Derived

Generated by Doxygen 1.9.2

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 Cargo Class Reference

Inherited by **Satellite**.

Public Member Functions

- **Cargo** (string name, double weight, string descp)

  *Construct a new **Cargo** (p. 7) object.*
- string **getName** ()

  Get the Name.
- string **getDescription** ()

  Get the Description.
- double **getWeight** ()

  Get the Weight.
- ~**Cargo** ()

  *Destroy the **Cargo** (p. 7) object.*

4.1.1 Constructor & Destructor Documentation

4.1.1.1 Cargo()

```
Cargo::Cargo (
            string name,
            double weight,
            string descp )
```

Construct a new **Cargo** (p. 7) object.

Author

Derived

Parameters

name	
weight	
descp	

4.1.1.2 ∼Cargo()

```
Cargo::∼Cargo ( )
```

Destroy the **Cargo** (p. 7) object.

Author

Derived

4.1.2 Member Function Documentation

4.1.2.1 getDescription()

```
string Cargo::getDescription ( )
```

Get the Description.

Author

Derived

Returns

string

4.1.2.2 getName()

```
string Cargo::getName ( )
```

Get the Name.

Author

Derived

Returns

string

4.1.2.3 getWeight()

```
double Cargo::getWeight ( )
```

Get the Weight.

Author

Derived

Returns

double

The documentation for this class was generated from the following files:

- Cargo.h
- Cargo.cpp

4.2 CargoCollection Class Reference

Inherited by **VectorOfCargo**.

Public Member Functions

- **CargoCollection** ()=default

  *Construct a new default **Cargo** (*p. *7) Collection object.*
- virtual **CargoIterator** ∗ **createCargoIterator** ()=0

  *Create a **Cargo** (*p. *7) **Iterator** (*p. *61) object.*
- virtual void **addCargo** (**Cargo** ∗)=0

  *Add **Cargo** (*p. *7) to the Collection.*
- virtual void **removeCargo** (**Cargo** ∗)=0

  *Remove **Cargo** (*p. *7) from the Collection.*
- virtual bool **isEmpty** ()=0

  *Check in the **Cargo** (*p. *7) Collection is empty.*
- virtual ∼**CargoCollection** ()=default

  *Destroy the **Cargo** (*p. *7) Collection object.*

4.2.1 Constructor & Destructor Documentation

4.2.1.1 CargoCollection()

```
CargoCollection::CargoCollection ( )  [default]
```

Construct a new default **Cargo** (p. 7) Collection object.

Author

Derived

4.2.1.2 ∼CargoCollection()

```
virtual CargoCollection::∼CargoCollection ( )  [virtual], [default]
```

Destroy the **Cargo** (p. 7) Collection object.

Author

Derived

4.2.2 Member Function Documentation

4.2.2.1 addCargo()

```
virtual void CargoCollection::addCargo (
              Cargo * )  [pure virtual]
```

Add **Cargo** (p. 7) to the Collection.

Author

Derived

Implemented in **VectorOfCargo** (p. 102).

4.2.2.2 createCargoIterator()

```
virtual  CargoIterator * CargoCollection::createCargoIterator ( )  [pure virtual]
```

Create a **Cargo** (p. 7) **Iterator** (p. 61) object.

Author

Derived

Returns

CargoIterator*

Implemented in **VectorOfCargo** (p. 102).

4.2.2.3 isEmpty()

```
virtual bool CargoCollection::isEmpty ( )  [pure virtual]
```

Check in the **Cargo** (p. 7) Collection is empty.

Author

Derived

Returns

true

false

Implemented in **VectorOfCargo** (p. 102).

4.2.2.4 removeCargo()

```
virtual void CargoCollection::removeCargo (
              Cargo *  )  [pure virtual]
```

Remove **Cargo** (p. 7) from the Collection.

Author

Derived

Implemented in **VectorOfCargo** (p. 102).

The documentation for this class was generated from the following file:

- CargoCollection.h

4.3 CargoFactory Class Reference

Public Member Functions

- **CargoFactory** ()

  *Construct a new **Cargo** (p. 7) Factory object.*
- ∼**CargoFactory** ()

  *Destroy the **Cargo** (p. 7) Factory object.*
- **Cargo** ∗ **createCargo** (string name, double weight, string description)

  *Create a **Cargo** (p. 7) object.*

4.3.1 Constructor & Destructor Documentation

4.3.1.1 CargoFactory()

```
CargoFactory::CargoFactory ( )
```

Construct a new **Cargo** (p. 7) Factory object.

Author

Derived

4.3.1.2 ∼CargoFactory()

```
CargoFactory::∼CargoFactory ( )
```

Destroy the **Cargo** (p. 7) Factory object.

Author

Derived

4.3.2 Member Function Documentation

4.3.2.1 createCargo()

```
 Cargo * CargoFactory::createCargo (
            string name,
            double weight,
            string description )
```

Create a **Cargo** (p. 7) object.

Author

Derived

Parameters

name	
weight	
description	

Returns

Cargo∗

The documentation for this class was generated from the following files:

- CargoFactory.h
- CargoFactory.cpp

4.4 CargoIterator Class Reference

Inherited by **OperationsIteratorCargo**.

Public Member Functions

- **CargoIterator** ()=default

  *Construct a new default **Cargo** (p. 7) **Iterator** (p. 61) object.*
- virtual **Cargo** ∗ **first** ()=0

  *Return the first **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual **Cargo** ∗ **next** ()=0

  *Return the next **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual bool **hasNext** ()=0

  *Check if there is **Cargo** (p. 7) in the next **Iterator** (p. 61).*
- virtual **Cargo** ∗ **current** ()=0

  *Return the Current **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual ∼**CargoIterator** ()=default

  *Destroy the **Cargo** (p. 7) **Iterator** (p. 61) object.*

4.4.1 Constructor & Destructor Documentation

4.4.1.1 CargoIterator()

```
CargoIterator::CargoIterator ( )  [default]
```

Construct a new default **Cargo** (p. 7) **Iterator** (p. 61) object.

Author

Derived

4.4.1.2 ∼CargoIterator()

```
virtual CargoIterator::∼CargoIterator ( )  [virtual], [default]
```

Destroy the **Cargo** (p. 7) **Iterator** (p. 61) object.

Author

Derived

4.4.2 Member Function Documentation

4.4.2.1 current()

```
virtual  Cargo ∗ CargoIterator::current ( )  [pure virtual]
```

Return the Current **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implemented in **OperationsIteratorCargo** (p. 65).

4.4.2.2 first()

```
virtual  Cargo ∗ CargoIterator::first ( )  [pure virtual]
```

Return the first **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implemented in **OperationsIteratorCargo** (p. 66).

4.4.2.3 hasNext()

```
virtual bool CargoIterator::hasNext ( )  [pure virtual]
```

Check if there is **Cargo** (p. 7) in the next **Iterator** (p. 61).

Author

Derived

Returns

true

false

Implemented in **OperationsIteratorCargo** (p. 66).

4.4.2.4 next()

```
virtual  Cargo ∗ CargoIterator::next ( )  [pure virtual]
```

Return the next **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implemented in **OperationsIteratorCargo** (p. 66).

The documentation for this class was generated from the following file:

- CargoIterator.h

4.5 Colours Class Reference

Static Public Member Functions

- static string **black** (string s)

  Changes the Colour of the String.
- static string **red** (string s)

  Changes the Colour of the String.
- static string **green** (string s)

  Changes the Colour of the String.
- static string **yellow** (string s)

  Changes the Colour of the String.
- static string **blue** (string s)

  Changes the Colour of the String.
- static string **purple** (string s)

  Changes the Colour of the String.
- static string **cyan** (string s)

  Changes the Colour of the String.

4.5.1 Member Function Documentation

4.5.1.1 black()

```
static string Colours::black (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.2 blue()

```
static string Colours::blue (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.3 cyan()

```
static string Colours::cyan (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.4 green()

```
static string Colours::green (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.5 purple()

```
static string Colours::purple (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.6 red()

```
static string Colours::red (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

4.5.1.7 yellow()

```
static string Colours::yellow (
             string s )  [inline], [static]
```

Changes the Colour of the String.

Author

Derived

Parameters

s	

Returns

string

The documentation for this class was generated from the following file:

- Colours.h

4.6 ConcreteGroundMissionControl Class Reference

Inherits **GroundMissionControl**.

Public Member Functions

- **ConcreteGroundMissionControl** ()

  Construct a new Concrete Ground Mission Control object.
- bool **hasConnected** ()

  Check if the Concrete object is Connected.
- void **setConnection** (bool)

  Set the Connection of the Concrete object.
- ∼**ConcreteGroundMissionControl** ()

  Destroy the Concrete Ground Mission Control object.

4.6.1 Constructor & Destructor Documentation

4.6.1.1 ConcreteGroundMissionControl()

```
ConcreteGroundMissionControl::ConcreteGroundMissionControl ( )
```

Construct a new Concrete Ground Mission Control object.

Author

Derived

4.6.1.2 ∼ConcreteGroundMissionControl()

```
ConcreteGroundMissionControl::∼ConcreteGroundMissionControl ( )
```

Destroy the Concrete Ground Mission Control object.

Author

Derived

4.6.2 Member Function Documentation

4.6.2.1 hasConnected()

```
bool ConcreteGroundMissionControl::hasConnected ( )
```

Check if the Concrete object is Connected.

Author

Derived

Returns

true

false

4.6.2.2 setConnection()

```
void ConcreteGroundMissionControl::setConnection (
            bool setConnection )
```

Set the Connection of the Concrete object.

Author

Derived

The documentation for this class was generated from the following files:

- ConcreteGroundMissionControl.h
- ConcreteGroundMissionControl.cpp

4.7 ConfigurationManager Class Reference

Public Member Functions

- **ConfigurationManager** (**CargoCollection** ∗c, **HumanCollection** ∗h)

  Construct a new Configuration Manager object.
- ∼**ConfigurationManager** ()

  Destroy the Configuration Manager object.
- **Rocket** ∗ **BuildAndDecorateRocket** ()

  Build and Decorate a ***Rocket*** *(*p. *77).*

4.7.1 Constructor & Destructor Documentation

4.7.1.1 ConfigurationManager()

```
ConfigurationManager::ConfigurationManager (
              CargoCollection * c,
              HumanCollection * h )
```

Construct a new Configuration Manager object.

Author

Derived

Parameters

c	
h	

4.7.1.2 ∼ConfigurationManager()

```
ConfigurationManager::∼ConfigurationManager ( )
```

Destroy the Configuration Manager object.

Author

Derived

4.7.2 Member Function Documentation

4.7.2.1 BuildAndDecorateRocket()

```
Rocket * ConfigurationManager::BuildAndDecorateRocket ( )
```

Build and Decorate a **Rocket** (p. 77).

Author

Derived

Returns

Rocket∗

The documentation for this class was generated from the following files:

- ConfigurationManager.h
- ConfigurationManager.cpp

4.8 Core Class Reference

Inherited by **Falcon9Core**.

The documentation for this class was generated from the following file:

- Core.h

4.9 CrewDragon Class Reference

Inherits **SpaceCraft**.

Public Member Functions

- **CrewDragon** (**CargoCollection** ∗c, **HumanCollection** ∗h)

  *Construct a new Crew **Dragon** (*p. *23) object.*
- ∼**CrewDragon** ()

  *Destroy the Crew **Dragon** (*p. *23) object.*
- void **printSpaceCraft** ()

  *Print the Crew **Dragon** (*p. *23) object.*

4.9.1 Constructor & Destructor Documentation

4.9.1.1 CrewDragon()

```
CrewDragon::CrewDragon (
              CargoCollection ∗ c,
              HumanCollection ∗ h )
```

Construct a new Crew **Dragon** (p. 23) object.

Author

Derived

Parameters

c	
h	

4.9.1.2 ∼CrewDragon()

```
CrewDragon::∼CrewDragon ( )
```

Destroy the Crew **Dragon** (p. 23) object.

Author

Derived

4.9.2 Member Function Documentation

4.9.2.1 printSpaceCraft()

```
void CrewDragon::printSpaceCraft ( )  [virtual]
```

Print the Crew **Dragon** (p. 23) object.

Author

Derived

Implements **SpaceCraft** (p. 95).

The documentation for this class was generated from the following files:

- CrewDragon.h
- CrewDragon.cpp

4.10 Dragon Class Reference

Inherits **SpaceCraft**.

Public Member Functions

- **Dragon** (**CargoCollection** ∗c)

  *Construct a new **Dragon** (p. 23) object.*
- ∼**Dragon** ()

  *Destroy the **Dragon** (p. 23) object.*
- void **printSpaceCraft** ()

  *Print the **Dragon** (p. 23) object.*

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Dragon()

```
Dragon::Dragon (
              CargoCollection * c )
```

Construct a new **Dragon** (p. 23) object.

Construct a new **Dragon** (p. 23):: **Dragon** (p. 23) object.

Author

Derived

Parameters

c	

4.10.1.2 ∼Dragon()

```
Dragon::∼Dragon ( )
```

Destroy the **Dragon** (p. 23) object.

Destroy the **Dragon** (p. 23):: **Dragon** (p. 23) object.

Author

Derived

4.10.2 Member Function Documentation

4.10.2.1 printSpaceCraft()

```
void Dragon::printSpaceCraft ( )  [virtual]
```

Print the **Dragon** (p. 23) object.

Author

Derived

Implements **SpaceCraft** (p. 95).

The documentation for this class was generated from the following files:

- Dragon.h
- Dragon.cpp

4.11 Engine Class Reference

Inherited by **MerlinEngine**, and **VacuumMerlinEngine**.

Public Member Functions

- **Engine** ()

  *Construct a new **Engine** (p. 25) object.*

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Engine()

```
Engine::Engine ( )  [inline]
```

Construct a new **Engine** (p. 25) object.

Author

Derived

The documentation for this class was generated from the following file:

- Engine.h

4.12 F9Stage1 Class Reference

Inherits **RocketState**.

Public Member Functions

- void **handleChange** (**Rocket** ∗r)

  *Handle Change of **F9Stage1** (p. 25) object.*
- **F9Stage1** ()

  *Construct a new **F9Stage1** (p. 25) object.*
- ∼**F9Stage1** ()

  *Destroy the **F9Stage1** (p. 25) object.*

4.12.1 Constructor & Destructor Documentation

4.12.1.1 F9Stage1()

```
F9Stage1::F9Stage1 ( )
```

Construct a new **F9Stage1** (p. 25) object.

Author

Derived

4.12.1.2 ∼F9Stage1()

```
F9Stage1::∼F9Stage1 ( )
```

Destroy the **F9Stage1** (p. 25) object.

Author

Derived

4.12.2 Member Function Documentation

4.12.2.1 handleChange()

```
void F9Stage1::handleChange (
              Rocket * r ) [virtual]
```

Handle Change of **F9Stage1** (p. 25) object.

Author

Derived

Parameters

r	

Implements **RocketState** (p. 86).

The documentation for this class was generated from the following files:

- F9Stage1.h
- F9Stage1.cpp

4.13 F9Stage2 Class Reference

Inherits **RocketState**.

Public Member Functions

- void **handleChange** (**Rocket** ∗r)

  *Handle Change of **F9Stage1** (p. 25) object.*
- **F9Stage2** ()

  *Construct a new **F9Stage2** (p. 27) object.*
- ∼**F9Stage2** ()

  *Destroy the **F9Stage2** (p. 27) object.*

4.13.1 Constructor & Destructor Documentation

4.13.1.1 F9Stage2()

```
F9Stage2::F9Stage2 ( )
```

Construct a new **F9Stage2** (p. 27) object.

Author

Derived

4.13.1.2 ∼F9Stage2()

```
F9Stage2::∼F9Stage2 ( )
```

Destroy the **F9Stage2** (p. 27) object.

Author

Derived

4.13.2 Member Function Documentation

4.13.2.1 handleChange()

```
void F9Stage2::handleChange (
              Rocket * r )  [virtual]
```

Handle Change of **F9Stage1** (p. 25) object.

Author

Derived

Parameters

r	

Implements **RocketState** (p. 86).

The documentation for this class was generated from the following files:

- F9Stage2.h
- F9Stage2.cpp

4.14 Falcon9 Class Reference

Inherits **Rocket**.

Public Member Functions

- void **nextstage** ()

  Change to the Next Stage of the Falcon 9 object.
- **Falcon9** (**Payload** ∗)

  Construct a new Falcon 9 object.
- ∼**Falcon9** ()

  Destroy the Falcon 9 object.
- string **getName** ()

  Get the Name object.

Additional Inherited Members

4.14.1 Constructor & Destructor Documentation

4.14.1.1 Falcon9()

```
Falcon9::Falcon9 (
            Payload * payload )
```

Construct a new Falcon 9 object.

Author

Derived

4.14.1.2 ∼Falcon9()

```
Falcon9::∼Falcon9 ( )
```

Destroy the Falcon 9 object.

Author

Derived

4.14.2 Member Function Documentation

4.14.2.1 getName()

```
string Falcon9::getName ( ) [virtual]
```

Get the Name object.

Returns

string

Implements **Rocket** (p. 79).

4.14.2.2 nextstage()

```
void Falcon9::nextstage ( ) [virtual]
```

Change to the Next Stage of the Falcon 9 object.

Author

Derived

Implements **Rocket** (p. 80).

The documentation for this class was generated from the following files:

- Falcon9.h
- Falcon9.cpp

4.15 Falcon9Builder Class Reference

Inherits **RocketBuilder**.

Public Member Functions

- **Falcon9Builder** (**CargoCollection** ∗c)

  Construct a new Falcon 9 Builder object.
- ∼**Falcon9Builder** ()

  Destroy the Falcon 9 Builder object.
- void **reset** ()

  Reset the Falcon 9 Builder.
- void **addPayload** ()

  Add ***Payload*** *(*p. *69) to the Falcon 9 Builder.*
- void **addPropulsion** ()

  Add ***Propulsion*** *(*p. *75) to the Falcon 9 Builder.*
- **Rocket** ∗ **getRocket** ()

  Get the Falcon 9 Builder.

4.15.1 Constructor & Destructor Documentation

4.15.1.1 Falcon9Builder()

```
Falcon9Builder::Falcon9Builder (
            CargoCollection * c )
```

Construct a new Falcon 9 Builder object.

Author

Derived

Parameters

c	

4.15.1.2 ∼Falcon9Builder()

```
Falcon9Builder::∼Falcon9Builder ( )
```

Destroy the Falcon 9 Builder object.

Author

Derived

4.15.2 Member Function Documentation

4.15.2.1 addPayload()

```
void Falcon9Builder::addPayload ( )  [virtual]
```

Add **Payload** (p. 69) to the Falcon 9 Builder.

Author

Derived

Implements **RocketBuilder** (p. 84).

4.15.2.2 addPropulsion()

```
void Falcon9Builder::addPropulsion ( )  [virtual]
```

Add **Propulsion** (p. 75) to the Falcon 9 Builder.

Author

Derived

Implements **RocketBuilder** (p. 84).

4.15.2.3 getRocket()

```
Rocket * Falcon9Builder::getRocket ( )  [virtual]
```

Get the Falcon 9 Builder.

Author

Derived

Returns

Rocket∗

Implements **RocketBuilder** (p. 84).

4.15.2.4 reset()

```
void Falcon9Builder::reset ( )  [virtual]
```

Reset the Falcon 9 Builder.

Author

Derived

Implements **RocketBuilder** (p. 85).

The documentation for this class was generated from the following files:

- Falcon9Builder.h
- Falcon9Builder.cpp

4.16 Falcon9Core Class Reference

Inherits **Core**.

Public Member Functions

- **Falcon9Core** (string name)
  *Construct a new Falcon 9 **Core** (p. 22) object.*
- ∼**Falcon9Core** ()
  *Destroy the Falcon 9 **Core** (p. 22) object.*
- string **getName** ()
  *Get the Name of the Falcon 9 **Core** (p. 22) object.*

4.16.1 Constructor & Destructor Documentation

4.16.1.1 Falcon9Core()

```
Falcon9Core::Falcon9Core (
             string name )
```

Construct a new Falcon 9 **Core** (p. 22) object.

Author

Derived

Parameters

name	

4.16.1.2 ∼Falcon9Core()

```
Falcon9Core::∼Falcon9Core ( )
```

Destroy the Falcon 9 **Core** (p. 22) object.

Author

Derived

4.16.2 Member Function Documentation

4.16.2.1 getName()

```
string Falcon9Core::getName ( )
```

Get the Name of the Falcon 9 **Core** (p. 22) object.

Author

Derived

Returns

string

The documentation for this class was generated from the following files:

- Falcon9Core.h
- Falcon9Core.cpp

4.17 Falcon9Payload Class Reference

Inherits **Payload**.

Public Member Functions

- **Falcon9Payload** (**SpaceCraft** ∗s, **VacuumMerlinEngine** ∗vme)

  *Construct a new Falcon 9 **Payload** (p. 69) object.*
- ∼**Falcon9Payload** ()

  *Destroy the Falcon 9 **Payload** (p. 69) object.*
- void **launch** ()

  *Launch the Falcon 9 **Payload** (p. 69).*

4.17.1 Constructor & Destructor Documentation

4.17.1.1 Falcon9Payload()

```
Falcon9Payload::Falcon9Payload (
              SpaceCraft ∗ s,
              VacuumMerlinEngine ∗ vme )
```

Construct a new Falcon 9 **Payload** (p. 69) object.

Author

Derived

Parameters

s	
vme	

4.17.1.2 ∼Falcon9Payload()

```
Falcon9Payload::∼Falcon9Payload ( )
```

Destroy the Falcon 9 **Payload** (p. 69) object.

Author

Derived

4.17.2 Member Function Documentation

4.17.2.1 launch()

```
void Falcon9Payload::launch ( )  [virtual]
```

Launch the Falcon 9 **Payload** (p. 69).

Author

Derived

Implements **Payload** (p. 71).

The documentation for this class was generated from the following files:

- Falcon9Payload.h
- Falcon9Payload.cpp

4.18 Falcon9Propulsion Class Reference

Inherits **Propulsion**.

Public Member Functions

- **Falcon9Propulsion** (**Payload** ∗p)

    Construct a new Falcon 9 ***Propulsion*** *(p. 75) object.*
- ∼**Falcon9Propulsion** ()

    Destroy the Falcon 9 ***Propulsion*** *(p. 75) object.*
- **PropulsionMemento** ∗ **backup** ()

    Backup the Falcon 9 ***Propulsion*** *(p. 75).*
- void **restore** (**PropulsionMemento** ∗)

    Restore from Falcon 9 ***Propulsion*** *(p. 75) Backup.*
- void **land** ()

    Land Falcon 9 ***Propulsion*** *(p. 75).*
- void **launch** ()

    Launch Falcon 9 ***Propulsion*** *(p. 75).*

4.18.1 Constructor & Destructor Documentation

4.18.1.1 Falcon9Propulsion()

```
Falcon9Propulsion::Falcon9Propulsion (
              Payload ∗ p )
```

Construct a new Falcon 9 **Propulsion** (p. 75) object.

Author

Derived

Parameters

p	

4.18.1.2 ∼Falcon9Propulsion()

```
Falcon9Propulsion::∼Falcon9Propulsion ( )
```

Destroy the Falcon 9 **Propulsion** (p. 75) object.

Author

Derived

4.18.2 Member Function Documentation

4.18.2.1 backup()

```
PropulsionMemento * Falcon9Propulsion::backup ( )
```

Backup the Falcon 9 **Propulsion** (p. 75).

Author

Derived

Returns

PropulsionMemento∗

4.18.2.2 land()

```
void Falcon9Propulsion::land ( ) [virtual]
```

Land Falcon 9 **Propulsion** (p. 75).

Author

Derived

Implements **Propulsion** (p. 75).

4.18.2.3 launch()

```
void Falcon9Propulsion::launch ( )  [virtual]
```

Launch Falcon 9 **Propulsion** (p. 75).

Author

Derived

Implements **Payload** (p. 71).

4.18.2.4 restore()

```
void Falcon9Propulsion::restore (
              PropulsionMemento * corerestore )
```

Restore from Falcon 9 **Propulsion** (p. 75) Backup.

Author

Derived

The documentation for this class was generated from the following files:

- Falcon9Propulsion.h
- Falcon9Propulsion.cpp

4.19 FalconHeavy Class Reference

Inherits **Rocket**.

Public Member Functions

- void **nextstage** ()

  Move to Next Stage of Falcon Heavy.
- **FalconHeavy** (**Payload** ∗payload)

  Construct a new Falcon Heavy object.
- ∼**FalconHeavy** ()

  Destroy the Falcon Heavy object.
- string **getName** ()

  Get the Name object.

Additional Inherited Members

4.19.1 Constructor & Destructor Documentation

4.19.1.1 FalconHeavy()

```
FalconHeavy::FalconHeavy (
            Payload * payload )
```

Construct a new Falcon Heavy object.

Author

Derived

Parameters

payload	

4.19.1.2 ∼FalconHeavy()

```
FalconHeavy::∼FalconHeavy ( )
```

Destroy the Falcon Heavy object.

Author

Derived

4.19.2 Member Function Documentation

4.19.2.1 getName()

```
string FalconHeavy::getName ( )  [virtual]
```

Get the Name object.

Returns

string

Implements **Rocket** (p. 79).

4.19.2.2 nextstage()

```
void FalconHeavy::nextstage ( )  [virtual]
```

Move to Next Stage of Falcon Heavy.

Author

Derived

Implements **Rocket** (p. 80).

The documentation for this class was generated from the following files:

- FalconHeavy.h
- FalconHeavy.cpp

4.20 FalconHeavyBuilder Class Reference

Inherits **RocketBuilder**.

Public Member Functions

- **FalconHeavyBuilder** (**CargoCollection** ∗c, **HumanCollection** ∗h)

  Construct a new Falcon Heavy Builder object.
- ∼**FalconHeavyBuilder** ()

  Destroy the Falcon Heavy Builder object.
- void **reset** ()

  Reset the Falcon Heavy Builder
- void **addPayload** ()

  Add Paylaod to the Falcon Heavy Builder.
- void **addPropulsion** ()

  *Add **Propulsion** (p. 75) to the Falcon Heavy Builder.*
- **FalconHeavy** ∗ **getRocket** ()

  *Get the Falcon Heavy **Rocket** (p. 77).*

4.20.1 Constructor & Destructor Documentation

4.20.1.1 FalconHeavyBuilder()

```
FalconHeavyBuilder::FalconHeavyBuilder (
              CargoCollection * c,
              HumanCollection * h )
```

Construct a new Falcon Heavy Builder object.

Author

Derived

Parameters

c	
h	

4.20.1.2 ∼FalconHeavyBuilder()

```
FalconHeavyBuilder::∼FalconHeavyBuilder ( )
```

Destroy the Falcon Heavy Builder object.

Author

Derived

4.20.2 Member Function Documentation

4.20.2.1 addPayload()

```
void FalconHeavyBuilder::addPayload ( )  [virtual]
```

Add Paylaod to the Falcon Heavy Builder.

Author

Derived

Implements **RocketBuilder** (p. 84).

4.20.2.2 addPropulsion()

```
void FalconHeavyBuilder::addPropulsion ( )  [virtual]
```

Add **Propulsion** (p. 75) to the Falcon Heavy Builder.

Author

Derived

Implements **RocketBuilder** (p. 84).

4.20.2.3 getRocket()

```
FalconHeavy * FalconHeavyBuilder::getRocket ( ) [virtual]
```

Get the Falcon Heavy **Rocket** (p. 77).

Author

Derived

Returns

FalconHeavy∗

Implements **RocketBuilder** (p. 84).

4.20.2.4 reset()

```
void FalconHeavyBuilder::reset ( ) [virtual]
```

Reset the Falcon Heavy Builder

Author

Derived

Implements **RocketBuilder** (p. 85).

The documentation for this class was generated from the following files:

- FalconHeavyBuilder.h
- FalconHeavyBuilder.cpp

4.21 FalconHeavyPayload Class Reference

Inherits **Payload**.

Public Member Functions

- **FalconHeavyPayload** (**SpaceCraft** ∗s, **VacuumMerlinEngine** ∗vme)

  *Construct a new Falcon Heavy **Payload** (p. 69) object.*
- ∼**FalconHeavyPayload** ()

  *Destroy the Falcon Heavy **Payload** (p. 69) object.*
- void **launch** ()

  *Launch the Falcon Heavy **Payload** (p. 69).*

4.21.1 Constructor & Destructor Documentation

4.21.1.1 FalconHeavyPayload()

```
FalconHeavyPayload::FalconHeavyPayload (
              SpaceCraft * s,
              VacuumMerlinEngine * vme )
```

Construct a new Falcon Heavy **Payload** (p. 69) object.

Author

Derived

Parameters

s	
vme	

4.21.1.2 ∼FalconHeavyPayload()

```
FalconHeavyPayload::∼FalconHeavyPayload ( )
```

Destroy the Falcon Heavy **Payload** (p. 69) object.

Author

Derived

4.21.2 Member Function Documentation

4.21.2.1 launch()

```
void FalconHeavyPayload::launch ( ) [virtual]
```

Launch the Falcon Heavy **Payload** (p. 69).

Author

Derived

Implements **Payload** (p. 71).

The documentation for this class was generated from the following files:

- FalconHeavyPayload.h
- FalconHeavyPayload.cpp

4.22 FalconHeavyPropulsion Class Reference

Inherits **Propulsion**.

Public Member Functions

- **FalconHeavyPropulsion** (**Payload** ∗p)

  *Construct a new Falcon Heavy **Propulsion** (p. 75) object.*
- ∼**FalconHeavyPropulsion** ()

  *Destroy the Falcon Heavy **Propulsion** (p. 75) object.*
- **FalconHeavyPropulsionMemento** ∗ **backup** ()

  *Backup the Falcon Heavy **Propulsion** (p. 75).*
- void **restore** (**FalconHeavyPropulsionMemento** ∗)

  *Resotre the Falcon Heavy **Propulsion** (p. 75) from the Backup.*
- void **land** ()

  *Land the Falcon Heavy **Propulsion** (p. 75).*
- void **launch** ()

  *Launch the Falcon Heavy **Propulsion** (p. 75).*

4.22.1 Constructor & Destructor Documentation

4.22.1.1 FalconHeavyPropulsion()

```
FalconHeavyPropulsion::FalconHeavyPropulsion (
            Payload * p )
```

Construct a new Falcon Heavy **Propulsion** (p. 75) object.

Author

Derived

Parameters

p	

4.22.1.2 ∼FalconHeavyPropulsion()

```
FalconHeavyPropulsion::∼FalconHeavyPropulsion ( )
```

Destroy the Falcon Heavy **Propulsion** (p. 75) object.

Author

Derived

4.22.2 Member Function Documentation

4.22.2.1 backup()

```
FalconHeavyPropulsionMemento * FalconHeavyPropulsion::backup ( )
```

Backup the Falcon Heavy **Propulsion** (p. 75).

Author

Derived

Returns

FalconHeavyPropulsionMemento∗

4.22.2.2 land()

```
void FalconHeavyPropulsion::land ( ) [virtual]
```

Land the Falcon Heavy **Propulsion** (p. 75).

Author

Derived

Implements **Propulsion** (p. 75).

4.22.2.3 launch()

```
void FalconHeavyPropulsion::launch ( ) [virtual]
```

Launch the Falcon Heavy **Propulsion** (p. 75).

Author

Derived

Implements **Payload** (p. 71).

4.22.2.4 restore()

```
void FalconHeavyPropulsion::restore (
              FalconHeavyPropulsionMemento * corerestore )
```

Resotre the Falcon Heavy **Propulsion** (p. 75) from the Backup.

Author

Derived

The documentation for this class was generated from the following files:

- FalconHeavyPropulsion.h
- FalconHeavyPropulsion.cpp

4.23 FalconHeavyPropulsionMemento Class Reference

Public Member Functions

- **FalconHeavyPropulsionMemento** (**Falcon9Core** ∗, **Falcon9Core** ∗, **Falcon9Core** ∗, **Payload** ∗)

  Construct a new Falcon Heavy ***Propulsion*** *(p. 75) Memento object.*
- **Falcon9Core** ∗ **getCore1** ()

  Get the Falcon Heavy ***Core*** *(p. 22) 1.*
- **Falcon9Core** ∗ **getCore2** ()

  Get the Falcon Heavy ***Core*** *(p. 22) 2.*
- **Falcon9Core** ∗ **getCore3** ()

  Get the Falcon Heavy ***Core*** *(p. 22) 3.*
- **Payload** ∗ **getPay** ()

  Get the ***Payload*** *(p. 69).*

4.23.1 Constructor & Destructor Documentation

4.23.1.1 FalconHeavyPropulsionMemento()

```
FalconHeavyPropulsionMemento::FalconHeavyPropulsionMemento (
              Falcon9Core * core1,
              Falcon9Core * core2,
              Falcon9Core * core3,
              Payload * pay )
```

Construct a new Falcon Heavy **Propulsion** (p. 75) Memento object.

Author

Derived

4.23.2 Member Function Documentation

4.23.2.1 getCore1()

```
Falcon9Core * FalconHeavyPropulsionMemento::getCore1 ( )
```

Get the Falcon Heavy **Core** (p. 22) 1.

Author

Derived

Returns

Falcon9Core∗

4.23.2.2 getCore2()

```
Falcon9Core * FalconHeavyPropulsionMemento::getCore2 ( )
```

Get the Falcon Heavy **Core** (p. 22) 2.

Author

Derived

Returns

Falcon9Core∗

4.23.2.3 getCore3()

```
Falcon9Core * FalconHeavyPropulsionMemento::getCore3 ( )
```

Get the Falcon Heavy **Core** (p. 22) 3.

Author

Derived

Returns

Falcon9Core∗

4.23.2.4 getPay()

```
Payload * FalconHeavyPropulsionMemento::getPay ( )
```

Get the **Payload** (p. 69).

Author

Derived

Returns

Payload∗

The documentation for this class was generated from the following files:

- FalconHeavyPropulsionMemento.h
- FalconHeavyPropulsionMemento.cpp

4.24 FHStage1 Class Reference

Inherits **RocketState**.

Public Member Functions

- void **handleChange** (**Rocket** ∗r)

  *Handle Change of **FHStage1** (p. 47).*
- **FHStage1** ()

  *Construct a new **FHStage1** (p. 47) object.*
- ∼**FHStage1** ()

  *Destroy the **FHStage1** (p. 47) object.*

4.24.1 Constructor & Destructor Documentation

4.24.1.1 FHStage1()

```
FHStage1::FHStage1 ( )
```

Construct a new **FHStage1** (p. 47) object.

Author

Derived

4.24.1.2 ∼FHStage1()

```
FHStage1::∼FHStage1 ( )
```

Destroy the **FHStage1** (p. 47) object.

Author

Derived

4.24.2 Member Function Documentation

4.24.2.1 handleChange()

```
void FHStage1::handleChange (
              Rocket * r )  [virtual]
```

Handle Change of **FHStage1** (p. 47).

Author

Derived

Parameters

r	

Implements **RocketState** (p. 86).

The documentation for this class was generated from the following files:

- FHStage1.h
- FHStage1.cpp

4.25 FHStage2 Class Reference

Inherits **RocketState**.

Public Member Functions

- void **handleChange** (**Rocket** ∗r)
  Handle Change of ***FHStage2*** *(p. 48).*
- **FHStage2** ()
  Construct a new ***FHStage2*** *(p. 48) object.*
- ∼**FHStage2** ()
  Destroy the ***FHStage2*** *(p. 48) object.*

4.25.1 Constructor & Destructor Documentation

4.25.1.1 FHStage2()

```
FHStage2::FHStage2 ( )
```

Construct a new **FHStage2** (p. 48) object.

Author

Derived

4.25.1.2 ∼FHStage2()

```
FHStage2::∼FHStage2 ( )
```

Destroy the **FHStage2** (p. 48) object.

Author

Derived

4.25.2 Member Function Documentation

4.25.2.1 handleChange()

```
void FHStage2::handleChange (
              Rocket * r )  [virtual]
```

Handle Change of **FHStage2** (p. 48).

Author

Derived

Parameters

r	

Implements **RocketState** (p. 86).

The documentation for this class was generated from the following files:

- FHStage2.h
- FHStage2.cpp

4.26 GroundMissionControl Class Reference

Inherited by **ConcreteGroundMissionControl**.

Public Member Functions

- **GroundMissionControl** ()

  Construct a new Ground Mission Control object.
- void **attach** (**Satellite** ∗)

  *Attach **Satellite** (p. 87) from the Ground Mission Control.*
- void **detach** (**Satellite** ∗)

  *Detach **Satellite** (p. 87) from the Ground Mission Control.*
- void **notify** ()

  Notify the Ground Mission Control.
- ∼**GroundMissionControl** ()

  Destroy the Ground Mission Control object.

4.26.1 Constructor & Destructor Documentation

4.26.1.1 GroundMissionControl()

```
GroundMissionControl::GroundMissionControl ( )
```

Construct a new Ground Mission Control object.

Author

Derived

4.26.1.2 ∼GroundMissionControl()

```
GroundMissionControl::∼GroundMissionControl ( )
```

Destroy the Ground Mission Control object.

Author

Derived

4.26.2 Member Function Documentation

4.26.2.1 attach()

```
void GroundMissionControl::attach (
              Satellite * addSatellite )
```

Attach **Satellite** (p. 87) from the Ground Mission Control.

Author

Derived

4.26.2.2 detach()

```
void GroundMissionControl::detach (
              Satellite * removeSatellite )
```

Detach **Satellite** (p. 87) from the Ground Mission Control.

Author

Derived

4.26.2.3 notify()

```
void GroundMissionControl::notify ( )
```

Notify the Ground Mission Control.

Author

Derived

The documentation for this class was generated from the following files:

- GroundMissionControl.h
- GroundMissionControl.cpp

4.27 HPCaretaker Class Reference

Public Member Functions

- **HPCaretaker** ()

    Construct a new ***HPCaretaker*** *(p. 52) object.*
- **FalconHeavyPropulsionMemento** ∗ **getBackup** ()

    Get the Backup of the ***HPCaretaker*** *(p. 52).*
- void **setBackup** (**FalconHeavyPropulsionMemento** ∗)

    Set the Backup of the ***HPCaretaker*** *(p. 52).*

4.27.1 Constructor & Destructor Documentation

4.27.1.1 HPCaretaker()

```
HPCaretaker::HPCaretaker ( )
```

Construct a new **HPCaretaker** (p. 52) object.

Author

Derived

4.27.2 Member Function Documentation

4.27.2.1 getBackup()

```
FalconHeavyPropulsionMemento * HPCaretaker::getBackup ( )
```

Get the Backup of the **HPCaretaker** (p. 52).

Author

Derived

Returns

FalconHeavyPropulsionMemento∗

4.27.2.2 setBackup()

```
void HPCaretaker::setBackup (
              FalconHeavyPropulsionMemento * backup )
```

Set the Backup of the **HPCaretaker** (p. 52).

Author

Derived

The documentation for this class was generated from the following files:

- FHCaretaker.h
- FHCaretaker.cpp

4.28 Human Class Reference

Public Member Functions

- string **getName** ()

  Get the Name.
- double **getWeight** ()

  Get the Weight.
- string **getRole** ()

  Get the Role.
- **Human** (string name, double weight, string role)

  *Construct a new **Human** (p. 53) object.*
- ∼**Human** ()

  *Destroy the **Human** (p. 53) object.*

4.28.1 Constructor & Destructor Documentation

4.28.1.1 Human()

```
Human::Human (
             string name,
             double weight,
             string role )
```

Construct a new **Human** (p. 53) object.

Author

Derived

Parameters

name	
weight	
role	

4.28.1.2 ∼Human()

```
Human::∼Human ( )
```

Destroy the **Human** (p. 53) object.

Author

Derived

4.28.2 Member Function Documentation

4.28.2.1 getName()

```
string Human::getName ( )
```

Get the Name.

Author

Derived

Returns

string

4.28.2.2 getRole()

```
string Human::getRole ( )
```

Get the Role.

Author

Derived

Returns

string

4.28.2.3 getWeight()

```
double Human::getWeight ( )
```

Get the Weight.

Author

Derived

Returns

double

The documentation for this class was generated from the following files:

- Human.h
- Human.cpp

4.29 HumanCollection Class Reference

Inherited by **VectorOfHumans**.

Public Member Functions

- **HumanCollection** ()

  *Construct a new **Human** (*p. *53) Collection object.*
- virtual **HumanIterator** ∗ **createHumanIterator** ()=0

  *Create a **Human** (*p. *53) **Iterator** (*p. *61) object.*
- virtual void **addHuman** (**Human** ∗)=0

  *Add a **Human** (*p. *53) to the **Human** (*p. *53) Collection.*
- virtual void **removeHuman** (**Human** ∗)=0

  *Remove a **Human** (*p. *53) from the **Human** (*p. *53) Collection.*
- virtual bool **isEmpty** ()=0

  *Check if the **Human** (*p. *53) Collection object is empty.*
- virtual ∼**HumanCollection** ()

  *Destroy the **Human** (*p. *53) Collection object.*

4.29.1 Constructor & Destructor Documentation

4.29.1.1 HumanCollection()

```
HumanCollection::HumanCollection ( )
```

Construct a new **Human** (p. 53) Collection object.

Author

Derived

4.29.1.2 ∼HumanCollection()

```
HumanCollection::∼HumanCollection ( )  [virtual]
```

Destroy the **Human** (p. 53) Collection object.

Author

Derived

4.29.2 Member Function Documentation

4.29.2.1 addHuman()

```
virtual void HumanCollection::addHuman (
              Human ∗  )  [pure virtual]
```

Add a **Human** (p. 53) to the **Human** (p. 53) Collection.

Author

Derived

Implemented in **VectorOfHumans** (p. 104).

4.29.2.2 createHumanIterator()

```
virtual  HumanIterator ∗ HumanCollection::createHumanIterator ( )  [pure virtual]
```

Create a **Human** (p. 53) **Iterator** (p. 61) object.

Author

Derived

Returns

HumanIterator∗

Implemented in **VectorOfHumans** (p. 104).

4.29.2.3 isEmpty()

```
virtual bool HumanCollection::isEmpty ( )  [pure virtual]
```

Check if the **Human** (p. 53) Collection object is empty.

Author

Derived

Returns

true

false

Implemented in **VectorOfHumans** (p. 104).

4.29.2.4 removeHuman()

```
virtual void HumanCollection::removeHuman (
              Human *  )  [pure virtual]
```

Remove a **Human** (p. 53) from the **Human** (p. 53) Collection.

Author

Derived

Implemented in **VectorOfHumans** (p. 105).

The documentation for this class was generated from the following files:

- HumanCollection.h
- HumanCollection.cpp

4.30 HumanFactory Class Reference

Public Member Functions

- **HumanFactory** ()

  *Construct a new **Human** (p. 53) Factory object.*
- ∼**HumanFactory** ()

  *Destroy the **Human** (p. 53) Factory object.*
- **Human** ∗ **createHuman** (string name, double weight, string role)

  *Create a **Human** (p. 53) object.*

4.30.1 Constructor & Destructor Documentation

4.30.1.1 HumanFactory()

```
HumanFactory::HumanFactory ( )
```

Construct a new **Human** (p. 53) Factory object.

Author

Derived

4.30.1.2 ∼HumanFactory()

```
HumanFactory::∼HumanFactory ( )
```

Destroy the **Human** (p. 53) Factory object.

Author

Derived

4.30.2 Member Function Documentation

4.30.2.1 createHuman()

```
Human * HumanFactory::createHuman (
            string name,
            double weight,
            string role )
```

Create a **Human** (p. 53) object.

Author

Derived

Parameters

name	
weight	
role	

Returns

Human∗

The documentation for this class was generated from the following files:

- HumanFactory.h
- HumanFactory.cpp

4.31 HumanIterator Class Reference

Inherited by **OperationsIteratorHumans**.

Public Member Functions

- **HumanIterator** ()

  *Construct a new **Human** (p. 53) **Iterator** (p. 61) object.*
- virtual **Human** ∗ **first** ()=0

  *Return the first **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual **Human** ∗ **next** ()=0

  *Return the next **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual bool **hasNext** ()=0

  *Check if there is **Cargo** (p. 7) in the next **Iterator** (p. 61).*
- virtual **Human** ∗ **current** ()=0

  *Return the Current **Cargo** (p. 7) in the **Iterator** (p. 61).*
- virtual ∼**HumanIterator** ()

  *Destroy the **Human** (p. 53) **Iterator** (p. 61) object.*

4.31.1 Constructor & Destructor Documentation

4.31.1.1 HumanIterator()

```
HumanIterator::HumanIterator ( )
```

Construct a new **Human** (p. 53) **Iterator** (p. 61) object.

Author

Derived

4.31.1.2 ∼HumanIterator()

```
HumanIterator::∼HumanIterator ( )  [virtual]
```

Destroy the **Human** (p. 53) **Iterator** (p. 61) object.

Author

Derived

4.31.2 Member Function Documentation

4.31.2.1 current()

```
virtual  Human * HumanIterator::current ( )  [pure virtual]
```

Return the Current **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Human∗

Implemented in **OperationsIteratorHumans** (p. 68).

4.31.2.2 first()

```
virtual  Human * HumanIterator::first ( )  [pure virtual]
```

Return the first **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Human∗

Implemented in **OperationsIteratorHumans** (p. 68).

4.31.2.3 hasNext()

```
virtual bool HumanIterator::hasNext ( )  [pure virtual]
```

Check if there is **Cargo** (p. 7) in the next **Iterator** (p. 61).

Author

Derived

Returns

true

false

Implemented in **OperationsIteratorHumans** (p. 68).

4.31.2.4 next()

```
virtual  Human * HumanIterator::next ( )  [pure virtual]
```

Return the next **Cargo** (p. 7) in the **Iterator** (p. 61).

Author

Derived

Returns

Human∗

Implemented in **OperationsIteratorHumans** (p. 69).

The documentation for this class was generated from the following files:

- HumanIterator.h
- HumanIterator.cpp

4.32 Iterator Class Reference

The documentation for this class was generated from the following file:

- Iterator.h

4.33 LaunchMode Class Reference

Public Member Functions

- **LaunchMode** (**Simulation** ∗obj)

  Construct a new Launch Mode object.
- void **execute** ()

  Execture Launch Mode.
- **Simulation** ∗ **getR** ()

  Return Launch Mode Receiver.

4.33.1 Constructor & Destructor Documentation

4.33.1.1 LaunchMode()

```
LaunchMode::LaunchMode (
              Simulation * obj )
```

Construct a new Launch Mode object.

Author

Derived

Parameters

obj	

4.33.2 Member Function Documentation

4.33.2.1 execute()

```
void LaunchMode::execute ( )
```

Execture Launch Mode.

Author

Derived

4.33.2.2 getR()

```
Simulation * LaunchMode::getR ( )
```

Return Launch Mode Receiver.

Author

Derived

Returns

Simulation*

The documentation for this class was generated from the following files:

- LaunchMode.h
- LaunchMode.cpp

4.34 MerlinEngine Class Reference

Inherits **Engine**.

Public Member Functions

- **MerlinEngine** (string name)

  Construct a new Merlin ***Engine*** *(*p. *25) object.*
- **MerlinEngine * clone** ()

  Clone the Merlin ***Engine*** *(*p. *25) object.*
- ∼**MerlinEngine** ()

  Destroy the Merlin ***Engine*** *(*p. *25) object.*

4.34.1 Constructor & Destructor Documentation

4.34.1.1 MerlinEngine()

```
MerlinEngine::MerlinEngine (
             string name )
```

Construct a new Merlin **Engine** (p. 25) object.

Author

Derived

Parameters

name	

4.34.1.2 ∼MerlinEngine()

```
MerlinEngine::∼MerlinEngine ( )
```

Destroy the Merlin **Engine** (p. 25) object.

Author

Derived

4.34.2 Member Function Documentation

4.34.2.1 clone()

```
MerlinEngine ∗ MerlinEngine::clone ( )
```

Clone the Merlin **Engine** (p. 25) object.

Author

Derived

Returns

MerlinEngine∗

The documentation for this class was generated from the following files:

- MerlinEngine.h
- MerlinEngine.cpp

4.35 OperationsIteratorCargo Class Reference

Inherits **CargoIterator**.

Public Member Functions

- **OperationsIteratorCargo** (vector< **Cargo** ∗ > vect)

  *Construct a new Operations **Iterator** (p. 61) **Cargo** (p. 7) object.*
- **Cargo** ∗ **first** () override

  *Return the first Operation in the **Iterator** (p. 61).*
- **Cargo** ∗ **next** () override

  *Return the next Operation in the **Iterator** (p. 61).*
- bool **hasNext** () override

  *Check if there is an Operation in the next **Iterator** (p. 61).*
- **Cargo** ∗ **current** () override

  *Return the Current Operation in the **Iterator** (p. 61).*
- ∼**OperationsIteratorCargo** () override=default

  *Destroy the Operations **Iterator** (p. 61) **Cargo** (p. 7) object.*

4.35.1 Constructor & Destructor Documentation

4.35.1.1 OperationsIteratorCargo()

```
OperationsIteratorCargo::OperationsIteratorCargo (
            vector< Cargo ∗ > vect ) [explicit]
```

Construct a new Operations **Iterator** (p. 61) **Cargo** (p. 7) object.

Author

Derived

Parameters

vect	

4.35.1.2 ∼OperationsIteratorCargo()

```
OperationsIteratorCargo::∼OperationsIteratorCargo ( ) [override], [default]
```

Destroy the Operations **Iterator** (p. 61) **Cargo** (p. 7) object.

Author

Derived

4.35.2 Member Function Documentation

4.35.2.1 current()

```
Cargo * OperationsIteratorCargo::current ( )  [override], [virtual]
```

Return the Current Operation in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implements **CargoIterator** (p. 14).

4.35.2.2 first()

```
Cargo * OperationsIteratorCargo::first ( )  [override], [virtual]
```

Return the first Operation in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implements **CargoIterator** (p. 14).

4.35.2.3 hasNext()

```
bool OperationsIteratorCargo::hasNext ( )  [override], [virtual]
```

Check if there is an Operation in the next **Iterator** (p. 61).

Author

Derived

Returns

true

false

Implements **CargoIterator** (p. 14).

4.35.2.4 next()

```
Cargo * OperationsIteratorCargo::next ( ) [override], [virtual]
```

Return the next Operation in the **Iterator** (p. 61).

Author

Derived

Returns

Cargo∗

Implements **CargoIterator** (p. 15).

The documentation for this class was generated from the following files:

- OperationsIteratorCargo.h
- OperationsIteratorCargo.cpp

4.36 OperationsIteratorHumans Class Reference

Inherits **HumanIterator**.

Public Member Functions

- **OperationsIteratorHumans** (vector< **Human** ∗ > vect)
  *Construct a new Operations **Iterator** (p. 61) Humans object.*
- **Human** ∗ **first** () override
  *Return the first **Human** (p. 53) in the **Iterator** (p. 61).*
- **Human** ∗ **next** () override
  *Return the next **Human** (p. 53) in the **Iterator** (p. 61).*
- bool **hasNext** () override
  *Check if there is a **Human** (p. 53) in the next **Iterator** (p. 61).*
- **Human** ∗ **current** () override
  *Return the current **Human** (p. 53) in the **Iterator** (p. 61).*
- ∼**OperationsIteratorHumans** () override=default
  *Destroy the Operations **Iterator** (p. 61) Humans object.*

4.36.1 Constructor & Destructor Documentation

4.36.1.1 OperationsIteratorHumans()

```
OperationsIteratorHumans::OperationsIteratorHumans (
            vector< Human * > vect ) [explicit]
```

Construct a new Operations **Iterator** (p. 61) Humans object.

Author

Derived

Parameters

vect	

4.36.1.2 ∼OperationsIteratorHumans()

```
OperationsIteratorHumans::∼OperationsIteratorHumans ( )  [override], [default]
```

Destroy the Operations **Iterator** (p. 61) Humans object.

Author

Derived

4.36.2 Member Function Documentation

4.36.2.1 current()

```
Human ∗ OperationsIteratorHumans::current ( )  [override], [virtual]
```

Return the current **Human** (p. 53) in the **Iterator** (p. 61).

Author

Derived

Returns

Human∗

Implements **HumanIterator** (p. 60).

4.36.2.2 first()

```
Human ∗ OperationsIteratorHumans::first ( )  [override], [virtual]
```

Return the first **Human** (p. 53) in the **Iterator** (p. 61).

Author

Derived

Returns

Human∗

Implements **HumanIterator** (p. 60).

4.36.2.3 hasNext()

```
bool OperationsIteratorHumans::hasNext ( )  [override], [virtual]
```

Check if there is a **Human** (p. 53) in the next **Iterator** (p. 61).

Author

Derived

Returns

true

false

Implements **HumanIterator** (p. 60).

4.36.2.4 next()

```
Human * OperationsIteratorHumans::next ( )  [override], [virtual]
```

Return the next **Human** (p. 53) in the **Iterator** (p. 61).

Author

Derived

Returns

Human*

Implements **HumanIterator** (p. 61).

The documentation for this class was generated from the following files:

- OperationsIteratorHumans.h
- OperationsIteratorHumans.cpp

4.37 Payload Class Reference

Inherited by **Falcon9Payload**, **FalconHeavyPayload**, and **Propulsion**.

Public Member Functions

- **VacuumMerlinEngine** ∗ **getVacuumMerlinEngine** ()

  *Get the Vacuum Merlin **Engine** (p. 25).*
- void **setVacuumMerlinEngine** (**VacuumMerlinEngine** ∗v)

  *Set the Vacuum Merlin **Engine** (p. 25).*
- **SpaceCraft** ∗ **getSpaceCraft** ()

  Get the Space Craft object.
- void **setSpaceCraft** (**SpaceCraft** ∗s)

  Set the Space Craft.
- void **print** ()

  Print the Space Craft.
- **Payload** (**SpaceCraft** ∗s, **VacuumMerlinEngine** ∗v)

  *Construct a new **Payload** (p. 69).*
- **Payload** ()

  *Construct a new **Payload** (p. 69) object.*
- virtual ∼**Payload** ()

  *Destroy the **Payload** (p. 69) object.*
- bool **testVacuumMerlinEngine** ()

  *Test the Vacuum Merlin **Engine** (p. 25).*
- virtual void **launch** ()=0

  *Launch the Vacuum Merlin **Engine** (p. 25).*

4.37.1 Constructor & Destructor Documentation

4.37.1.1 Payload() [1/2]

```
Payload::Payload (
              SpaceCraft * s,
              VacuumMerlinEngine * v )
```

Construct a new **Payload** (p. 69).

Author

Derived

Parameters

s	
v	

4.37.1.2 Payload() [2/2]

```
Payload::Payload ( )
```

Construct a new **Payload** (p. 69) object.

Author

Derived

4.37.1.3 ∼Payload()

```
Payload::∼Payload ( ) [virtual]
```

Destroy the **Payload** (p. 69) object.

Author

Derived

4.37.2 Member Function Documentation

4.37.2.1 getSpaceCraft()

```
SpaceCraft ∗ Payload::getSpaceCraft ( )
```

Get the Space Craft object.

Author

Derived

Returns

SpaceCraft∗

4.37.2.2 getVacuumMerlinEngine()

```
VacuumMerlinEngine ∗ Payload::getVacuumMerlinEngine ( )
```

Get the Vacuum Merlin **Engine** (p. 25).

Author

Derived

Returns

VacuumMerlinEngine∗

4.37.2.3 launch()

```
virtual void Payload::launch ( )  [pure virtual]
```

Launch the Vacuum Merlin **Engine** (p. 25).

Author

Derived

Implemented in **Falcon9Payload** (p. 34), **Falcon9Propulsion** (p. 36), **FalconHeavyPayload** (p. 42), and **FalconHeavyPropulsion** (p. 44).

4.37.2.4 print()

```
void Payload::print ( )
```

Print the Space Craft.

Author

Derived

4.37.2.5 setSpaceCraft()

```
void Payload::setSpaceCraft (
              SpaceCraft * s )
```

Set the Space Craft.

Author

Derived

Parameters

s	

4.37.2.6 setVacuumMerlinEngine()

```
void Payload::setVacuumMerlinEngine (
              VacuumMerlinEngine * v )
```

Set the Vacuum Merlin **Engine** (p. 25).

Author

Derived

Parameters

v	

4.37.2.7 testVacuumMerlinEngine()

```
bool Payload::testVacuumMerlinEngine ( )
```

Test the Vacuum Merlin **Engine** (p. 25).

Author

Derived

Returns

true

false

The documentation for this class was generated from the following files:

- Payload.h
- Payload.cpp

4.38 PropCaretaker Class Reference

Public Member Functions

- **PropCaretaker** ()

  Construct a new Prop Caretaker object.
- **PropulsionMemento** ∗ **getBackup** ()

  Get the Backup of the ***PropulsionMemento*** *(*p. *76).*
- void **setBackup** (**PropulsionMemento** ∗)

  Set the Backup of the ***PropulsionMemento*** *(*p. *76).*

4.38.1 Constructor & Destructor Documentation

4.38.1.1 PropCaretaker()

```
PropCaretaker::PropCaretaker ( )
```

Construct a new Prop Caretaker object.

Author

Derived

4.38.2 Member Function Documentation

4.38.2.1 getBackup()

```
PropulsionMemento * PropCaretaker::getBackup ( )
```

Get the Backup of the **PropulsionMemento** (p. 76).

Author

Derived

Returns

PropulsionMemento*

4.38.2.2 setBackup()

```
void PropCaretaker::setBackup (
              PropulsionMemento * backup )
```

Set the Backup of the **PropulsionMemento** (p. 76).

Author

Derived

The documentation for this class was generated from the following files:

- PropCaretaker.h
- PropCaretaker.cpp

4.39 Propulsion Class Reference

Inherits **Payload**.

Inherited by **Falcon9Propulsion**, and **FalconHeavyPropulsion**.

Public Member Functions

- **Propulsion** ()

  Construct a new ***Propulsion*** *(p. 75) object.*
- virtual ∼**Propulsion** ()

  Destroy the ***Propulsion*** *(p. 75) object.*
- virtual void **land** ()=0

  Land the ***Propulsion*** *(p. 75).*
- bool **testPropulsion** ()

  Test the ***Propulsion*** *(p. 75).*

4.39.1 Constructor & Destructor Documentation

4.39.1.1 Propulsion()

```
Propulsion::Propulsion ( )
```

Construct a new **Propulsion** (p. 75) object.

Author

Derived

4.39.1.2 ∼Propulsion()

```
Propulsion::∼Propulsion ( ) [virtual]
```

Destroy the **Propulsion** (p. 75) object.

Author

Derived

4.39.2 Member Function Documentation

4.39.2.1 land()

```
virtual void Propulsion::land ( )  [pure virtual]
```

Land the **Propulsion** (p. 75).

Author

Derived

Implemented in **Falcon9Propulsion** (p. 36), and **FalconHeavyPropulsion** (p. 44).

4.39.2.2 testPropulsion()

```
bool Propulsion::testPropulsion ( )
```

Test the **Propulsion** (p. 75).

Author

Derived

Returns

true

false

The documentation for this class was generated from the following files:

- Propulsion.h
- Propulsion.cpp

4.40 PropulsionMemento Class Reference

Public Member Functions

- **PropulsionMemento** (**Falcon9Core** ∗core, **Payload** ∗pay)

  *Construct a new **Propulsion** (p. 75) Memento object.*
- **Falcon9Core** ∗ **getCore** ()

  *Get the **Core** (p. 22).*
- **Payload** ∗ **getPay** ()

  *Get the **Payload** (p. 69).*

4.40.1 Constructor & Destructor Documentation

4.40.1.1 PropulsionMemento()

```
PropulsionMemento::PropulsionMemento (
              Falcon9Core * core,
              Payload * pay )
```

Construct a new **Propulsion** (p. 75) Memento object.

Author

Derived

Parameters

core	
pay	

4.40.2 Member Function Documentation

4.40.2.1 getCore()

```
Falcon9Core * PropulsionMemento::getCore ( )
```

Get the **Core** (p. 22).

Author

Derived

Returns

Falcon9Core∗

4.40.2.2 getPay()

```
Payload * PropulsionMemento::getPay ( )
```

Get the **Payload** (p. 69).

Author

Derived

Returns

Payload∗

The documentation for this class was generated from the following files:

- PropulsionMemento.h
- PropulsionMemento.cpp

4.41 Rocket Class Reference

Inherited by **Falcon9**, and **FalconHeavy**.

Public Member Functions

- **Rocket** (**Payload** ∗payload)

  Construct a new ***Rocket*** *(p. 77) object.*
- virtual ∼**Rocket** ()

  Destroy the ***Rocket*** *(p. 77) object.*
- virtual void **nextstage** ()=0

  Go to Next Stage of ***Rocket*** *(p. 77).*
- void **setPayload** (**Payload** ∗p)

  Set the ***Payload*** *(p. 69).*
- void **print** ()

  Print ***Rocket*** *(p. 77) State.*
- **Payload** ∗ **getPayload** ()

  Get the ***Payload*** *(p. 69).*
- void **setState** (**RocketState** ∗state)

  Set the ***Rocket*** *(p. 77) State.*
- void **setPropulsion** (**Propulsion** ∗p)

  Set the ***Propulsion*** *(p. 75).*
- void **launch** ()

  Launch ***Rocket*** *(p. 77) object.*
- void **testRocket** ()

  Test ***Rocket*** *(p. 77) object.*
- bool **testPropulsion** ()

  Test ***Propulsion*** *(p. 75).*
- bool **testVacuumMerlinEngine** ()

  Test Vacuum Merlin ***Engine*** *(p. 25).*
- void **setName** (string name)

  Set the Name.
- virtual string **getName** ()=0

  Get the Name object.
- void **testThatFails** ()

  This Test will always fail.

Protected Attributes

- **Payload** ∗ **payload**
- **RocketState** ∗ **stage**
- **Propulsion** ∗ **propulsion**

4.41.1 Constructor & Destructor Documentation

4.41.1.1 Rocket()

```
Rocket::Rocket (
            Payload * payload )
```

Construct a new **Rocket** (p. 77) object.

Author

Derived

Parameters

payload	

4.41.1.2 ∼Rocket()

```
Rocket::∼Rocket ( )  [virtual]
```

Destroy the **Rocket** (p. 77) object.

Author

Derived

4.41.2 Member Function Documentation

4.41.2.1 getName()

```
virtual string Rocket::getName ( )  [pure virtual]
```

Get the Name object.

Author

Derived

Returns

string

Implemented in **Falcon9** (p. 29), and **FalconHeavy** (p. 38).

4.41.2.2 getPayload()

```
Payload ∗ Rocket::getPayload ( )
```

Get the **Payload** (p. 69).

Author

Derived

Returns

Payload∗

4.41.2.3 launch()

```
void Rocket::launch ( )
```

Launch **Rocket** (p. 77) object.

Author

Derived

4.41.2.4 nextstage()

```
void Rocket::nextstage ( )  [pure virtual]
```

Go to Next Stage of **Rocket** (p. 77).

Author

Derived

Implemented in **Falcon9** (p. 29), and **FalconHeavy** (p. 38).

4.41.2.5 print()

```
void Rocket::print ( )
```

Print **Rocket** (p. 77) State.

Author

Derived

4.41.2.6 setName()

```
void Rocket::setName (
             string name )
```

Set the Name.

Author

Derived

Parameters

name	

4.41.2.7 setPayload()

```
void Rocket::setPayload (
              Payload * p )
```

Set the **Payload** (p. 69).

Author

Derived

Parameters

p	

4.41.2.8 setPropulsion()

```
void Rocket::setPropulsion (
              Propulsion * p )
```

Set the **Propulsion** (p. 75).

Author

Derived

Parameters

p	

4.41.2.9 setState()

```
void Rocket::setState (
              RocketState * state )
```

Set the **Rocket** (p. 77) State.

Author

Derived

Parameters

state	

4.41.2.10 testPropulsion()

```
bool Rocket::testPropulsion ( )
```

Test **Propulsion** (p. 75).

Author

Derived

Returns

true

false

4.41.2.11 testRocket()

```
void Rocket::testRocket ( )
```

Test **Rocket** (p. 77) object.

Author

Derived

4.41.2.12 testThatFails()

```
void Rocket::testThatFails ( )
```

This Test will always fail.

Author

Derived

4.41.2.13 testVacuumMerlinEngine()

```
bool Rocket::testVacuumMerlinEngine ( )
```

Test Vacuum Merlin **Engine** (p. 25).

Author

Derived

Returns

true

false

The documentation for this class was generated from the following files:

- Rocket.h
- Rocket.cpp

4.42 RocketBuilder Class Reference

Inherited by **Falcon9Builder**, and **FalconHeavyBuilder**.

Public Member Functions

- **RocketBuilder** ()

  *Construct a new **Rocket** (p. 77) Builder object.*
- virtual ∼**RocketBuilder** ()

  *Destroy the **Rocket** (p. 77) Builder object.*
- virtual void **reset** ()=0

  *Reset the **Rocket** (p. 77).*
- virtual void **addPayload** ()=0

  *Add **Payload** (p. 69) to the **Rocket** (p. 77).*
- virtual void **addPropulsion** ()=0

  *Add **Propulsion** (p. 75) to the **Rocket** (p. 77).*
- virtual **Rocket** ∗ **getRocket** ()=0

  *Get the **Rocket** (p. 77) oject.*

4.42.1 Constructor & Destructor Documentation

4.42.1.1 RocketBuilder()

```
RocketBuilder::RocketBuilder ( )
```

Construct a new **Rocket** (p. 77) Builder object.

Author

Derived

4.42.1.2 ∼RocketBuilder()

```
RocketBuilder::∼RocketBuilder ( )  [virtual]
```

Destroy the **Rocket** (p. 77) Builder object.

Author

Derived

4.42.2 Member Function Documentation

4.42.2.1 addPayload()

```
virtual void RocketBuilder::addPayload ( )  [pure virtual]
```

Add **Payload** (p. 69) to the **Rocket** (p. 77).

Author

Derived

Implemented in **Falcon9Builder** (p. 30), and **FalconHeavyBuilder** (p. 40).

4.42.2.2 addPropulsion()

```
virtual void RocketBuilder::addPropulsion ( )  [pure virtual]
```

Add **Propulsion** (p. 75) to the **Rocket** (p. 77).

Author

Derived

Implemented in **Falcon9Builder** (p. 31), and **FalconHeavyBuilder** (p. 40).

4.42.2.3 getRocket()

```
virtual  Rocket * RocketBuilder::getRocket ( )  [pure virtual]
```

Get the **Rocket** (p. 77) oject.

Author

Derived

Returns

Rocket∗

Implemented in **Falcon9Builder** (p. 31), and **FalconHeavyBuilder** (p. 40).

4.42.2.4 reset()

```
virtual void RocketBuilder::reset ( )  [pure virtual]
```

Reset the **Rocket** (p. 77).

Author

Derived

Implemented in **Falcon9Builder** (p. 31), and **FalconHeavyBuilder** (p. 41).

The documentation for this class was generated from the following files:

- RocketBuilder.h
- RocketBuilder.cpp

4.43 RocketState Class Reference

Inherited by **F9Stage1**, **F9Stage2**, **FHStage1**, and **FHStage2**.

Public Member Functions

- virtual void **handleChange** (**Rocket** ∗r)=0

  *Handle Change for the **Rocket** (p. 77) State.*
- **RocketState** ()

  *Construct a new **Rocket** (p. 77) State object.*
- virtual ∼**RocketState** ()

  *Destroy the **Rocket** (p. 77) State object.*

4.43.1 Constructor & Destructor Documentation

4.43.1.1 RocketState()

```
RocketState::RocketState ( )
```

Construct a new **Rocket** (p. 77) State object.

Author

Derived

4.43.1.2 ∼RocketState()

```
RocketState::∼RocketState ( ) [virtual]
```

Destroy the **Rocket** (p. 77) State object.

Author

Derived

4.43.2 Member Function Documentation

4.43.2.1 handleChange()

```
virtual void RocketState::handleChange (
              Rocket * r ) [pure virtual]
```

Handle Change for the **Rocket** (p. 77) State.

Author

Derived

Parameters

r	

Implemented in **F9Stage1** (p. 26), **F9Stage2** (p. 27), **FHStage1** (p. 48), and **FHStage2** (p. 49).

The documentation for this class was generated from the following files:

- RocketState.h
- RocketState.cpp

4.44 Satellite Class Reference

Inherits **Cargo**.

Inherited by **StarLinkSatellite**.

Public Member Functions

- **Satellite** ()

  Construct a new ***Satellite*** *(*p. *87) object.*
- void **ConnectionChanged** ()

  Nofity ***Satellite*** *(*p. *87) when Connection Change occurs.*
- virtual string **getConnection** ()=0

  Get the Connection.
- virtual void **setConnection** (string)=0

  Set the Connection.
- virtual void **update** ()=0

  Update the ***Satellite*** *(*p. *87).*
- virtual void **print** ()=0

  Print the ***Satellite*** *(*p. *87).*
- ∼**Satellite** ()

  Destroy the ***Satellite*** *(*p. *87) object.*

4.44.1 Constructor & Destructor Documentation

4.44.1.1 Satellite()

```
Satellite::Satellite ( )
```

Construct a new **Satellite** (p. 87) object.

Author

Derived

4.44.1.2 ∼Satellite()

```
Satellite::∼Satellite ( )
```

Destroy the **Satellite** (p. 87) object.

Author

Derived

4.44.2 Member Function Documentation

4.44.2.1 ConnectionChanged()

```
void Satellite::ConnectionChanged ( )
```

Nofity **Satellite** (p. 87) when Connection Change occurs.

Author

Derived

4.44.2.2 getConnection()

```
virtual string Satellite::getConnection ( )  [pure virtual]
```

Get the Connection.

Author

Derived

Returns

string

Implemented in **StarLinkSatellite** (p. 97).

4.44.2.3 print()

```
virtual void Satellite::print ( )  [pure virtual]
```

Print the **Satellite** (p. 87).

Author

Derived

Implemented in **StarLinkSatellite** (p. 98).

4.44.2.4 setConnection()

```
virtual void Satellite::setConnection (
             string  )  [pure virtual]
```

Set the Connection.

Author

Derived

Implemented in **StarLinkSatellite** (p. 98).

4.44.2.5 update()

```
virtual void Satellite::update ( )  [pure virtual]
```

Update the **Satellite** (p. 87).

Author

Derived

Implemented in **StarLinkSatellite** (p. 99).

The documentation for this class was generated from the following files:

- Satellite.h
- Satellite.cpp

4.45 SatelliteTransmission Class Reference

Inherited by **SatelliteTransmissionController**.

Public Member Functions

- **SatelliteTransmission** ()

  Construct a new ***Satellite*** *(p. 87) Transmission object.*
- virtual void **notify** (**Satellite** ∗)=0

  Notify ***Satellite*** *(p. 87) Transmission.*
- ∼**SatelliteTransmission** ()

  Destroy the ***Satellite*** *(p. 87) Transmission object.*

4.45.1 Constructor & Destructor Documentation

4.45.1.1 SatelliteTransmission()

```
SatelliteTransmission::SatelliteTransmission ( )
```

Construct a new **Satellite** (p. 87) Transmission object.

Author

Derived

4.45.1.2 ∼SatelliteTransmission()

```
SatelliteTransmission::∼SatelliteTransmission ( )
```

Destroy the **Satellite** (p. 87) Transmission object.

Author

Derived

4.45.2 Member Function Documentation

4.45.2.1 notify()

```
virtual void SatelliteTransmission::notify (
            Satellite ∗  )  [pure virtual]
```

Notify **Satellite** (p. 87) Transmission.

Author

Derived

Implemented in **SatelliteTransmissionController** (p. 91).

The documentation for this class was generated from the following files:

- SatelliteTransmission.h
- SatelliteTransmission.cpp

4.46 SatelliteTransmissionController Class Reference

Inherits **SatelliteTransmission**.

Public Member Functions

- **SatelliteTransmissionController** ()

  *Construct a new **Satellite** (p. 87) Transmission Controller object.*
- void **notify** (**Satellite** ∗)

  *Notify **Satellite** (p. 87) Transmission Controller.*
- ∼**SatelliteTransmissionController** ()

  *Destroy the **Satellite** (p. 87) Transmission Controller object.*

4.46.1 Constructor & Destructor Documentation

4.46.1.1 SatelliteTransmissionController()

```
SatelliteTransmissionController::SatelliteTransmissionController ( )
```

Construct a new **Satellite** (p. 87) Transmission Controller object.

Author

Derived

4.46.1.2 ∼SatelliteTransmissionController()

```
SatelliteTransmissionController::∼SatelliteTransmissionController ( )
```

Destroy the **Satellite** (p. 87) Transmission Controller object.

Author

Derived

4.46.2 Member Function Documentation

4.46.2.1 notify()

```
void SatelliteTransmissionController::notify (
              Satellite * satellite )  [virtual]
```

Notify **Satellite** (p. 87) Transmission Controller.

Author

Derived

Implements **SatelliteTransmission** (p. 90).

The documentation for this class was generated from the following files:

- SatelliteTransmissionController.h
- SatelliteTransmissionController.cpp

4.47 Simulation Class Reference

Public Member Functions

- **Simulation** ()

  *Construct a new **Simulation** (p. 92) object.*
- ∼**Simulation** ()

  *Destroy the **Simulation** (p. 92) object.*
- void **setup** ()

  *Setup **Simulation** (p. 92).*
- void **liftOff** ()

  *Lift Off **Simulation** (p. 92).*
- void **staticFire** ()

  *Static Fire **Simulation** (p. 92).*
- void **testThatFails** ()

  *Test that Fails **Simulation** (p. 92).*

4.47.1 Constructor & Destructor Documentation

4.47.1.1 Simulation()

```
Simulation::Simulation ( )
```

Construct a new **Simulation** (p. 92) object.

Author

Derived

4.47.1.2 ∼Simulation()

```
Simulation::∼Simulation ( )
```

Destroy the **Simulation** (p. 92) object.

Author

Derived

4.47.2 Member Function Documentation

4.47.2.1 liftOff()

```
void Simulation::liftOff ( )
```

Lift Off **Simulation** (p. 92).

Author

Derived

4.47.2.2 setup()

```
void Simulation::setup ( )
```

Setup **Simulation** (p. 92).

Author

Derived

4.47.2.3 staticFire()

```
void Simulation::staticFire ( )
```

Static Fire **Simulation** (p. 92).

Author

Derived

4.47.2.4 testThatFails()

```
void Simulation::testThatFails ( )
```

Test that Fails **Simulation** (p. 92).

Author

Derived

The documentation for this class was generated from the following files:

- Simulation.h
- Simulation.cpp

4.48 SpaceCraft Class Reference

Inherited by **CrewDragon**, and **Dragon**.

Public Member Functions

- **SpaceCraft** (**CargoCollection** ∗c)

  Construct a new Space Craft object.
- virtual ∼**SpaceCraft** ()

  Destroy the Space Craft object.
- void **setCargoCollection** (**CargoCollection** ∗c)

  *Set the **Cargo** (p. 7) Collection.*
- **CargoCollection** ∗ **getCargoCollection** ()

  *Get the **Cargo** (p. 7) Collection.*
- virtual void **printSpaceCraft** ()=0

  Print Space Crafts in Collection.

4.48.1 Constructor & Destructor Documentation

4.48.1.1 SpaceCraft()

```
SpaceCraft::SpaceCraft (
              CargoCollection * c )
```

Construct a new Space Craft object.

Author

Derived

Parameters

c	

4.48.1.2 ∼SpaceCraft()

```
SpaceCraft::∼SpaceCraft ( )  [virtual]
```

Destroy the Space Craft object.

Author

Derived

4.48.2 Member Function Documentation

4.48.2.1 getCargoCollection()

```
CargoCollection * SpaceCraft::getCargoCollection ( )
```

Get the **Cargo** (p. 7) Collection.

Author

Derived

Returns

CargoCollection∗

4.48.2.2 printSpaceCraft()

```
virtual void SpaceCraft::printSpaceCraft ( )  [pure virtual]
```

Print Space Crafts in Collection.

Author

Derived

Implemented in **CrewDragon** (p. 23), and **Dragon** (p. 24).

4.48.2.3 setCargoCollection()

```
void SpaceCraft::setCargoCollection (
              CargoCollection * c )
```

Set the **Cargo** (p. 7) Collection.

Author

Derived

Parameters

c	

The documentation for this class was generated from the following files:

- SpaceCraft.h
- SpaceCraft.cpp

4.49 SpaceCraftFactory Class Reference

The documentation for this class was generated from the following file:

- SpaceCraftFactory.h

4.50 StarLinkSatellite Class Reference

Inherits **Satellite**.

Public Member Functions

- **StarLinkSatellite** (**ConcreteGroundMissionControl** ∗)

  *Construct a new Star Link **Satellite** (*p. *87) object.*
- **StarLinkSatellite** (string)

  *Construct a new Star Link **Satellite** (*p. *87) object.*
- **StarLinkSatellite** ∗ **clone** ()

  *Clone the Star Link **Satellite** (*p. *87).*
- string **getName** ()

  *Get the Name of the Star Link **Satellite** (*p. *87).*
- void **setName** (string)

  *Set the Name of the Star Link **Satellite** (*p. *87).*
- void **update** ()

  *Update the Star Link **Satellite** (*p. *87).*
- void **print** ()

  *Print the Star Link **Satellite** (*p. *87).*
- string **getConnection** ()

  Get the Connection.
- void **setConnection** (string)

  Set the Connection.
- ∼**StarLinkSatellite** ()

  *Destroy the Star Link **Satellite** (*p. *87) object.*

4.50.1 Constructor & Destructor Documentation

4.50.1.1 StarLinkSatellite() [1/2]

```
StarLinkSatellite::StarLinkSatellite (
              ConcreteGroundMissionControl * setMissionControl )
```

Construct a new Star Link **Satellite** (p. 87) object.

Author

Derived

4.50.1.2 StarLinkSatellite() [2/2]

```
StarLinkSatellite::StarLinkSatellite (
             string setName )
```

Construct a new Star Link **Satellite** (p. 87) object.

Author

Derived

4.50.1.3 ∼StarLinkSatellite()

```
StarLinkSatellite::∼StarLinkSatellite ( )
```

Destroy the Star Link **Satellite** (p. 87) object.

Author

Derived

4.50.2 Member Function Documentation

4.50.2.1 clone()

```
 StarLinkSatellite * StarLinkSatellite::clone ( )
```

Clone the Star Link **Satellite** (p. 87).

Author

Derived

Returns

StarLinkSatellite∗

4.50.2.2 getConnection()

```
string StarLinkSatellite::getConnection ( )  [virtual]
```

Get the Connection.

Author

Derived

Returns

string

Implements **Satellite** (p. 88).

4.50.2.3 getName()

```
string StarLinkSatellite::getName ( )
```

Get the Name of the Star Link **Satellite** (p. 87).

Author

Derived

Returns

string

4.50.2.4 print()

```
void StarLinkSatellite::print ( )  [virtual]
```

Print the Star Link **Satellite** (p. 87).

Author

Derived

Implements **Satellite** (p. 88).

4.50.2.5 setConnection()

```
void StarLinkSatellite::setConnection (
            string setConnection )  [virtual]
```

Set the Connection.

Author

Derived

Implements **Satellite** (p. 89).

4.50.2.6 setName()

```
void StarLinkSatellite::setName (
            string name )
```

Set the Name of the Star Link **Satellite** (p. 87).

Author

Derived

4.50.2.7 update()

```
void StarLinkSatellite::update ( )  [virtual]
```

Update the Star Link **Satellite** (p. 87).

Author

Derived

Implements **Satellite** (p. 89).

The documentation for this class was generated from the following files:

- StarLinkSatellite.h
- StarLinkSatellite.cpp

4.51 VacuumMerlinEngine Class Reference

Inherits **Engine**.

Public Member Functions

- **VacuumMerlinEngine** ()

  Construct a new Vacuum Merlin ***Engine*** *(p. 25) object.*
- **VacuumMerlinEngine** ∗ **clone** ()

  Clone the Vacuum Merlin ***Engine*** *(p. 25) object.*
- ∼**VacuumMerlinEngine** ()

  Destroy the Vacuum Merlin ***Engine*** *(p. 25) object.*

4.51.1 Constructor & Destructor Documentation

4.51.1.1 VacuumMerlinEngine()

```
VacuumMerlinEngine::VacuumMerlinEngine ( )
```

Construct a new Vacuum Merlin **Engine** (p. 25) object.

Author

Derived

4.51.1.2 ∼VacuumMerlinEngine()

```
VacuumMerlinEngine::∼VacuumMerlinEngine ( )
```

Destroy the Vacuum Merlin **Engine** (p. 25) object.

Author

Derived

4.51.2 Member Function Documentation

4.51.2.1 clone()

```
VacuumMerlinEngine ∗ VacuumMerlinEngine::clone ( )
```

Clone the Vacuum Merlin **Engine** (p. 25) object.

Author

Derived

Returns

VacuumMerlinEngine∗

The documentation for this class was generated from the following files:

- VacuumMerlinEngine.h
- VacuumMerlinEngine.cpp

4.52 VectorOfCargo Class Reference

Inherits **CargoCollection**.

Public Member Functions

- **VectorOfCargo** ()

  *Construct a new Vector Of **Cargo** (*p. 7*) object.*
- **CargoIterator** ∗ **createCargoIterator** () override

  *Create a **Cargo** (*p. 7*) **Iterator** (*p. 61*) object.*
- void **addCargo** (**Cargo** ∗) override

  *Add **Cargo** (*p. 7*) to the Vector.*
- void **removeCargo** (**Cargo** ∗) override

  *Remove **Cargo** (*p. 7*) from the Vector.*
- bool **isEmpty** () override

  Check if the Vector is empty.
- ∼**VectorOfCargo** ()

  *Destroy the Vector Of **Cargo** (*p. 7*) object.*

4.52.1 Constructor & Destructor Documentation

4.52.1.1 VectorOfCargo()

```
VectorOfCargo::VectorOfCargo ( )  [default]
```

Construct a new Vector Of **Cargo** (p. 7) object.

Author

Derived

4.52.1.2 ∼VectorOfCargo()

```
VectorOfCargo::∼VectorOfCargo ( )
```

Destroy the Vector Of **Cargo** (p. 7) object.

Author

Derived

4.52.2 Member Function Documentation

4.52.2.1 addCargo()

```
void VectorOfCargo::addCargo (
              Cargo * cargo )  [override], [virtual]
```

Add **Cargo** (p. 7) to the Vector.

Author

Derived

Implements **CargoCollection** (p. 10).

4.52.2.2 createCargoIterator()

```
CargoIterator * VectorOfCargo::createCargoIterator ( )  [override], [virtual]
```

Create a **Cargo** (p. 7) **Iterator** (p. 61) object.

Author

Derived

Returns

CargoIterator∗

Implements **CargoCollection** (p. 10).

4.52.2.3 isEmpty()

```
bool VectorOfCargo::isEmpty ( )  [override], [virtual]
```

Check if the Vector is empty.

Author

Derived

Returns

true

false

Implements **CargoCollection** (p. 10).

4.52.2.4 removeCargo()

```
void VectorOfCargo::removeCargo (
              Cargo * cargos )  [override], [virtual]
```

Remove **Cargo** (p. 7) from the Vector.

Author

Derived

Implements **CargoCollection** (p. 11).

The documentation for this class was generated from the following files:

- VectorOfCargo.h
- VectorOfCargo.cpp

4.53 VectorOfHumans Class Reference

Inherits **HumanCollection**.

Public Member Functions

- **VectorOfHumans** ()

  Construct a new Vector Of Humans object.
- **HumanIterator** ∗ **createHumanIterator** () override

  *Create a **Human** (p. 53) **Iterator** (p. 61) object.*
- void **addHuman** (**Human** ∗) override

  *Add **Human** (p. 53) to Vector.*
- void **removeHuman** (**Human** ∗) override

  *Remove **Human** (p. 53) from Vector.*
- bool **isEmpty** () override

  Check if the Vector is empty.
- ∼**VectorOfHumans** () override

  Destroy the Vector Of Humans object.

4.53.1 Constructor & Destructor Documentation

4.53.1.1 VectorOfHumans()

```
VectorOfHumans::VectorOfHumans ( )  [default]
```

Construct a new Vector Of Humans object.

Author

Derived

4.53.1.2 ∼VectorOfHumans()

```
VectorOfHumans::∼VectorOfHumans ( )  [override]
```

Destroy the Vector Of Humans object.

Author

Derived

4.53.2 Member Function Documentation

4.53.2.1 addHuman()

```
void VectorOfHumans::addHuman (
              Human * human )  [override], [virtual]
```

Add **Human** (p. 53) to Vector.

Author

Derived

Implements **HumanCollection** (p. 56).

4.53.2.2 createHumanIterator()

```
 HumanIterator * VectorOfHumans::createHumanIterator ( )  [override], [virtual]
```

Create a **Human** (p. 53) **Iterator** (p. 61) object.

Author

Derived

Returns

HumanIterator∗

Implements **HumanCollection** (p. 56).

4.53.2.3 isEmpty()

```
bool VectorOfHumans::isEmpty ( )  [override], [virtual]
```

Check if the Vector is empty.

Author

Derived

Returns

true

false

Implements **HumanCollection** (p. 56).

4.53.2.4 removeHuman()

```
void VectorOfHumans::removeHuman (
              Human * human )  [override], [virtual]
```

Remove **Human** (p. 53) from Vector.

Author

Derived

Implements **HumanCollection** (p. 57).

The documentation for this class was generated from the following files:

- VectorOfHumans.h
- VectorOfHumans.cpp

Chapter 5

File Documentation

5.1 AllHeaders.h

```
#include "Cargo.h"
#include "CargoCollection.h"
#include "CargoIterator.h"
#include "Colours.h"
#include "ConcreteGroundMissionControl.h"
#include "ConfigurationManager.h"
#include "Core.h"
#include "CrewDragon.h"
#include "Dragon.h"
#include "Engine.h"
#include "F9Stage1.h"
#include "F9Stage2.h"
#include "FHStage1.h"
#include "FHStage2.h"
#include "Falcon9.h"
#include "Falcon9Builder.h"
#include "Falcon9Core.h"
#include "Falcon9Payload.h"
#include "Falcon9Propulsion.h"
#include "FalconHeavy.h"
#include "FalconHeavyBuilder.h"
#include "FalconHeavyPayload.h"
#include "FalconHeavyPropulsion.h"
#include "GroundMissionControl.h"
#include "Human.h"
#include "HumanCollection.h"
#include "HumanIterator.h"
#include "Iterator.h"
#include "MerlinEngine.h"
#include "OperationsIteratorCargo.h"
#include "OperationsIteratorHumans.h"
#include "Payload.h"
#include "Propulsion.h"
#include "Rocket.h"
#include "RocketBuilder.h"
#include "RocketState.h"
#include "Satellite.h"
#include "SpaceCraft.h"
#include "SpaceCraftFactory.h"
#include "StarLinkSatellite.h"
#include "VacuumMerlinEngine.h"
#include "VectorOfCargo.h"
#include "VectorOfHumans.h"
```

5.2 Cargo.h

```
#ifndef __Cargo_h__
#define __Cargo_h__

#include <string>
#include "VectorOfCargo.h"

using namespace std;

```

```
class Cargo
{
    private:

        string name;
        double weight;
        string description;

    public:

        Cargo (string name, double weight, string descp);

        string getName();

        string getDescription();

        double getWeight();

        ~Cargo();

};

#endif
```

5.3 CargoCollection.h

```
#ifndef __CargoCollection_h__
#define __CargoCollection_h__

class Cargo;
class CargoIterator;

class CargoCollection
{
    public:

        CargoCollection() = default;

        virtual CargoIterator* createCargoIterator()=0;

        virtual void addCargo(Cargo*)=0;

        virtual void removeCargo(Cargo*)=0;

        virtual bool isEmpty()=0;

        virtual ~CargoCollection()=default;

};

#endif
```

5.4 CargoFactory.h

```
#ifndef CARGOFACTORY_H
#define CARGOFACTORY_H

#include "Cargo.h"
#include "Colours.h"

class CargoFactory
{
    private:

    public:

        CargoFactory();

        ~CargoFactory();

        Cargo* createCargo(string name, double weight, string description);

};

#endif
```

5.5 CargoIterator.h

```
#ifndef __CargoIterator_h__
#define __CargoIterator_h__

#include "CargoCollection.h"

class Cargo;

class CargoIterator
{
    public:

        CargoIterator()=default;

        virtual Cargo* first()=0;

        virtual Cargo* next()=0;

        virtual bool hasNext()=0;

        virtual Cargo* current()=0;

        virtual ~CargoIterator()=default;

};

#endif
```

5.6 Colours.h

```
#ifndef COLOURS_H
#define COLOURS_H

#include <iostream>
#include <iomanip>
#include <ctime>

using namespace std;

/*
This is a class used for the output of code
Pass in the code string you want and the colour will be changed
Example use:
  cout«Colours::purple("hello World")«endl;
The code above will print the text hello World in the colour purple.
*/
class Colours
{
    public:

        static string black(string s) {
            string output = "\033[30m" + s + "\033[0m";
            return output;
        }

        static string red(string s) {
            string output = "\033[31m" + s + "\033[0m";
            return output;
        }

        static string green(string s) {
            string output = "\033[32m" + s + "\033[0m";
            return output;
        }

        static  string yellow(string s) {
            string output = "\033[33m" + s + "\033[0m";
            return output;
        }

        static string blue(string s) {
            string output = "\033[34m" + s + "\033[0m";
            return output;
        }

        static string purple(string s) {
            string output = "\033[35m" + s + "\033[0m";
            return output;
        }

        static string cyan(string s) {
            string output = "\033[36m" + s + "\033[0m";
            return output;
        }

};

#endif
```

5.7 ConcreteGroundMissionControl.h

```
#ifndef CONCRETEGROUNDMISSIONCONTROL_H
#define CONCRETEGROUNDMISSIONCONTROL_H

#include <iostream>
#include "GroundMissionControl.h"

using namespace std;

class ConcreteGroundMissionControl : public GroundMissionControl
{
    private:

        bool connection;

    public:

        ConcreteGroundMissionControl();

        bool hasConnected();

        void setConnection(bool);

        ~ConcreteGroundMissionControl();

};

#endif
```

5.8 ConfigurationManager.h

```
#ifndef DERIVED_CONFIGURATIONMANAGER_H
#define DERIVED_CONFIGURATIONMANAGER_H

#include <iostream>
#include <vector>
#include "RocketBuilder.h"
#include "Human.h"
#include "Cargo.h"
#include "HumanCollection.h"
#include "CargoCollection.h"
#include "Falcon9Builder.h"
#include "FalconHeavyBuilder.h"

class ConfigurationManager {
    private:

        RocketBuilder * builder;

    public:

        ConfigurationManager(CargoCollection * c, HumanCollection * h);

        ~ConfigurationManager();

        Rocket * BuildAndDecorateRocket();

};

#endif //DERIVED_CONFIGURATIONMANAGER_H
```

5.9 Core.h

```
#ifndef __Core_h__
#define __Core_h__

class Core;

class Core
{

};

#endif
```

5.10 CrewDragon.h

```
#ifndef __CrewDragon_h__
#define __CrewDragon_h__

#include "SpaceCraft.h"
#include "HumanCollection.h"
#include "OperationsIteratorHumans.h"

// class SpaceCraft;
class CrewDragon;

class CrewDragon: public SpaceCraft
{
    private:

        HumanCollection * h;
        CargoCollection * c;

    public:

        CrewDragon (CargoCollection * c, HumanCollection * h);

        ~CrewDragon();

        void printSpaceCraft();

};

#endif
```

5.11 Dragon.h

```
#ifndef __Dragon_h__
#define __Dragon_h__

#include "SpaceCraft.h"
#include "CargoCollection.h"

class Dragon: public SpaceCraft
{
    private:

        CargoCollection * c;

    public:

        Dragon (CargoCollection * c);

        ~Dragon();

        void printSpaceCraft();
};

#endif
```

5.12 Engine.h

```
#ifndef __Engine_h__
#define __Engine_h__

#include <iostream>

using namespace std;

class Engine;

 class Engine
 {
     private:

         Engine * next;

     public:

         Engine(){}

 };

 #endif
```

5.13 F9Stage1.h

```
#ifndef DERIVED_F9STAGE1_H
#define DERIVED_F9STAGE1_H

#include "RocketState.h"

class Rocket;

class F9Stage1 : public RocketState
{
    public:

        void handleChange(Rocket* r);

        F9Stage1();

        ~F9Stage1();

};


#endif //DERIVED_F9STAGE1_H
```

5.14 F9Stage2.h

```
#ifndef DERIVED_F9STAGE2_H
#define DERIVED_F9STAGE2_H

#include "RocketState.h"

class Rocket;

class F9Stage2 : public RocketState
{
    public:

        void handleChange(Rocket* r);

        F9Stage2();

        ~F9Stage2();
};


#endif //DERIVED_F9STAGE2_H
```

5.15 Falcon9.h

```
#ifndef __Falcon9_h__
#define __Falcon9_h__

#include "Rocket.h"

class Falcon9 : public Rocket
{

        public:

                void nextstage();

                Falcon9(Payload*);

                ~Falcon9();

                string getName();

};

#endif
```

5.16 Falcon9Builder.h

```
#ifndef DERIVED_FALCON9BUILDER_H
#define DERIVED_FALCON9BUILDER_H

#include <iostream>
#include <vector>
#include "RocketBuilder.h"
#include "Cargo.h"
#include "CargoCollection.h"
#include "Falcon9Payload.h"
#include "Falcon9Propulsion.h"

class Falcon9Builder : public RocketBuilder
{
    private:

        Rocket * product;
        CargoCollection * c;

    public:

        Falcon9Builder(CargoCollection * c);

        ~Falcon9Builder();

        void reset();

        void addPayload();

        void addPropulsion();

        Rocket * getRocket();

};

#endif //DERIVED_FALCON9BUILDER_H
```

5.17 Falcon9Core.h

```
#ifndef __Falcon9Core_h__
#define __Falcon9Core_h__

#include <vector>
#include <iostream>
#include "Core.h"
#include "MerlinEngine.h"

using namespace std;

class Falcon9Core: public Core
{
    private:

        string name;
        vector< MerlinEngine* > coreEngines;

    public:

        Falcon9Core(string name);

        ~Falcon9Core();

        string getName();

};

#endif

```

5.18 Falcon9Payload.h

```
#ifndef DERIVED_FALCON9PAYLOAD_H
#define DERIVED_FALCON9PAYLOAD_H

#include <iostream>
#include <vector>
#include "Cargo.h"
#include "Payload.h"
#include "Falcon9Core.h"
#include "MerlinEngine.h"
#include "Dragon.h"

using namespace std;

class Falcon9Payload : public Payload
{
    public:

        Falcon9Payload(SpaceCraft * s, VacuumMerlinEngine * vme);

        ~Falcon9Payload();

        void launch();
};

#endif //DERIVED_FALCON9PAYLOAD_H
```

5.19 Falcon9Propulsion.h

```
#ifndef DERIVED_FALCON9PROPULSION_H
#define DERIVED_FALCON9PROPULSION_H

#include "Payload.h"
#include "Propulsion.h"
#include "PropulsionMemento.h"
#include "Falcon9Core.h"

class Falcon9Propulsion : public Propulsion
{
    private:

        Falcon9Core * core1;
        Payload * payload;

    public:

        Falcon9Propulsion(Payload * p);

        ~Falcon9Propulsion();

        PropulsionMemento* backup ();

        void restore(PropulsionMemento * );

        void land();

        void launch();
};

#endif //DERIVED_FALCON9PROPULSION_H
```

5.20 FalconHeavy.h

```
#ifndef __FalconHeavy_h__
#define __FalconHeavy_h__

#include "Rocket.h"

class FalconHeavy: public Rocket
{
    public:

        void nextstage();

        FalconHeavy(Payload * payload);

        ~FalconHeavy();

        string getName();

};

#endif
```

5.21 FalconHeavyBuilder.h

```
#ifndef DERIVED_FALCONHEAVYBUILDER_H
#define DERIVED_FALCONHEAVYBUILDER_H

#include <iostream>
#include <vector>
#include "Human.h"
#include "Cargo.h"
#include "HumanCollection.h"
#include "CargoCollection.h"
#include "RocketBuilder.h"
#include "Payload.h"
#include "FalconHeavy.h"
#include "FalconHeavyPayload.h"
#include "Propulsion.h"
#include "FalconHeavyPropulsion.h"


class FalconHeavyBuilder : public RocketBuilder
{
    private:

        FalconHeavy * product;
        CargoCollection * c;
        HumanCollection * h;

    public:

        FalconHeavyBuilder(CargoCollection * c, HumanCollection * h);

        ~FalconHeavyBuilder();

        void reset();

        void addPayload();

        void addPropulsion();

        FalconHeavy* getRocket();
};

#endif //DERIVED_FALCONHEAVYBUILDER_H
```

5.22 FalconHeavyPayload.h

```
#ifndef DERIVED_FALCONHEAVYPAYLOAD_H
#define DERIVED_FALCONHEAVYPAYLOAD_H

#include "Payload.h"
#include "CrewDragon.h"
#include "SpaceCraft.h"
#include "VacuumMerlinEngine.h"
#include "Falcon9Core.h"

class FalconHeavyPayload : public Payload
{
    public:

        FalconHeavyPayload(SpaceCraft * s, VacuumMerlinEngine * vme);

        ~FalconHeavyPayload();

        void launch();

};

#endif //DERIVED_FALCONHEAVYPAYLOAD_H
```

5.23 FalconHeavyPropulsion.h

```
#ifndef DERIVED_FALCONHEAVYPROPULSION_H
#define DERIVED_FALCONHEAVYPROPULSION_H

#include "Payload.h"
#include "Propulsion.h"
#include "Falcon9Core.h"
#include "FalconHeavyPropulsionMemento.h"

class FalconHeavyPropulsion : public Propulsion
{
    private:

        Falcon9Core * core1;
        Falcon9Core * core2;
        Falcon9Core * core3;
        Payload * payload;

    public:

        FalconHeavyPropulsion(Payload * p);

        ~FalconHeavyPropulsion();

        FalconHeavyPropulsionMemento* backup();

        void restore(FalconHeavyPropulsionMemento * );

        void land();

        void launch();

};

#endif //DERIVED_FALCONHEAVYPROPULSION_H
```

5.24 FalconHeavyPropulsionMemento.h

```
#ifndef FALCONHEAVYPROPULSIONMEMENTO_H
#define FALCONHEAVYPROPULSIONMEMENTO_H

#include <iostream>
#include "Payload.h"
#include "Falcon9Core.h"

class FalconHeavyPropulsionMemento
{
    private:

        Falcon9Core * core1;
        Falcon9Core * core2;
        Falcon9Core * core3;
        Payload * payload;

    public:

        FalconHeavyPropulsionMemento(Falcon9Core * , Falcon9Core * , Falcon9Core * , Payload *);

        Falcon9Core * getCore1();

        Falcon9Core * getCore2();

        Falcon9Core * getCore3();

        Payload * getPay();
};
#endif
```

5.25 FHCaretaker.h

```
#ifndef FHCARETAKER_H
#define FHCARETAKER_H

#include "FalconHeavyPropulsionMemento.h"

class HPCaretaker
{
    private:

        FalconHeavyPropulsionMemento * backup;

    public:

        HPCaretaker();

        FalconHeavyPropulsionMemento * getBackup();

        void setBackup(FalconHeavyPropulsionMemento*);

};
#endif
```

5.26 FHStage1.h

```
#ifndef DERIVED_FHSTAGE1_H
#define DERIVED_FHSTAGE1_H

#include "RocketState.h"

class Rocket;

class FHStage1: public RocketState
{
    public:

        void handleChange(Rocket* r);

        FHStage1();

        ~FHStage1();
};


#endif //DERIVED_FHSTAGE1_H
```

5.27 FHStage2.h

```
#ifndef DERIVED_FHSTAGE2_H
#define DERIVED_FHSTAGE2_H

#include "RocketState.h"

class Rocket;

class FHStage2: public RocketState
{
    public:

        void handleChange(Rocket* r);

        FHStage2();

        ~FHStage2();
};


#endif //DERIVED_FHSTAGE2_H
```

5.28 GroundMissionControl.h

```
#ifndef __GroundMissionControl_h__
#define __GroundMissionControl_h__

#include <vector>
#include "Satellite.h"

using namespace std;

class GroundMissionControl
{
    private:

        vector<Satellite*> satelliteList;

    public:

        GroundMissionControl();

        void attach(Satellite*);

        void detach(Satellite*);

        void notify();

        ~GroundMissionControl();

};

#endif
```

5.29 Human.h

```
#ifndef __Human_h__
#define __Human_h__

#include <string>
#include "VectorOfHumans.h"

using namespace std;

class Human
{
    private:

        string name;
        double weight;
        string role;

    public:

        string getName();

        double getWeight();

        string getRole();

        Human(string name, double weight, string role);

        ~Human();
};
#endif
```

5.30 HumanCollection.h

```
#ifndef __HumanCollection_h__
#define __HumanCollection_h__

class Human;
class HumanIterator;
class HumanCollection;

class HumanCollection
{
    public:

        HumanCollection();

        virtual HumanIterator* createHumanIterator()=0;

        virtual void addHuman(Human*)=0;

        virtual void removeHuman(Human*)=0;

        virtual bool isEmpty()=0;

        virtual ~HumanCollection();

};

#endif
```

5.31 HumanFactory.h

```
#ifndef HUMANFACTORY_H
#define HUMANFACTORY_H

#include "Human.h"
#include "Colours.h"

class HumanFactory
{
    private:

    public:

        HumanFactory();

        ~HumanFactory();

        Human* createHuman(string name, double weight, string role);
};


#endif
```

5.32 HumanIterator.h

```
#ifndef __HumanIterator_h__
#define __HumanIterator_h__

#include "HumanCollection.h"

class Human;
class HumanIterator;

class HumanIterator
{
    public:

        HumanIterator();

        virtual Human* first()=0;

        virtual Human* next()=0;

        virtual bool hasNext()=0;

        virtual Human* current()=0;

        virtual ~HumanIterator();

};

#endif
```

5.33 Iterator.h

```
#ifndef PROJECT_ITERATOR_H
#define PROJECT_ITERATOR_H

class Iterator {

};

#endif //PROJECT_ITERATOR_H
```

5.34 LaunchMode.h

```
#ifndef __LaunchMode_h__
#define __LaunchMode_h__

#include "Simulation.h"

class LaunchMode
{
    private:

        Simulation * receiver;

    public:

        LaunchMode(Simulation * obj);

        void execute();

        Simulation * getR();

};

#endif
```

5.35 MerlinEngine.h

```
#ifndef __MerlinEngine_h__
#define __MerlinEngine_h__

#include <iostream>
#include "Engine.h"

class MerlinEngine : public Engine
{
    private:

        string name;

    public:

        MerlinEngine(string name);

        MerlinEngine * clone();

        ~MerlinEngine();
};

#endif
```

5.36 OperationsIteratorCargo.h

```
#ifndef DERIVED_OPERATIONSITERATORCARGO_H
#define DERIVED_OPERATIONSITERATORCARGO_H

#include <iostream>
#include <vector>
#include "Cargo.h"

using namespace std;

class Cargo;
class CargoIterator;

class OperationsIteratorCargo: public CargoIterator
{
    private:

        int currentpos{};
        vector<Cargo*> itlist;

    public:

        explicit OperationsIteratorCargo(vector<Cargo*> vect);

        Cargo* first() override;

        Cargo* next() override;

        bool hasNext() override;

        Cargo* current() override;

        ~OperationsIteratorCargo() override=default;

};

#endif //DERIVED_OPERATIONSITERATOR_H
```

5.37 OperationsIteratorHumans.h

```
#ifndef DERIVED_OPERATIONSITERATORHUMANS_H
#define DERIVED_OPERATIONSITERATORHUMANS_H

#include <vector>
#include "HumanIterator.h"

using namespace std;

class Human;

class OperationsIteratorHumans : public HumanIterator
{
    private:

        int currentpos;
        vector<Human*> itlist;

    public:

        explicit OperationsIteratorHumans(vector<Human*> vect);

        Human* first() override;

        Human* next() override;

        bool hasNext() override;

        Human* current() override;

        ~OperationsIteratorHumans() override=default;

};

#endif //DERIVED_OPERATIONSITERATORHUMANS_H
```

5.38 Payload.h

```
#ifndef DERIVED_PAYLOAD_H
#define DERIVED_PAYLOAD_H

#include <iostream>
#include <vector>
#include "VacuumMerlinEngine.h"
#include "SpaceCraft.h"

using namespace std;


class Payload
{

    private:

        VacuumMerlinEngine * vacmerlinengine;
        SpaceCraft * spacecraft;

    public:

        VacuumMerlinEngine * getVacuumMerlinEngine();

        void setVacuumMerlinEngine(VacuumMerlinEngine * v);

        SpaceCraft * getSpaceCraft();

        void setSpaceCraft(SpaceCraft * s);

        void print();

        Payload(SpaceCraft * s, VacuumMerlinEngine * v);

        Payload();

        virtual ~Payload();

        bool testVacuumMerlinEngine();

        virtual void launch() = 0;

};

#endif //DERIVED_PAYLOAD_H
```

5.39 PropCaretaker.h

```
#ifndef PROPCARETAKER_H
#define PROPCARETAKER_H

#include "PropulsionMemento.h"

class PropCaretaker
{
    private:

        PropulsionMemento * backup;

    public:

        PropCaretaker();

        PropulsionMemento * getBackup();

        void setBackup(PropulsionMemento* );
};
#endif
```

5.40 Propulsion.h

```
#ifndef DERIVED_PROPULSION_H
#define DERIVED_PROPULSION_H

#include "Payload.h"

class Propulsion : public Payload
{

    private:

    public:

        Propulsion();

        virtual ~Propulsion();

        virtual void land()=0;

        bool testPropulsion();

};

#endif //DERIVED_PROPULSION_H
```

5.41 PropulsionMemento.h

```
#ifndef PROPULSIONMEMENTO_H
#define PROPULSIONMEMENTO_H

#include <iostream>
#include "Falcon9Core.h"
#include "Payload.h"

class PropulsionMemento
{
    private:

        Falcon9Core * core ;
        Payload * payload;

    public:

        PropulsionMemento(Falcon9Core * core, Payload * pay);

        Falcon9Core * getCore();

        Payload * getPay();

};
#endif
```

5.42 Rocket.h

```
#ifndef __Rocket_h__
#define __Rocket_h__

#include "Payload.h"
#include "RocketState.h"
#include "Propulsion.h"

using namespace std;

class Rocket
{

    private:

        bool readytolaunch;
        bool hasbeentested;
        string name;

    protected:

        Payload * payload;
        RocketState* stage;
        Propulsion * propulsion;

    public:

        Rocket(Payload * payload);

        virtual ~Rocket();

        virtual void nextstage()=0;

        void setPayload(Payload * p);

        void print();

        Payload * getPayload();

        void setState(RocketState *state);

        void setPropulsion(Propulsion * p);

        void launch();

        void testRocket();

        bool testPropulsion();

        bool testVacuumMerlinEngine();

        void setName(string name);

        virtual string getName()=0;

        void testThatFails();
};

#endif
```

5.43 RocketBuilder.h

```
#ifndef DERIVED_ROCKETBUILDER_H
#define DERIVED_ROCKETBUILDER_H

#include "Rocket.h"

class RocketBuilder
{
    public:

        RocketBuilder();

        virtual ~RocketBuilder();

        virtual void reset() = 0;

        virtual void addPayload() = 0;

        virtual void addPropulsion() = 0;

        virtual Rocket * getRocket() = 0;

};

#endif //DERIVED_ROCKETBUILDER_H
```

5.44 RocketState.h

```
#ifndef DERIVED_ROCKETSTATE_H
#define DERIVED_ROCKETSTATE_H

class Rocket;

class RocketState
{
    private:

    public:

        virtual void handleChange(Rocket* r)=0;

        RocketState();

        virtual ~RocketState();

};

#endif //DERIVED_ROCKETSTATE_H
```

5.45 Satellite.h

```
#ifndef __Satellite_h__
#define __Satellite_h__

#include <iostream>
#include <string>
#include "VacuumMerlinEngine.h"
#include "SatelliteTransmission.h"
#include "Cargo.h"

class SatelliteTransmission;

using namespace std;

class Satellite : public Cargo
{
    private:

        VacuumMerlinEngine* vacuumMerlinEngine;
        SatelliteTransmission* mediator;
        string name;

    public:

        Satellite();

        void ConnectionChanged();

        virtual string getConnection() = 0;

        virtual void setConnection(string) = 0;

        virtual void update() = 0;

        virtual void print() = 0;

        ~Satellite();

};

#endif
```

5.46 SatelliteTransmission.h

```
#ifndef SATELLITETRANSMISSION_H
#define SATELLITETRANSMISSION_H
```

```

#include <iostream>
#include "Satellite.h"

class Satellite;

using namespace std;
//Mediator - Controls Communication between Satellites
class SatelliteTransmission
{
    private:

    public:

        SatelliteTransmission();

        virtual void notify(Satellite*) = 0;

        ~SatelliteTransmission();

};

#endif
```

5.47 SatelliteTransmissionController.h

```
#ifndef SATELLITETRANSMISSIONCONTROLLER_H
#define SATELLITETRANSMISSIONCONTROLLER_H

#include <iostream>
#include "SatelliteTransmission.h"

using namespace std;

class SatelliteTransmissionController : public SatelliteTransmission
{
    private:

    public:

        SatelliteTransmissionController();

        void notify(Satellite*);

        ~SatelliteTransmissionController();

};

#endif
```

5.48 Simulation.h

```
#ifndef __Simulation_h__
#define __Simulation_h__

#include "AllHeaders.h"

class Simulation
{
   private:

        HumanCollection * voh ; //vector of humans.
        CargoCollection * voc ; //vecotre of cargor.
        ConcreteGroundMissionControl * groundMissionControl;
        Rocket* dogeToMoon;
        int send;

    public:

        Simulation();

        ~Simulation();

        void setup();

        void liftOff();

        void staticFire();

        void testThatFails();

};

#endif
```

5.49 SpaceCraft.h

```
#ifndef __SpaceCraft_h__
#define __SpaceCraft_h__

#include <string>
#include "VacuumMerlinEngine.h"
#include "Human.h"
#include "CargoCollection.h"

class SpaceCraft;

class SpaceCraft
{
    private:

        static int counter;
        int id;
        string mission;
        Human* captain;
        CargoCollection * c;

    public:

        SpaceCraft(CargoCollection * c);

        virtual ~SpaceCraft();

        void setCargoCollection(CargoCollection * c);

        CargoCollection * getCargoCollection();

        virtual void printSpaceCraft()=0; //to be re-impliment in the CrewDragon
};

#endif
```

5.50 SpaceCraftFactory.h

```
#ifndef __SpaceCraftFactory_h__
#define __SpaceCraftFactory_h__

class SpaceCraftFactory;

class SpaceCraftFactory
{

};

#endif
```

5.51 StarLinkSatellite.h

```
#ifndef __StarLinkSatellite_h__
#define __StarLinkSatellite_h__

#include <iostream>
//#include "Colours.h"
#include "Satellite.h"
#include "ConcreteGroundMissionControl.h"

using namespace std;

// class ConcreteGroundMissionControl;

class StarLinkSatellite: public Satellite
{
    private:

        ConcreteGroundMissionControl* missionControl;
        bool communicating;
        string connection;
        string name;

    public:

        StarLinkSatellite(ConcreteGroundMissionControl*);

        StarLinkSatellite(string);

        StarLinkSatellite * clone();

        string getName();

        void setName(string);

        void update();

        void print();

        string getConnection();

        void setConnection(string);

        ~StarLinkSatellite();

};

#endif
```

5.52 VacuumMerlinEngine.h

```
#ifndef __VacuumMerlinEngine_h__
#define __VacuumMerlinEngine_h__

#include "Engine.h"

// class Engine;
class VacuumMerlinEngine;

class VacuumMerlinEngine: public Engine
{

    private:

        static int counter;
        int id;

    public:

        VacuumMerlinEngine();

        VacuumMerlinEngine * clone();

        ~VacuumMerlinEngine();

};

#endif
```

5.53 VectorOfCargo.h

```
#ifndef DERIVED_VECTOROFCARGO_H
#define DERIVED_VECTOROFCARGO_H

#include <iostream>
#include <vector>
#include "CargoIterator.h"
#include "OperationsIteratorCargo.h"


using namespace std;

class CargoCollection;

class VectorOfCargo: public CargoCollection
{
    private:

        vector<Cargo*>cargo;

    public:

        VectorOfCargo();

        CargoIterator* createCargoIterator() override;

        void addCargo(Cargo*) override;

        void removeCargo(Cargo*) override;

        bool isEmpty() override;

        ~VectorOfCargo() ;

};

#endif //DERIVED_VECTOROFCARGO_H
```

5.54 VectorOfHumans.h

```
#ifndef DERIVED_VECTOROFHUMANS_H
#define DERIVED_VECTOROFHUMANS_H

#include <iostream>
#include <vector>
#include "HumanCollection.h"
#include "HumanIterator.h"
#include "OperationsIteratorHumans.h"

using namespace std;

class VectorOfHumans : public HumanCollection
{
    private:

        vector<Human*> people;

    public:

        VectorOfHumans();

        HumanIterator* createHumanIterator()override;

        void addHuman(Human*)override;

        void removeHuman(Human*) override;

        bool isEmpty() override;

        ~VectorOfHumans() override;

};

#endif //DERIVED_VECTOROFHUMANS_H
```

Index