPlacementConverter

Converts to and provides vector from TrainEvent point to neighbor TrainEvent in provided train path.

Inheritance

System.Object
TrainEventPlacementConverter

Implements

ITypeConverter<TrainEventPlacement, SegmentType<Station>>

Namespace: GTTG.Model.Strategies.Converters

Assembly: cs.temp.dll.dll

Syntax

```
public class TrainEventPlacementConverter : ITypeConverter<TrainEventPlacement, SegmentType<Track>>,
ITypeConverter<TrainEventPlacement, SegmentType<Station>>
```

## Constructors

### TrainEventPlacementConverter(ITrainPath)

Creates convertor for provided train path.

Declaration

```
public TrainEventPlacementConverter(ITrainPath trainPath)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ITrainPath | trainPath | |

## Methods

### ComputeVectorFromEvent(TrainEvent)

Returns vector from point representing event to neighbor event in path direction depending on event type. If first or last event in schedule provided, returns vector in opposite direction multiplied by (-1,-1).

Declaration

```
public (SKPoint SegmentBase, SKPoint VectorFromBase)ComputeVectorFromEvent(TrainEvent trainEvent)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TrainEvent | trainEvent | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.ValueTuple<SKPoint, SKPoint> | SegmentBase -- point in path `trainEvent` that represents `trainEvent`. VectorFromBase -- vector in direction to other point depending on TrainEventType. For departure and passage picks points of next events in schedule. For arrival picks points of previous events to schedule. |

Exceptions

| TYPE | CONDITION |
| --- | --- |
| System.ArgumentException | If `trainEvent` conversion can't be determined. |

## Implements

ITypeConverter<, >

# Namespace GTTG.Model.Strategies.Dockers

Classes

StationStrategyDocker<TElement>

Docks elements into segments of type. Element is placed on middle of line segment of intersection of train path and particular segment.

TracksStrategyDocker<TElement>

Docks elements into segments of type. Element is placed on horizontal line of segment depending on it's type nearby train path.

# Class StationStrategyDocker<TElement>

Docks elements into segments of type. Element is placed on middle of line segment of intersection of train path and particular segment.

Inheritance

System.Object

StationStrategyDocker<TElement>

Namespace: GTTG.Model.Strategies.Dockers

Assembly: cs.temp.dll.dll

Syntax

```
public class StationStrategyDocker<TElement> : IStrategyDocker where TElement : ViewElement
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TElement | |

## Constructors

### StationStrategyDocker(ITrainPath, TrainEventPlacementConverter, StrategyManager<TrainEventPlacement, TElement, SegmentType<Station>, MeasureableSegment>)

Creates docker for particular train path.

Declaration

```
public StationStrategyDocker(ITrainPath trainPath, TrainEventPlacementConverter trainEventPlacementConverter,
StrategyManager<TrainEventPlacement, TElement, SegmentType<Station>, MeasureableSegment> strategyManager)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ITrainPath | trainPath | Train path on which elements are positioned. |
| TrainEventPlacementConverter | trainEventPlacementConverter | Converter from which segments are received. |
| StrategyManager<TrainEventPlacement, TElement, SegmentType<Station>, MeasureableSegment> | strategyManager | Manager with elements to position. |

## Methods

### Dock()

Declaration

```
public void Dock()
```

### DockLowerAcute(TElement, SKPoint, SKPoint)

Docks element in upper segment to acute angle.

Declaration

```
protected void DockLowerAcute(TElement element, SKPoint lowerBoundOrigin, SKPoint vectorToUpperBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | lowerBoundOrigin | Point on lower horizontal line of segment. |
| SKPoint | vectorToUpperBound | Train path segment from `lowerBoundOrigin` to upper horizontal line of segment. |

### DockLowerObtuse(TElement, SKPoint, SKPoint)

Docks element in upper segment to acute angle.

Declaration

```
protected void DockLowerObtuse(TElement element, SKPoint upperBoundOrigin, SKPoint vectorToLowerBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | upperBoundOrigin | Intersection of upper horizontal line of segment and train path. |
| SKPoint | vectorToLowerBound | Train path segment from `upperBoundOrigin` to lower horizontal line of segment. |

### DockUpperAcute(TElement, SKPoint, SKPoint)

Docks element in lower segment to acute angle.

Declaration

```
protected void DockUpperAcute(TElement element, SKPoint upperBoundOrigin, SKPoint vectorToLowerBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | upperBoundOrigin | Intersection of upper horizontal line of segment and train path. |

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | vectorToLowerBound | Train path segment from `upperBoundOrigin` to lower horizontal line of segment. |

### DockUpperObtuse(TElement, SKPoint, SKPoint)

Docks element in lower segment to acute angle.

Declaration

```
protected void DockUpperObtuse(TElement element, SKPoint upperBoundOrigin, SKPoint vectorToLowerBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | upperBoundOrigin | Intersection of upper horizontal line of segment and train path. |
| SKPoint | vectorToLowerBound | Train path segment from `upperBoundOrigin` to lower horizontal line of segment. |

### MeasureHeight(TrainEventPlacement, TElement, SegmentType<Station>, ISegment)

Measures height of element after being positioned by strategy as elements could be rotated.

Declaration

```
public float MeasureHeight(TrainEventPlacement placementPlacement, TElement element, SegmentType<Station>
segmentType, ISegment segment)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TrainEventPlacement | placementPlacement | |
| TElement | element | |
| SegmentType<Station> | segmentType | |
| ISegment | segment | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | |

### ScaleToFitSegment(TElement, ISegment, SKPoint)

Arranges element with it's desired size and scales if does not match segment height.

Declaration

```
protected virtual void ScaleToFitSegment(TElement element, ISegment segment, SKPoint segmentVector)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to measure and scale. |
| ISegment | segment | Segment where element is placed. |
| SKPoint | segmentVector | Line segment of train path intersection the segment. |

# Class TracksStrategyDocker<TElement>

Docks elements into segments of type. Element is placed on horizontal line of segment depending on it's type nearby train path.

Inheritance

System.Object

TracksStrategyDocker<TElement>

Namespace: GTTG.Model.Strategies.Dockers

Assembly: cs.temp.dll.dll

Syntax

```
public class TracksStrategyDocker<TElement> : IStrategyDocker where TElement : ViewElement
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TElement | |

## Constructors

### TracksStrategyDocker(ITrainPath, TrainEventPlacementConverter, StrategyManager<TrainEventPlacement, TElement, SegmentType<Track>, MeasureableSegment>)

Creates docker for particular train path.

Declaration

```
public TracksStrategyDocker(ITrainPath trainPath, TrainEventPlacementConverter trainEventPlacementConverter,
StrategyManager<TrainEventPlacement, TElement, SegmentType<Track>, MeasureableSegment> strategyManager)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ITrainPath | trainPath | Train path nearby which elements are positioned. |
| TrainEventPlacementConverter | trainEventPlacementConverter | Converter from which segments are received. |
| StrategyManager<TrainEventPlacement, TElement, SegmentType<Track>, MeasureableSegment> | strategyManager | Manager with elements to position. |

## Methods

### Dock()

Declaration

```
public void Dock()
```

### DockLowerAcute(TElement, SKPoint, SKPoint)

Docks element in upper segment to acute angle.

Declaration

```
protected virtual void DockLowerAcute(TElement element, SKPoint lowerBoundOrigin, SKPoint vectorToUpperBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | lowerBoundOrigin | Point on lower horizontal line of segment. |
| SKPoint | vectorToUpperBound | Train path segment from `lowerBoundOrigin` to upper horizontal line of segment. |

### DockLowerObtuse(TElement, SKPoint, SKPoint)

Docks element in upper segment to acute angle.

Declaration

```
protected virtual void DockLowerObtuse(TElement element, SKPoint lowerBoundOrigin, SKPoint vectorToUpperBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | lowerBoundOrigin | Point on lower horizontal line of segment. |
| SKPoint | vectorToUpperBound | Train path segment from `lowerBoundOrigin` to upper horizontal line of segment. |

### DockUpperAcute(TElement, SKPoint, SKPoint)

Docks element in lower segment to acute angle.

Declaration

```
protected virtual void DockUpperAcute(TElement element, SKPoint upperBoundOrigin, SKPoint vectorToLowerBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | upperBoundOrigin | Intersection of upper horizontal line of segment and train path. |

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | vectorToLowerBound | Train path segment from `upperBoundOrigin` to lower horizontal line of segment. |

### DockUpperObtuse(TElement, SKPoint, SKPoint)

Docks element in lower segment to acute angle.

Declaration

```
protected virtual void DockUpperObtuse(TElement element, SKPoint upperBoundOrigin, SKPoint vectorToLowerBound)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to dock. |
| SKPoint | upperBoundOrigin | Intersection of upper horizontal line of segment and train path. |
| SKPoint | vectorToLowerBound | Train path segment from `upperBoundOrigin` to lower horizontal line of segment. |

### MeasureHeight(TrainEventPlacement, TElement, SegmentType<Track>, ISegment)

Measures height of element after being positioned by strategy. Returns height of element from arrange.

Declaration

```
public float MeasureHeight(TrainEventPlacement placementPlacement, TElement element, SegmentType<Track>
segmentType, ISegment segment)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TrainEventPlacement | placementPlacement | |
| TElement | element | |
| SegmentType<Track> | segmentType | |
| ISegment | segment | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | |

### ResizeContainerToFitSegment(TElement, ISegment)

Arranges element with it's desired size and scales if does not match segment height.

Declaration

```
protected virtual void ResizeContainerToFitSegment(TElement element, ISegment segment)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TElement | element | Element to measure and scale. |
| ISegment | segment | Segment where element is placed. |

# Namespace GTTG.Model.Strategies.Types

Structs

LineType

Represents horizontal line of track.

SegmentType<T>

Determines placement by vertical position above or below some object.

TrainEventPlacement

Determines placement by event which is represented as point on horizontal line through which passes intersecting line. Determines placement more accurately by selecting angle of formed line intersection.

Enums

AnglePlacement

Selection of angle of line intersecting horizontal line where element should be placed.

SegmentPlacement

Determines vertical placement of segment above or below horizontal line.

# Enum AnglePlacement

Selection of angle of line intersecting horizontal line where element should be placed.

Namespace: GTTG.Model.Strategies.Types

Assembly: cs.temp.dll.dll

Syntax

```
public enum AnglePlacement
```

## Fields

| NAME | DESCRIPTION |
| --- | --- |
| Acute | Element is placed in acute angle. |
| Obtuse | Element is placed in obtuse angle. |

# Struct LineType

Represents horizontal line of track.

Implements

System.IEquatable<LineType>

Inherited Members

System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()

Namespace: GTTG.Model.Strategies.Types

Assembly: cs.temp.dll.dll

Syntax

```
public struct LineType : IEquatable<LineType>
```

## Properties

### Track

Track to which horizontal line belongs.

Declaration

```
public Track Track { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Track | |

## Methods

### Equals(LineType)

Declaration

```
public bool Equals(LineType other)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| LineType | other | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Object | obj | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

Overrides

System.ValueType.Equals(System.Object)

### GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Int32 | |

Overrides

System.ValueType.GetHashCode()

### Of(Track)

Creates line type of particular track.

Declaration

```
public static LineType Of(Track track)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Track | track | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| LineType | |

### Implements

System.IEquatable<T>

# Enum SegmentPlacement

Determines vertical placement of segment above or below horizontal line.

Namespace: GTTG.Model.Strategies.Types

Assembly: cs.temp.dll.dll

Syntax

```
public enum SegmentPlacement
```

## Fields

| NAME | DESCRIPTION |
| --- | --- |
| Lower | Segment is placed below horizontal line. |
| Upper | Segment is placed above horizontal line. |

# Struct SegmentType<T>

Determines placement by vertical position above or below some object.

Implements

System.IEquatable<SegmentType<T>>

Inherited Members

System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()

Namespace: GTTG.Model.Strategies.Types

Assembly: cs.temp.dll.dll

Syntax

```
public struct SegmentType<T> : IEquatable<SegmentType<T>>
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| T | |

## Constructors

### SegmentType(T, SegmentPlacement)

Creates placement determined by vertical position.

Declaration

```
public SegmentType(T type, SegmentPlacement segmentPlacement)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| T | type | Type of object above or below is placement determined |
| SegmentPlacement | segmentPlacement | Type of vertical placement |

## Properties

### SegmentPlacement

Vertical position of placement above or below some object.

Declaration

```
public SegmentPlacement SegmentPlacement { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SegmentPlacement | |

### Type

Instance of object above or below which is placement determined.

Declaration

```
public T Type { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| T | |

## Methods

### Equals(SegmentType<T>)

Declaration

```
public bool Equals(SegmentType<T> other)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SegmentType<T> | other | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Object | obj | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

Overrides

System.ValueType.Equals(System.Object)

### GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Int32 | |

Overrides

System.ValueType.GetHashCode()

## Implements

System.IEquatable<T>

# Struct TrainEventPlacement

Determines placement by event which is represented as point on horizontal line through which passes intersecting line. Determines placement more accurately by selecting angle of formed line intersection.

Implements

System.IEquatable<TrainEventPlacement>

Inherited Members

System.ValueType.ToString()
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetType()

Namespace: GTTG.Model.Strategies.Types

Assembly: cs.temp.dll.dll

Syntax

```
public struct TrainEventPlacement : IEquatable<TrainEventPlacement>
```

## Constructors

### TrainEventPlacement(TrainEvent, AnglePlacement)

Creates placement determined by event and angle.

Declaration

```
public TrainEventPlacement(TrainEvent trainEvent, AnglePlacement anglePlacement)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TrainEvent | trainEvent | Placement to train event. |
| AnglePlacement | anglePlacement | Placement to angle. |

## Properties

### AnglePlacement

Placement to angle.

Declaration

```
public AnglePlacement AnglePlacement { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| AnglePlacement | |

### TrainEvent

Placement to train event represented as point.

Declaration

```
public TrainEvent TrainEvent { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| TrainEvent | |

## Methods

### Equals(TrainEventPlacement)

Declaration

```
public bool Equals(TrainEventPlacement other)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TrainEventPlacement | other | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| System.Object | obj | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

Overrides

System.ValueType.Equals(System.Object)

### GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Int32 | |

Overrides

System.ValueType.GetHashCode()

## Implements

System.IEquatable<T>

# Namespace GTTG.Model.ViewModel.Infrastructure

Classes

InfrastructureViewElement

Represents view element of infrastructure with infinite width.

# Class InfrastructureViewElement

Represents view element of infrastructure with infinite width.

Inheritance

System.Object
InfrastructureViewElement
RailwayView<TStationView, TTrackView>
StationView<TTrackView>
TrackView

Namespace: GTTG.Model.ViewModel.Infrastructure

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class InfrastructureViewElement : ViewElement
```

## Constructors

### InfrastructureViewElement()

Creates infrastructure view element with infinite width.

Declaration

```
protected InfrastructureViewElement()
```

# Namespace GTTG.Model.ViewModel.Infrastructure.Railways

Classes

RailwayView<TStationView, TTrackView>

Represents visualization of Railway.

StrategyRailwayView<TStationView, TTrackView>

Represents visualization of with segments to which elements in strategies can be placed.

Interfaces

IRailwayViewFactory<TRailwayView, TStationView, TTrackView>

Factory for railway view classes deriving from RailwayView<TStationView, TTrackView>.

# Interface IRailwayViewFactory<TRailwayView, TStationView, TTrackView>

Factory for railway view classes deriving from RailwayView<TStationView, TTrackView>.

Namespace: GTTG.Model.ViewModel.Infrastructure.Railways

Assembly: cs.temp.dll.dll

Syntax

```
public interface IRailwayViewFactory<out TRailwayView, TStationView, TTrackView>
    where TRailwayView : RailwayView<TStationView, TTrackView> where TStationView : StationView<TTrackView>
where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TRailwayView | Implementation of railway view. |
| TStationView | Implementation of railway view contains stations views of `TStationView`. |
| TTrackView | `TStationView` contains track views of `TTrackView`. |

## Methods

### CreateRailwayView(Railway)

Creates specific implementation of railway view from railway instance.

Declaration

```
TRailwayView CreateRailwayView(Railway railway)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Railway | railway | instance which is backed by this view. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TRailwayView | Implementation of railway view derived from RailwayView<TStationView, TTrackView>. |

# Class RailwayView<TStationView, TTrackView>

Represents visualization of Railway.

Inheritance

System.Object
InfrastructureViewElement
RailwayView<TStationView, TTrackView>
StrategyRailwayView<TStationView, TTrackView>

Namespace: GTTG.Model.ViewModel.Infrastructure.Railways

Assembly: cs.temp.dll.dll

Syntax

```
public class RailwayView<TStationView, TTrackView> : InfrastructureViewElement where TStationView :
StationView<TTrackView> where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TStationView | Concrete implementation of used by this instance. |
| TTrackView | Concrete implementation of used by this instance. |

## Constructors

### RailwayView(Railway, IStationViewFactory<TStationView, TTrackView>)

Creates visualization of Railway.

Declaration

```
public RailwayView(Railway railway, IStationViewFactory<TStationView, TTrackView> stationViewFactory)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Railway | railway | Instance of Railway to be visualized. |
| IStationViewFactory<TStationView, TTrackView> | stationViewFactory | Interface with factory method to convert list of instances in Railway to StationViews. |

## Properties

### Railway

Instance of Railway being visualized.

Declaration

```
public Railway Railway { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Railway | |

### StationViews

Visualization of stations in Railway.

Declaration

```
public ImmutableList<TStationView> StationViews { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableList<TStationView> | |

## Methods

### ArrangeOverride(SKSize)

Arranges StationViews proportionally in `finalSize` height. If height returned from MeasureOverride(SKSize) is higher than `finalSize`, stations receives in ArrangeOverride(SKSize) scaled desired height. Otherwise remaining space is split equally between stations.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### MeasureOverride(SKSize)

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

## OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

## ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

# Class StrategyRailwayView<TStationView, TTrackView>

Represents visualization of with segments to which elements in strategies can be placed.

Inheritance

System.Object
InfrastructureViewElement
RailwayView<TStationView, TTrackView>
StrategyRailwayView<TStationView, TTrackView>

Inherited Members

RailwayView<TStationView, TTrackView>.Railway
RailwayView<TStationView, TTrackView>.StationViews
RailwayView<TStationView, TTrackView>.OnDraw(DrawingCanvas)
RailwayView<TStationView, TTrackView>.ProvideVisuals()
RailwayView<TStationView, TTrackView>.MeasureOverride(SKSize)
RailwayView<TStationView, TTrackView>.ArrangeOverride(SKSize)

Namespace: GTTG.Model.ViewModel.Infrastructure.Railways

Assembly: cs.temp.dll.dll

Syntax

```
public class StrategyRailwayView<TStationView, TTrackView> : RailwayView<TStationView, TTrackView> where
TStationView : StationView<TTrackView> where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TStationView | Concrete implementation of used by this instance. |
| TTrackView | Concrete implementation of used by this instance. |

## Constructors

### StrategyRailwayView(Railway, IStationViewFactory<TStationView, TTrackView>, ISegmentRegistry<SegmentType<Station>, MeasureableSegment>)

Creates visualization of and segments placed to `stationSegments`.

Declaration

```
public StrategyRailwayView(Railway railway, IStationViewFactory<TStationView, TTrackView> stationViewFactory,
ISegmentRegistry<SegmentType<Station>, MeasureableSegment> stationSegments)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Railway | railway | Instance of to be visualized. |
| IStationViewFactory<TStationView, TTrackView> | stationViewFactory | Interface with factory method to convert list of instances in to StationViews. |

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ISegmentRegistry<SegmentType<Station>, MeasureableSegment> | stationSegments | Registry where segments are registered. |

## Fields

### StationSegments

Segment registry where created segments above and below station were registered.

Declaration

```
protected ISegmentRegistry<SegmentType<Station>, MeasureableSegment> StationSegments
```

Field Value

| TYPE | DESCRIPTION |
| --- | --- |
| ISegmentRegistry<SegmentType<Station>, MeasureableSegment> | |

## Methods

### ArrangeOverride(SKSize)

Arranges StationViews and segments proportionally in `finalSize` height. If height returned from MeasureOverride(SKSize) is higher than `finalSize`, stations and segments receives in ArrangeOverride(SKSize) scaled desired height. Otherwise remaining space is split equally between stations.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

Overrides

GTTG.Model.ViewModel.Infrastructure.Railways.RailwayView<TStationView, TTrackView>.ArrangeOverride(SKSize)

### MeasureOverride(SKSize)

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

Overrides

GTTG.Model.ViewModel.Infrastructure.Railways.RailwayView<TStationView, TTrackView>.MeasureOverride(SKSize)

## OnSegmentRegistration(ISegmentRegistry<SegmentType<Station>, MeasureableSegment>)

Registers segments to StationSegments.

Declaration

```
protected virtual void OnSegmentRegistration(ISegmentRegistry<SegmentType<Station>, MeasureableSegment>
segmentRegistry)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ISegmentRegistry<SegmentType<Station>, MeasureableSegment> | segmentRegistry | |

# Namespace GTTG.Model.ViewModel.Infrastructure.Stations

Classes

StationView<TTrackView>

Represents visualization of StationView<TTrackView>.

StrategyStationView<TTrackView>

Represents visualization of with segments to which elements in strategies can be placed.

Interfaces

IStationViewFactory<TStationView, TTrackView>

Factory for station view classes.

# Interface IStationViewFactory<TStationView, TTrackView>

Factory for station view classes.

Namespace: GTTG.Model.ViewModel.Infrastructure.Stations

Assembly: cs.temp.dll.dll

Syntax

```
public interface IStationViewFactory<out TStationView, TTrackView>
    where TStationView : StationView<TTrackView> where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TStationView | Station view class deriving from StationView<TTrackView>. |
| TTrackView | `TStationView` contains track views of `TTrackView`. |

## Methods

### CreateStationView(Station)

Creates specific implementation of station view from station instance.

Declaration

```
TStationView CreateStationView(Station station)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Station | station | instance visualized by this view. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TStationView | Implementation of station view derived from StationView<TTrackView>. |

# Class StationView<TTrackView>

Represents visualization of StationView<TTrackView>.

Inheritance

System.Object
InfrastructureViewElement
StationView<TTrackView>
StrategyStationView<TTrackView>

Namespace: GTTG.Model.ViewModel.Infrastructure.Stations

Assembly: cs.temp.dll.dll

Syntax

```
public class StationView<TTrackView> : InfrastructureViewElement where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrackView | Concrete implementation of used by this instance. |

## Constructors

### StationView(Station, ITrackViewFactory<TTrackView>)

Creates visualization of Station.

Declaration

```
public StationView(Station station, ITrackViewFactory<TTrackView> trackViewFactory)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Station | station | Instance of Station to be visualized. |
| ITrackViewFactory<TTrackView> | trackViewFactory | Interface with factory method to convert list of instances in Station to TrackViews. |

## Properties

### Station

Instance of Station being visualized.

Declaration

```
public Station Station { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Station | |

### TrackViews

Visualization of tracks in Station.

Declaration

```
public ImmutableList<TTrackView> TrackViews { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableList<TTrackView> | |

## Methods

### ArrangeOverride(SKSize)

Arranges TrackViews proportionally in `finalSize` height. If height returned from MeasureOverride(SKSize) is higher than `finalSize`, tracks receives in ArrangeOverride(SKSize) scaled desired height. Otherwise remaining space is split equally between tracks.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### MeasureOverride(SKSize)

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

## ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

# Class StrategyStationView<TTrackView>

Represents visualization of with segments to which elements in strategies can be placed.

Inheritance

System.Object
InfrastructureViewElement
StationView<TTrackView>
StrategyStationView<TTrackView>

Inherited Members

StationView<TTrackView>.Station
StationView<TTrackView>.TrackViews
StationView<TTrackView>.OnDraw(DrawingCanvas)
StationView<TTrackView>.ProvideVisuals()

Namespace: GTTG.Model.ViewModel.Infrastructure.Stations

Assembly: cs.temp.dll.dll

Syntax

```
public class StrategyStationView<TTrackView> : StationView<TTrackView> where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrackView | Concrete implementation of used by this instance. |

## Constructors

### StrategyStationView(Station, ISegmentRegistry<SegmentType<Track>, MeasureableSegment>, ITrackViewFactory<TTrackView>)

Creates visualization of and segments placed to `trackSegments`.

Declaration

```
public StrategyStationView(Station station, ISegmentRegistry<SegmentType<Track>, MeasureableSegment>
trackSegments, ITrackViewFactory<TTrackView> trackViewFactory)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Station | station | Instance of to be visualized. |
| ISegmentRegistry<SegmentType<Track>, MeasureableSegment> | trackSegments | Registry where segments are registered. |
| ITrackViewFactory<TTrackView> | trackViewFactory | Interface with factory method to convert list of instances in to StationView<TTrackView>. |

## Properties

### TrackSegments

Segment registry where created segments above and below tracks were registered.

Declaration

```
protected ISegmentRegistry<SegmentType<Track>, MeasureableSegment> TrackSegments { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ISegmentRegistry<SegmentType<Track>, MeasureableSegment> | |

## Methods

### ArrangeOverride(SKSize)

Arranges StationView<TTrackView> and segments proportionally in `finalSize` height. If height returned from MeasureOverride(SKSize) is higher than `finalSize`, tracks and segments receives in ArrangeOverride(SKSize) scaled desired height. Otherwise remaining space is split equally between tracks.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

Overrides

GTTG.Model.ViewModel.Infrastructure.Stations.StationView<TTrackView>.ArrangeOverride(SKSize)

### MeasureOverride(SKSize)

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

Overrides

GTTG.Model.ViewModel.Infrastructure.Stations.StationView<TTrackView>.MeasureOverride(SKSize)

## RegisterStationSegments(ISegmentRegistry<SegmentType<Track>, MeasureableSegment>)

Registers segments to TrackSegments.

### Declaration

```
protected void RegisterStationSegments(ISegmentRegistry<SegmentType<Track>, MeasureableSegment>
segmentRegistry)
```

### Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| ISegmentRegistry<SegmentType<Track>, MeasureableSegment> | segmentRegistry | |

# Namespace GTTG.Model.ViewModel.Infrastructure.Tracks

Classes

TrackView

Represents visualization of Track.

Interfaces

ITrackViewFactory<TTrackView>

Factory for creating visualizations of instances.

# Interface ITrackViewFactory<TTrackView>

Factory for creating visualizations of instances.

Namespace: GTTG.Model.ViewModel.Infrastructure.Tracks

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITrackViewFactory<out TTrackView>
    where TTrackView : TrackView
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrackView | Visualization of tracks deriving from TrackView. |

## Methods

### CreateTrackView(Track)

Creates `TTrackView` instance from `track`

Declaration

```
TTrackView CreateTrackView(Track track)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Track | track | to visualize. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TTrackView | Instance of `TTrackView` as visualization of `track` |

# Class TrackView

Represents visualization of Track.

Inheritance

System.Object
InfrastructureViewElement
TrackView

Namespace: GTTG.Model.ViewModel.Infrastructure.Tracks

Assembly: cs.temp.dll.dll

Syntax

```
public class TrackView : InfrastructureViewElement
```

## Constructors

### TrackView(Track, LinePaint, MeasureableSegment)

Creates visualization of `track`.

Declaration

```
public TrackView(Track track, LinePaint linePaint, MeasureableSegment trackLineSegment)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| Track | track | Instance of track being visualized. |
| LinePaint | linePaint | Paint to use when drawing representing horizontal line of track. |
| MeasureableSegment | trackLineSegment | Segment used to determine and position of horizontal line. |

## Properties

### LinePaint

Paint for TrackPath.

Declaration

```
protected LinePaint LinePaint { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| LinePaint | |

### Track

Instance of track being visualized.

Declaration

```
public Track Track { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| Track | |

### TrackLineSegment

Segment to determine height ( of TrackPath.

Declaration

```
protected MeasureableSegment TrackLineSegment { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| MeasureableSegment | |

### TrackPath

Horizontal line representing track.

Declaration

```
protected SKPath TrackPath { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| SKPath | |

## Methods

### ArrangeOverride(SKSize)

Set height of horizontal line.

Declaration

```
protected override SKSize ArrangeOverride(SKSize finalSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | finalSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### MeasureOverride(SKSize)

Declaration

```
protected override SKSize MeasureOverride(SKSize availableSize)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKSize | availableSize | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| SKSize | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

### ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

# Namespace GTTG.Model.ViewModel.Traffic

Classes

StrategyTrainView<TStrategy, TTrain>

Represents visualization of with strategies which positions added elements nearby train path.

TrafficView<TTrainView, TTrain>

Represents visualization of .

TrainView<TTrain>

Represents visualization of train.

Interfaces

ITrafficViewFactory<TTrafficView, TTrainView, TTrain>

Factory for traffic view classes.

ITrainViewFactory<TTrainView, TTrain>

Factory for train view classes.

# Interface ITrafficViewFactory<TTrafficView, TTrainView, TTrain>

Factory for traffic view classes.

Namespace: GTTG.Model.ViewModel.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITrafficViewFactory<out TTrafficView, TTrainView, TTrain>
    where TTrafficView : TrafficView<TTrainView, TTrain> where TTrainView : TrainView<TTrain> where TTrain :
Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrafficView | Traffic view class deriving from TrafficView<TTrainView, TTrain>. |
| TTrainView | Train view class deriving from TrainView<TTrain>. |
| TTrain | Train class deriving from . |

## Methods

### CreateTrafficView(Traffic<TTrain>)

Creates specific implementation of traffic view from traffic instance.

Declaration

```
TTrafficView CreateTrafficView(Traffic<TTrain> traffic)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| GTTG.Model.ViewModel.Traffic<TTrain> | traffic | instance visualized by this view. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TTrafficView | |

# Interface ITrainViewFactory<TTrainView, TTrain>

Factory for train view classes.

Namespace: GTTG.Model.ViewModel.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public interface ITrainViewFactory<out TTrainView, in TTrain>
    where TTrainView : TrainView<TTrain> where TTrain : Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrainView | Train view class deriving from TrainView<TTrain>. |
| TTrain | Train class deriving from . |

## Methods

### CreateTrainView(TTrain)

Creates specific implementation of train view from train instance.

Declaration

```
TTrainView CreateTrainView(TTrain train)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TTrain | train | instance which is visualized by . |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TTrainView | |

# Class StrategyTrainView<TStrategy, TTrain>

Represents visualization of with strategies which positions added elements nearby train path.

Inheritance

System.Object
TrainView<TTrain>
StrategyTrainView<TStrategy, TTrain>

Inherited Members

TrainView<TTrain>.Train
TrainView<TTrain>.TrainPath
TrainView<TTrain>.DistanceFromPoint(SKPoint)

Namespace: GTTG.Model.ViewModel.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public class StrategyTrainView<TStrategy, TTrain> : TrainView<TTrain> where TStrategy : IStrategy where TTrain
: Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TStrategy | Concrete implementation of used by this instance. |
| TTrain | Concrete implementation of used by this instance. |

## Constructors

### StrategyTrainView(TTrain, ITrainPath, TStrategy)

Creates visualization of with provided `strategy`.

Declaration

```
public StrategyTrainView(TTrain train, ITrainPath trainPath, TStrategy strategy)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TTrain | train | Instance of to be visualized. |
| ITrainPath | trainPath | Train path representing schedule of train. |
| TStrategy | strategy | Implementation of Strategy. |

## Properties

### Strategy

Instance of being used to add elements nearby train path.

Declaration

```
public TStrategy Strategy { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| TStrategy | |

## Methods

### Arrange()

Declaration

```
public override void Arrange()
```

Overrides

GTTG.Model.ViewModel.Traffic.TrainView<TTrain>.Arrange()

### DrawContainers(DrawingCanvas)

Draws Strategy.

Declaration

```
protected virtual void DrawContainers(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

### HasHit(SKPoint)

Declaration

```
public override bool HasHit(SKPoint contentPoint)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | contentPoint | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

Overrides

GTTG.Model.ViewModel.Traffic.TrainView<TTrain>.OnDraw(DrawingCanvas)

### ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

### UpdateTrainViewContent()

Declaration

```
public override void UpdateTrainViewContent()
```

Overrides

GTTG.Model.ViewModel.Traffic.TrainView<TTrain>.UpdateTrainViewContent()

# Class TrafficView<TTrainView, TTrain>

Represents visualization of .

Inheritance

System.Object

TrafficView<TTrainView, TTrain>

Namespace: GTTG.Model.ViewModel.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public class TrafficView<TTrainView, TTrain> : Visual where TTrainView : TrainView<TTrain> where TTrain :
Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrainView | |
| TTrain | |

## Constructors

### TrafficView(Traffic<TTrain>, ITrainViewFactory<TTrainView, TTrain>)

Creates visualization of `traffic`.

Declaration

```
public TrafficView(Traffic<TTrain> traffic, ITrainViewFactory<TTrainView, TTrain> trainViewFactory)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| GTTG.Model.ViewModel.Traffic<TTrain> | traffic | Instance of traffic to be visualized. |
| ITrainViewFactory<TTrainView, TTrain> | trainViewFactory | Interface with factory method to convert list of instances in `traffic` to TrainViews. |

## Properties

### Traffic

Instance of traffic being visualized.

Declaration

```
public Traffic<TTrain> Traffic { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| GTTG.Model.ViewModel.Traffic<TTrain> | |

### TrainViews

Visualization of trains in Traffic.

Declaration

```
public ImmutableList<TTrainView> TrainViews { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ImmutableList<TTrainView> | |

## Methods

### Arrange()

Calls Arrange() on all trains TrainViews.

Declaration

```
public void Arrange()
```

### HasHit(SKPoint)

Declaration

```
public override bool HasHit(SKPoint contentPoint)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | contentPoint | |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Boolean | |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

### ProvideVisuals()

Declaration

```
public override IEnumerable<IVisual> ProvideVisuals()
```

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Collections.Generic.IEnumerable<IVisual> | |

### SelectNearestTrainView(SKPoint)

Selects nearest train in TrainViews with closest distance to provided point.

Declaration

```
public TTrainView SelectNearestTrainView(SKPoint canvasLocation)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | canvasLocation | Position on to which find the nearest train. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| TTrainView | Instance of nearest in TrainViews. |

### Update()

Calls UpdateTrainViewContent() on all trains TrainViews.

Declaration

```
public void Update()
```

# Class TrainView<TTrain>

Represents visualization of train.

Inheritance

System.Object
TrainView<TTrain>
StrategyTrainView<TStrategy, TTrain>

Namespace: GTTG.Model.ViewModel.Traffic

Assembly: cs.temp.dll.dll

Syntax

```
public abstract class TrainView<TTrain> : Visual where TTrain : Train
```

Type Parameters

| NAME | DESCRIPTION |
| --- | --- |
| TTrain | |

## Constructors

### TrainView(TTrain, ITrainPath)

Creates visualization of train.

Declaration

```
protected TrainView(TTrain train, ITrainPath trainPath)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| TTrain | train | Instance of train to visualize. |
| ITrainPath | trainPath | Train path representing schedule of train. |

## Properties

### Train

Instance of train being visualized.

Declaration

```
public TTrain Train { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| TTrain | |

### TrainPath

Line representing schedule of train.

Declaration

```
protected ITrainPath TrainPath { get; }
```

Property Value

| TYPE | DESCRIPTION |
| --- | --- |
| ITrainPath | |

## Methods

### Arrange()

Arranges content of train on canvas to reflect changes in arrangement of other view models. Re-arranges train path as line of points in railway.

Declaration

```
public virtual void Arrange()
```

### DistanceFromPoint(SKPoint)

Determines closest distance from point to train path.

Declaration

```
public virtual float DistanceFromPoint(SKPoint point)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| SKPoint | point | Point to which find the distance. |

Returns

| TYPE | DESCRIPTION |
| --- | --- |
| System.Single | Closest distance to `point`. |

### OnDraw(DrawingCanvas)

Declaration

```
protected override void OnDraw(DrawingCanvas drawingCanvas)
```

Parameters

| TYPE | NAME | DESCRIPTION |
| --- | --- | --- |
| DrawingCanvas | drawingCanvas | |

### UpdateTrainViewContent()

Updates content of train visualization to match new schedule of train. Re-validates point in train path as schedule creating train path can be changed.

Declaration

```
public virtual void UpdateTrainViewContent()
```