

Derived

Generated by Doxygen 1.9.2

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Cargo Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 Cargo()	7
4.1.1.2 ~Cargo()	8
4.1.2 Member Function Documentation	8
4.1.2.1 getDescription()	8
4.1.2.2 getName()	8
4.1.2.3 getWeight()	9
4.2 CargoCollection Class Reference	9
4.2.1 Constructor & Destructor Documentation	9
4.2.1.1 CargoCollection()	10
4.2.1.2 ~CargoCollection()	10
4.2.2 Member Function Documentation	10
4.2.2.1 addCargo()	10
4.2.2.2 createCargoIterator()	10
4.2.2.3 isEmpty()	11
4.2.2.4 removeCargo()	11
4.3 CargoFactory Class Reference	11
4.3.1 Constructor & Destructor Documentation	12
4.3.1.1 CargoFactory()	12
4.3.1.2 ~CargoFactory()	12
4.3.2 Member Function Documentation	12
4.3.2.1 createCargo()	12
4.4 CargoIterator Class Reference	13
4.4.1 Constructor & Destructor Documentation	13
4.4.1.1 CargoIterator()	13
4.4.1.2 ~CargoIterator()	14
4.4.2 Member Function Documentation	14
4.4.2.1 current()	14
4.4.2.2 first()	14
4.4.2.3 hasNext()	15
4.4.2.4 next()	15
4.5 Colours Class Reference	15

4.5.1 Member Function Documentation	16
4.5.1.1 black()	16
4.5.1.2 blue()	16
4.5.1.3 cyan()	17
4.5.1.4 green()	17
4.5.1.5 purple()	17
4.5.1.6 red()	18
4.5.1.7 yellow()	18
4.6 ConcreteGroundMissionControl Class Reference	19
4.6.1 Constructor & Destructor Documentation	19
4.6.1.1 ConcreteGroundMissionControl()	19
4.6.1.2 ~ConcreteGroundMissionControl()	19
4.6.2 Member Function Documentation	20
4.6.2.1 hasConnected()	20
4.6.2.2 setConnection()	20
4.7 ConfigurationManager Class Reference	20
4.7.1 Constructor & Destructor Documentation	21
4.7.1.1 ConfigurationManager()	21
4.7.1.2 ~ConfigurationManager()	21
4.7.2 Member Function Documentation	21
4.7.2.1 BuildAndDecorateRocket()	21
4.8 Core Class Reference	22
4.9 CrewDragon Class Reference	22
4.9.1 Constructor & Destructor Documentation	22
4.9.1.1 CrewDragon()	22
4.9.1.2 ~CrewDragon()	23
4.9.2 Member Function Documentation	23
4.9.2.1 printSpaceCraft()	23
4.10 Dragon Class Reference	23
4.10.1 Constructor & Destructor Documentation	23
4.10.1.1 Dragon()	24
4.10.1.2 ~Dragon()	24
4.10.2 Member Function Documentation	24
4.10.2.1 printSpaceCraft()	24
4.11 Engine Class Reference	25
4.11.1 Constructor & Destructor Documentation	25
4.11.1.1 Engine()	25
4.12 F9Stage1 Class Reference	25
4.12.1 Constructor & Destructor Documentation	25
4.12.1.1 F9Stage1()	26
4.12.1.2 ~F9Stage1()	26
4.12.2 Member Function Documentation	26

4.12.2.1 handleChange()	26
4.13 F9Stage2 Class Reference	27
4.13.1 Constructor & Destructor Documentation	27
4.13.1.1 F9Stage2()	27
4.13.1.2 ~F9Stage2()	27
4.13.2 Member Function Documentation	27
4.13.2.1 handleChange()	27
4.14 Falcon9 Class Reference	28
4.14.1 Constructor & Destructor Documentation	28
4.14.1.1 Falcon9()	28
4.14.1.2 ~Falcon9()	29
4.14.2 Member Function Documentation	29
4.14.2.1 getName()	29
4.14.2.2 nextstage()	29
4.15 Falcon9Builder Class Reference	29
4.15.1 Constructor & Destructor Documentation	30
4.15.1.1 Falcon9Builder()	30
4.15.1.2 ~Falcon9Builder()	30
4.15.2 Member Function Documentation	30
4.15.2.1 addPayload()	31
4.15.2.2 addPropulsion()	31
4.15.2.3 getRocket()	31
4.15.2.4 reset()	32
4.16 Falcon9Core Class Reference	32
4.16.1 Constructor & Destructor Documentation	32
4.16.1.1 Falcon9Core()	32
4.16.1.2 ~Falcon9Core()	33
4.16.2 Member Function Documentation	33
4.16.2.1 getName()	33
4.17 Falcon9Payload Class Reference	33
4.17.1 Constructor & Destructor Documentation	34
4.17.1.1 Falcon9Payload()	34
4.17.1.2 ~Falcon9Payload()	34
4.17.2 Member Function Documentation	34
4.17.2.1 launch()	35
4.18 Falcon9Propulsion Class Reference	35
4.18.1 Constructor & Destructor Documentation	35
4.18.1.1 Falcon9Propulsion()	35
4.18.1.2 ~Falcon9Propulsion()	36
4.18.2 Member Function Documentation	36
4.18.2.1 backup()	36
4.18.2.2 land()	36

4.18.2.3 launch()	37
4.18.2.4 restore()	37
4.19 FalconHeavy Class Reference	37
4.19.1 Constructor & Destructor Documentation	38
4.19.1.1 FalconHeavy()	38
4.19.1.2 ~FalconHeavy()	38
4.19.2 Member Function Documentation	38
4.19.2.1 getName()	38
4.19.2.2 nextstage()	39
4.20 FalconHeavyBuilder Class Reference	39
4.20.1 Constructor & Destructor Documentation	39
4.20.1.1 FalconHeavyBuilder()	39
4.20.1.2 ~FalconHeavyBuilder()	40
4.20.2 Member Function Documentation	40
4.20.2.1 addPayload()	40
4.20.2.2 addPropulsion()	40
4.20.2.3 getRocket()	41
4.20.2.4 reset()	41
4.21 FalconHeavyPayload Class Reference	41
4.21.1 Constructor & Destructor Documentation	42
4.21.1.1 FalconHeavyPayload()	42
4.21.1.2 ~FalconHeavyPayload()	42
4.21.2 Member Function Documentation	42
4.21.2.1 launch()	42
4.22 FalconHeavyPropulsion Class Reference	43
4.22.1 Constructor & Destructor Documentation	43
4.22.1.1 FalconHeavyPropulsion()	43
4.22.1.2 ~FalconHeavyPropulsion()	43
4.22.2 Member Function Documentation	44
4.22.2.1 backup()	44
4.22.2.2 land()	44
4.22.2.3 launch()	44
4.22.2.4 restore()	45
4.23 FalconHeavyPropulsionMemento Class Reference	45
4.23.1 Constructor & Destructor Documentation	45
4.23.1.1 FalconHeavyPropulsionMemento()	45
4.23.2 Member Function Documentation	46
4.23.2.1 getCore1()	46
4.23.2.2 getCore2()	46
4.23.2.3 getCore3()	46
4.23.2.4 getPay()	47
4.24 FHStage1 Class Reference	47

4.24.1 Constructor & Destructor Documentation	47
4.24.1.1 FHStage1()	47
4.24.1.2 ~FHStage1()	48
4.24.2 Member Function Documentation	48
4.24.2.1 handleChange()	48
4.25 FHStage2 Class Reference	48
4.25.1 Constructor & Destructor Documentation	49
4.25.1.1 FHStage2()	49
4.25.1.2 ~FHStage2()	49
4.25.2 Member Function Documentation	49
4.25.2.1 handleChange()	49
4.26 GroundMissionControl Class Reference	50
4.26.1 Constructor & Destructor Documentation	50
4.26.1.1 GroundMissionControl()	50
4.26.1.2 ~GroundMissionControl()	50
4.26.2 Member Function Documentation	51
4.26.2.1 attach()	51
4.26.2.2 detach()	51
4.26.2.3 notify()	51
4.27 HPCaretaker Class Reference	52
4.27.1 Constructor & Destructor Documentation	52
4.27.1.1 HPCaretaker()	52
4.27.2 Member Function Documentation	52
4.27.2.1 getBackup()	52
4.27.2.2 setBackup()	53
4.28 Human Class Reference	53
4.28.1 Constructor & Destructor Documentation	53
4.28.1.1 Human()	53
4.28.1.2 ~Human()	54
4.28.2 Member Function Documentation	54
4.28.2.1 getName()	54
4.28.2.2 getRole()	54
4.28.2.3 getWeight()	55
4.29 HumanCollection Class Reference	55
4.29.1 Constructor & Destructor Documentation	55
4.29.1.1 HumanCollection()	56
4.29.1.2 ~HumanCollection()	56
4.29.2 Member Function Documentation	56
4.29.2.1 addHuman()	56
4.29.2.2 createHumanIterator()	56
4.29.2.3 isEmpty()	57
4.29.2.4 removeHuman()	57

4.30 HumanFactory Class Reference	57
4.30.1 Constructor & Destructor Documentation	58
4.30.1.1 HumanFactory()	58
4.30.1.2 ~HumanFactory()	58
4.30.2 Member Function Documentation	58
4.30.2.1 createHuman()	58
4.31 HumanIterator Class Reference	59
4.31.1 Constructor & Destructor Documentation	59
4.31.1.1 HumanIterator()	59
4.31.1.2 ~HumanIterator()	60
4.31.2 Member Function Documentation	60
4.31.2.1 current()	60
4.31.2.2 first()	60
4.31.2.3 hasNext()	61
4.31.2.4 next()	61
4.32 Iterator Class Reference	61
4.33 LaunchMode Class Reference	62
4.33.1 Constructor & Destructor Documentation	62
4.33.1.1 LaunchMode()	62
4.33.2 Member Function Documentation	62
4.33.2.1 execute()	62
4.33.2.2 getR()	63
4.34 MerlinEngine Class Reference	63
4.34.1 Constructor & Destructor Documentation	63
4.34.1.1 MerlinEngine()	63
4.34.1.2 ~MerlinEngine()	64
4.34.2 Member Function Documentation	64
4.34.2.1 clone()	64
4.35 OperationsIteratorCargo Class Reference	64
4.35.1 Constructor & Destructor Documentation	65
4.35.1.1 OperationsIteratorCargo()	65
4.35.1.2 ~OperationsIteratorCargo()	65
4.35.2 Member Function Documentation	65
4.35.2.1 current()	66
4.35.2.2 first()	66
4.35.2.3 hasNext()	66
4.35.2.4 next()	67
4.36 OperationsIteratorHumans Class Reference	67
4.36.1 Constructor & Destructor Documentation	67
4.36.1.1 OperationsIteratorHumans()	67
4.36.1.2 ~OperationsIteratorHumans()	68
4.36.2 Member Function Documentation	68

4.36.2.1 current()	68
4.36.2.2 first()	68
4.36.2.3 hasNext()	69
4.36.2.4 next()	69
4.37 Payload Class Reference	69
4.37.1 Constructor & Destructor Documentation	70
4.37.1.1 Payload() [1/2]	70
4.37.1.2 Payload() [2/2]	70
4.37.1.3 ~Payload()	71
4.37.2 Member Function Documentation	71
4.37.2.1 getSpaceCraft()	71
4.37.2.2 getVacuumMerlinEngine()	71
4.37.2.3 launch()	72
4.37.2.4 print()	72
4.37.2.5 setSpaceCraft()	72
4.37.2.6 setVacuumMerlinEngine()	72
4.37.2.7 testVacuumMerlinEngine()	73
4.38 PropCaretaker Class Reference	73
4.38.1 Constructor & Destructor Documentation	73
4.38.1.1 PropCaretaker()	74
4.38.2 Member Function Documentation	74
4.38.2.1 getBackup()	74
4.38.2.2 setBackup()	74
4.39 Propulsion Class Reference	75
4.39.1 Constructor & Destructor Documentation	75
4.39.1.1 Propulsion()	75
4.39.1.2 ~Propulsion()	75
4.39.2 Member Function Documentation	75
4.39.2.1 land()	76
4.39.2.2 testPropulsion()	76
4.40 PropulsionMemento Class Reference	76
4.40.1 Constructor & Destructor Documentation	76
4.40.1.1 PropulsionMemento()	76
4.40.2 Member Function Documentation	77
4.40.2.1 getCore()	77
4.40.2.2 getPay()	77
4.41 Rocket Class Reference	77
4.41.1 Constructor & Destructor Documentation	78
4.41.1.1 Rocket()	78
4.41.1.2 ~Rocket()	79
4.41.2 Member Function Documentation	79
4.41.2.1 getName()	79

4.41.2.2	getPayload()	79
4.41.2.3	launch()	80
4.41.2.4	nextstage()	80
4.41.2.5	print()	80
4.41.2.6	setName()	80
4.41.2.7	setPayload()	81
4.41.2.8	setPropulsion()	81
4.41.2.9	setState()	81
4.41.2.10	testPropulsion()	82
4.41.2.11	testRocket()	82
4.41.2.12	testThatFails()	82
4.41.2.13	testVacuumMerlinEngine()	83
4.42	RocketBuilder Class Reference	83
4.42.1	Constructor & Destructor Documentation	83
4.42.1.1	RocketBuilder()	84
4.42.1.2	~RocketBuilder()	84
4.42.2	Member Function Documentation	84
4.42.2.1	addPayload()	84
4.42.2.2	addPropulsion()	84
4.42.2.3	getRocket()	85
4.42.2.4	reset()	85
4.43	RocketState Class Reference	85
4.43.1	Constructor & Destructor Documentation	86
4.43.1.1	RocketState()	86
4.43.1.2	~RocketState()	86
4.43.2	Member Function Documentation	86
4.43.2.1	handleChange()	86
4.44	Satellite Class Reference	87
4.44.1	Constructor & Destructor Documentation	87
4.44.1.1	Satellite()	87
4.44.1.2	~Satellite()	88
4.44.2	Member Function Documentation	88
4.44.2.1	ConnectionChanged()	88
4.44.2.2	getConnection()	88
4.44.2.3	print()	89
4.44.2.4	setConnection()	89
4.44.2.5	update()	89
4.45	SatelliteTransmission Class Reference	89
4.45.1	Constructor & Destructor Documentation	90
4.45.1.1	SatelliteTransmission()	90
4.45.1.2	~SatelliteTransmission()	90
4.45.2	Member Function Documentation	90

4.45.2.1 notify()	90
4.46 SatelliteTransmissionController Class Reference	91
4.46.1 Constructor & Destructor Documentation	91
4.46.1.1 SatelliteTransmissionController()	91
4.46.1.2 ~SatelliteTransmissionController()	91
4.46.2 Member Function Documentation	91
4.46.2.1 notify()	92
4.47 Simulation Class Reference	92
4.47.1 Constructor & Destructor Documentation	92
4.47.1.1 Simulation()	92
4.47.1.2 ~Simulation()	93
4.47.2 Member Function Documentation	93
4.47.2.1 liftOff()	93
4.47.2.2 setup()	93
4.47.2.3 staticFire()	93
4.47.2.4 testThatFails()	94
4.48 SpaceCraft Class Reference	94
4.48.1 Constructor & Destructor Documentation	94
4.48.1.1 SpaceCraft()	94
4.48.1.2 ~SpaceCraft()	95
4.48.2 Member Function Documentation	95
4.48.2.1 getCargoCollection()	95
4.48.2.2 printSpaceCraft()	95
4.48.2.3 setCargoCollection()	95
4.49 SpaceCraftFactory Class Reference	96
4.50 StarLinkSatellite Class Reference	96
4.50.1 Constructor & Destructor Documentation	96
4.50.1.1 StarLinkSatellite() [1/2]	97
4.50.1.2 StarLinkSatellite() [2/2]	97
4.50.1.3 ~StarLinkSatellite()	97
4.50.2 Member Function Documentation	97
4.50.2.1 clone()	97
4.50.2.2 getConnection()	98
4.50.2.3 getName()	98
4.50.2.4 print()	98
4.50.2.5 setConnection()	99
4.50.2.6 setName()	99
4.50.2.7 update()	99
4.51 VacuumMerlinEngine Class Reference	99
4.51.1 Constructor & Destructor Documentation	100
4.51.1.1 VacuumMerlinEngine()	100
4.51.1.2 ~VacuumMerlinEngine()	100

4.51.2 Member Function Documentation	100
4.51.2.1 clone()	100
4.52 VectorOfCargo Class Reference	101
4.52.1 Constructor & Destructor Documentation	101
4.52.1.1 VectorOfCargo()	101
4.52.1.2 ~VectorOfCargo()	101
4.52.2 Member Function Documentation	102
4.52.2.1 addCargo()	102
4.52.2.2 createCargoIterator()	102
4.52.2.3 isEmpty()	102
4.52.2.4 removeCargo()	103
4.53 VectorOfHumans Class Reference	103
4.53.1 Constructor & Destructor Documentation	103
4.53.1.1 VectorOfHumans()	103
4.53.1.2 ~VectorOfHumans()	104
4.53.2 Member Function Documentation	104
4.53.2.1 addHuman()	104
4.53.2.2 createHumanIterator()	104
4.53.2.3 isEmpty()	105
4.53.2.4 removeHuman()	105
5 File Documentation	107
5.1 AllHeaders.h	107
5.2 Cargo.h	107
5.3 CargoCollection.h	108
5.4 CargoFactory.h	108
5.5 CargoIterator.h	109
5.6 Colours.h	109
5.7 ConcreteGroundMissionControl.h	110
5.8 ConfigurationManager.h	110
5.9 Core.h	110
5.10 CrewDragon.h	111
5.11 Dragon.h	111
5.12 Engine.h	111
5.13 F9Stage1.h	112
5.14 F9Stage2.h	112
5.15 Falcon9.h	112
5.16 Falcon9Builder.h	113
5.17 Falcon9Core.h	113
5.18 Falcon9Payload.h	114
5.19 Falcon9Propulsion.h	114
5.20 FalconHeavy.h	114

5.21 FalconHeavyBuilder.h	115
5.22 FalconHeavyPayload.h	115
5.23 FalconHeavyPropulsion.h	116
5.24 FalconHeavyPropulsionMemento.h	116
5.25 FHCaretaker.h	116
5.26 FHStage1.h	117
5.27 FHStage2.h	117
5.28 GroundMissionControl.h	117
5.29 Human.h	118
5.30 HumanCollection.h	118
5.31 HumanFactory.h	119
5.32 HumanIterator.h	119
5.33 Iterator.h	119
5.34 LaunchMode.h	119
5.35 MerlinEngine.h	120
5.36 OperationsIteratorCargo.h	120
5.37 OperationsIteratorHumans.h	121
5.38 Payload.h	121
5.39 PropCaretaker.h	122
5.40 Propulsion.h	122
5.41 PropulsionMemento.h	122
5.42 Rocket.h	123
5.43 RocketBuilder.h	123
5.44 RocketState.h	124
5.45 Satellite.h	124
5.46 SatelliteTransmission.h	124
5.47 SatelliteTransmissionController.h	125
5.48 Simulation.h	125
5.49 SpaceCraft.h	126
5.50 SpaceCraftFactory.h	126
5.51 StarLinkSatellite.h	126
5.52 VacuumMerlinEngine.h	127
5.53 VectorOfCargo.h	127
5.54 VectorOfHumans.h	128

Index

129

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Cargo	7
Satellite	87
StarLinkSatellite	96
CargoCollection	9
VectorOfCargo	101
CargoFactory	11
CargoIterator	13
OperationsIteratorCargo	64
Colours	15
ConfigurationManager	20
Core	22
Falcon9Core	32
Engine	25
MerlinEngine	63
VacuumMerlinEngine	99
FalconHeavyPropulsionMemento	45
GroundMissionControl	50
ConcreteGroundMissionControl	19
HPCCaretaker	52
Human	53
HumanCollection	55
VectorOfHumans	103
HumanFactory	57
HumanIterator	59
OperationsIteratorHumans	67
Iterator	61
LaunchMode	62
Payload	69
Falcon9Payload	33
FalconHeavyPayload	41
Propulsion	75
Falcon9Propulsion	35
FalconHeavyPropulsion	43

PropCaretaker	73
PropulsionMemento	76
Rocket	77
Falcon9	28
FalconHeavy	37
RocketBuilder	83
Falcon9Builder	29
FalconHeavyBuilder	39
RocketState	85
F9Stage1	25
F9Stage2	27
FHStage1	47
FHStage2	48
SatelliteTransmission	89
SatelliteTransmissionController	91
Simulation	92
SpaceCraft	94
CrewDragon	22
Dragon	23
SpaceCraftFactory	96

Chapter 2

Class Index

2.1 Class List

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

Cargo	7
CargoCollection	9
CargoFactory	11
CargoIterator	13
Colours	15
ConcreteGroundMissionControl	19
ConfigurationManager	20
Core	22
CrewDragon	22
Dragon	23
Engine	25
F9Stage1	25
F9Stage2	27
Falcon9	28
Falcon9Builder	29
Falcon9Core	32
Falcon9Payload	33
Falcon9Propulsion	35
FalconHeavy	37
FalconHeavyBuilder	39
FalconHeavyPayload	41
FalconHeavyPropulsion	43
FalconHeavyPropulsionMemento	45
FHStage1	47
FHStage2	48
GroundMissionControl	50
HCaretaker	52
Human	53
HumanCollection	55
HumanFactory	57
HumanIterator	59
Iterator	61
LaunchMode	62
MerlinEngine	63
OperationsIteratorCargo	64

OperationsIteratorHumans	67
Payload	69
PropCaretaker	73
Propulsion	75
PropulsionMemento	76
Rocket	77
RocketBuilder	83
RocketState	85
Satellite	87
SatelliteTransmission	89
SatelliteTransmissionController	91
Simulation	92
SpaceCraft	94
SpaceCraftFactory	96
StarLinkSatellite	96
VacuumMerlinEngine	99
VectorOfCargo	101
VectorOfHumans	103

Chapter 3

File Index

3.1 File List

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

AllHeaders.h	??
Cargo.h	??
CargoCollection.h	??
CargoFactory.h	??
CargoIterator.h	??
Colours.h	??
ConcreteGroundMissionControl.h	??
ConfigurationManager.h	??
Core.h	??
CrewDragon.h	??
Dragon.h	??
Engine.h	??
F9Stage1.h	??
F9Stage2.h	??
Falcon9.h	??
Falcon9Builder.h	??
Falcon9Core.h	??
Falcon9Payload.h	??
Falcon9Propulsion.h	??
FalconHeavy.h	??
FalconHeavyBuilder.h	??
FalconHeavyPayload.h	??
FalconHeavyPropulsion.h	??
FalconHeavyPropulsionMemento.h	??
FHCaretaker.h	??
FHStage1.h	??
FHStage2.h	??
GroundMissionControl.h	??
Human.h	??
HumanCollection.h	??
HumanFactory.h	??
HumanIterator.h	??
Iterator.h	??
LaunchMode.h	??
MerlinEngine.h	??

OperationsIteratorCargo.h	??
OperationsIteratorHumans.h	??
Payload.h	??
PropCaretaker.h	??
Propulsion.h	??
PropulsionMemento.h	??
Rocket.h	??
RocketBuilder.h	??
RocketState.h	??
Satellite.h	??
SatelliteTransmission.h	??
SatelliteTransmissionController.h	??
Simulation.h	??
SpaceCraft.h	??
SpaceCraftFactory.h	??
StarLinkSatellite.h	??
VacuumMerlinEngine.h	??
VectorOfCargo.h	??
VectorOfHumans.h	??

Chapter 4

Class Documentation

4.1 Cargo Class Reference

Inherited by **Satellite**.

Public Member Functions

- **Cargo** (string name, double weight, string desc)
*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 desc )
```

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

Author

Derived

Parameters

<i>name</i>	
<i>weight</i>	
<i>descp</i>	

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 **Cargolterator** * **createCargolterator** ()=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

<i>name</i>	
<i>weight</i>	
<i>description</i>	

Returns

Cargo*

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

- CargoFactory.h
- CargoFactory.cpp

4.4 Cargolterator Class Reference

Inherited by **OperationsIteatorCargo**.

Public Member Functions

- **Cargolterator** ()=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 ~**Cargolterator** ()=default
*Destroy the **Cargo** (p. 7) **Iterator** (p. 61) object.*

4.4.1 Constructor & Destructor Documentation

4.4.1.1 Cargolterator()

```
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

<i>s</i>	
----------	--

Returns

string

4.5.1.2 `blue()`

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

Changes the Colour of the String.

Author

Derived

Parameters

<i>s</i>	
----------	--

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

<i>s</i>	
----------	--

Returns

string

4.5.1.6 red()

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

Changes the Colour of the String.

Author

Derived

Parameters

<i>s</i>	
----------	--

Returns

string

4.5.1.7 yellow()

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

Changes the Colour of the String.

Author

Derived

Parameters

<i>s</i>	
----------	--

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

<i>c</i>	
<i>h</i>	

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

<i>c</i>	
<i>h</i>	

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

<i>r</i>	
----------	--

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

<i>r</i>	
----------	--

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

<i>name</i>	
-------------	--

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

<i>s</i>	
<i>vme</i>	

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

<i>payload</i>	
----------------	--

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

<i>c</i>	
<i>h</i>	

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 Payload 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

<i>s</i>	
<i>vme</i>	

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

<i>p</i>	
----------	--

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

<i>r</i>	
----------	--

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

<i>r</i>	
----------	--

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

<i>name</i>	
<i>weight</i>	
<i>role</i>	

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

<i>name</i>	
<i>weight</i>	
<i>role</i>	

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

<i>obj</i>	
------------	--

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

<i>name</i>	
-------------	--

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

<i>vect</i>	
-------------	--

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 **Cargolterator** (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 **Cargolterator** (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 **Cargolterator** (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

<i>vect</i>	
-------------	--

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

<i>core</i>	
<i>pay</i>	

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

<i>payload</i>	
----------------	--

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

<i>name</i>	
-------------	--

4.41.2.7 setPayload()

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

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

Author

Derived

Parameters

<i>p</i>	
----------	--

4.41.2.8 setPropulsion()

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

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

Author

Derived

Parameters

<i>p</i>	
----------	--

4.41.2.9 setState()

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

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

Author

Derived

Parameters

<i>state</i>	
--------------	--

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) object.*

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) object.

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

<i>r</i>	
----------	--

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** ()
*Notify **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 ( )
```

Notify **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

<i>c</i>	
----------	--

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

```
1 #include "Cargo.h"
2 #include "CargoCollection.h"
3 #include "CargoIterator.h"
4 #include "Colours.h"
5 #include "ConcreteGroundMissionControl.h"
6 #include "ConfigurationManager.h"
7 #include "Core.h"
8 #include "CrewDragon.h"
9 #include "Dragon.h"
10 #include "Engine.h"
11 #include "F9Stage1.h"
12 #include "F9Stage2.h"
13 #include "FHStage1.h"
14 #include "FHStage2.h"
15 #include "Falcon9.h"
16 #include "Falcon9Builder.h"
17 #include "Falcon9Core.h"
18 #include "Falcon9Payload.h"
19 #include "Falcon9Propulsion.h"
20 #include "FalconHeavy.h"
21 #include "FalconHeavyBuilder.h"
22 #include "FalconHeavyPayload.h"
23 #include "FalconHeavyPropulsion.h"
24 #include "GroundMissionControl.h"
25 #include "Human.h"
26 #include "HumanCollection.h"
27 #include "HumanIterator.h"
28 #include "Iterator.h"
29 #include "MerlinEngine.h"
30 #include "OperationsIteratorCargo.h"
31 #include "OperationsIteratorHumans.h"
32 #include "Payload.h"
33 #include "Propulsion.h"
34 #include "Rocket.h"
35 #include "RocketBuilder.h"
36 #include "RocketState.h"
37 #include "Satellite.h"
38 #include "SpaceCraft.h"
39 #include "SpaceCraftFactory.h"
40 #include "StarLinkSatellite.h"
41 #include "VacuumMerlinEngine.h"
42 #include "VectorOfCargo.h"
43 #include "VectorOfHumans.h"
```

5.2 Cargo.h

```
1 #ifndef __Cargo_h__
2 #define __Cargo_h__
3
4 #include <string>
5 #include "VectorOfCargo.h"
6
7 using namespace std;
8
```

```
9 class Cargo
10 {
11     private:
12
13         string name;
14         double weight;
15         string description;
16
17     public:
18
19         Cargo (string name, double weight, string desc);
20
21         string getName();
22
23         string getDescription();
24
25         double getWeight();
26
27         ~Cargo();
28 };
29 #endif
```

5.3 CargoCollection.h

```
1 #ifndef __CargoCollection_h__
2 #define __CargoCollection_h__
3
4 class Cargo;
5 class CargoIterator;
6
7 class CargoCollection
8 {
9     public:
10
11         CargoCollection() = default;
12
13         virtual CargoIterator* createCargoIterator()=0;
14
15         virtual void addCargo(Cargo*)=0;
16
17         virtual void removeCargo(Cargo*)=0;
18
19         virtual bool isEmpty()=0;
20
21         virtual ~CargoCollection()=default;
22 };
23 #endif
```

5.4 CargoFactory.h

```
1 #ifndef CARGOFACTORY_H
2 #define CARGOFACTORY_H
3
4 #include "Cargo.h"
5 #include "Colours.h"
6
7 class CargoFactory
8 {
9     private:
10
11     public:
12
13         CargoFactory();
14
15         ~CargoFactory();
16
17         Cargo* createCargo(string name, double weight, string description);
18 };
19 #endif
```

5.5 Cargolterator.h

```

1  #ifndef __CargoIterator_h__
2  #define __CargoIterator_h__
3
4  #include "CargoCollection.h"
5
6  class Cargo;
7
8  class CargoIterator
9  {
10     public:
11
16         CargoIterator()=default;
17
23         virtual Cargo* first()=0;
24
30         virtual Cargo* next()=0;
31
38         virtual bool hasNext()=0;
39
45         virtual Cargo* current()=0;
46
51         virtual ~CargoIterator()=default;
52
53 };
54
55 #endif

```

5.6 Colours.h

```

1  #ifndef COLOURS_H
2  #define COLOURS_H
3
4  #include <iostream>
5  #include <iomanip>
6  #include <ctime>
7
8  using namespace std;
9
10 /*
11 This is a class used for the output of code
12 Pass in the code string you want and the colour will be changed
13 Example use:
14     cout<<Colours::purple("hello World")<<endl;
15 The code above will print the text hello World in the colour purple.
16 */
17 class Colours
18 {
19     public:
20
27     static string black(string s) {
28         string output = "\033[30m" + s + "\033[0m";
29         return output;
30     }
31
38     static string red(string s) {
39         string output = "\033[31m" + s + "\033[0m";
40         return output;
41     }
42
49     static string green(string s) {
50         string output = "\033[32m" + s + "\033[0m";
51         return output;
52     }
53
60     static string yellow(string s) {
61         string output = "\033[33m" + s + "\033[0m";
62         return output;
63     }
64
71     static string blue(string s) {
72         string output = "\033[34m" + s + "\033[0m";
73         return output;
74     }
75
82     static string purple(string s) {
83         string output = "\033[35m" + s + "\033[0m";
84         return output;
85     }
86
93     static string cyan(string s) {
94         string output = "\033[36m" + s + "\033[0m";

```

```
95         return output;
96     }
97
98 };
99
100 #endif
```

5.7 ConcreteGroundMissionControl.h

```
1 #ifndef CONCRETEGROUNDMISSIONCONTROL_H
2 #define CONCRETEGROUNDMISSIONCONTROL_H
3
4 #include <iostream>
5 #include "GroundMissionControl.h"
6
7 using namespace std;
8
9 class ConcreteGroundMissionControl : public GroundMissionControl
10 {
11     private:
12
13         bool connection;
14
15     public:
16
17         ConcreteGroundMissionControl();
18
19         bool hasConnected();
20
21         void setConnection(bool);
22
23         ~ConcreteGroundMissionControl();
24 };
25
26 #endif
```

5.8 ConfigurationManager.h

```
1 #ifndef DERIVED_CONFIGURATIONMANAGER_H
2 #define DERIVED_CONFIGURATIONMANAGER_H
3
4 #include <iostream>
5 #include <vector>
6 #include "RocketBuilder.h"
7 #include "Human.h"
8 #include "Cargo.h"
9 #include "HumanCollection.h"
10 #include "CargoCollection.h"
11 #include "Falcon9Builder.h"
12 #include "FalconHeavyBuilder.h"
13
14 class ConfigurationManager {
15     private:
16
17         RocketBuilder * builder;
18
19     public:
20
21         ConfigurationManager(CargoCollection * c, HumanCollection * h);
22
23         ~ConfigurationManager();
24
25         Rocket * BuildAndDecorateRocket();
26 };
27
28 #endif //DERIVED_CONFIGURATIONMANAGER_H
```

5.9 Core.h

```
1 #ifndef __Core_h__
2 #define __Core_h__
3
4 class Core;
```

```

5
6 class Core
7 {
8
9 };
10
11 #endif

```

5.10 CrewDragon.h

```

1 #ifndef __CrewDragon_h__
2 #define __CrewDragon_h__
3
4 #include "SpaceCraft.h"
5 #include "HumanCollection.h"
6 #include "OperationsIteratorHumans.h"
7
8 // class SpaceCraft;
9 class CrewDragon;
10
11 class CrewDragon: public SpaceCraft
12 {
13     private:
14
15         HumanCollection * h;
16         CargoCollection * c;
17
18     public:
19
26         CrewDragon (CargoCollection * c, HumanCollection * h);
27
32         ~CrewDragon();
33
38         void printSpaceCraft();
39
40 };
41
42 #endif

```

5.11 Dragon.h

```

1 #ifndef __Dragon_h__
2 #define __Dragon_h__
3
4 #include "SpaceCraft.h"
5 #include "CargoCollection.h"
6
7 class Dragon: public SpaceCraft
8 {
9     private:
10
11         CargoCollection * c;
12
13     public:
14
20         Dragon (CargoCollection * c);
21
26         ~Dragon();
27
32         void printSpaceCraft();
33 };
34
35 #endif

```

5.12 Engine.h

```

1 #ifndef __Engine_h__
2 #define __Engine_h__
3
4 #include <iostream>
5
6 using namespace std;
7
8 class Engine;
9

```

```
10 class Engine
11 {
12     private:
13
14         Engine * next;
15
16     public:
17
18         Engine() {}
19
20 };
21
22 #endif
```

5.13 F9Stage1.h

```
1 #ifndef DERIVED_F9STAGE1_H
2 #define DERIVED_F9STAGE1_H
3
4 #include "RocketState.h"
5
6 class Rocket;
7
8 class F9Stage1 : public RocketState
9 {
10     public:
11
12         void handleChange(Rocket* r);
13
14         F9Stage1();
15
16         ~F9Stage1();
17
18 };
19
20 #endif //DERIVED_F9STAGE1_H
```

5.14 F9Stage2.h

```
1 #ifndef DERIVED_F9STAGE2_H
2 #define DERIVED_F9STAGE2_H
3
4 #include "RocketState.h"
5
6 class Rocket;
7
8 class F9Stage2 : public RocketState
9 {
10     public:
11
12         void handleChange(Rocket* r);
13
14         F9Stage2();
15
16         ~F9Stage2();
17
18 };
19
20 #endif //DERIVED_F9STAGE2_H
```

5.15 Falcon9.h

```
1 #ifndef __Falcon9_h__
2 #define __Falcon9_h__
3
4 #include "Rocket.h"
5
6 class Falcon9 : public Rocket
7 {
8     public:
9
10         void nextstage();
11
12 }
```

```

21         Falcon9(Payload*);
22
27         ~Falcon9();
28
34         string getName();
35
36 };
37
38 #endif

```

5.16 Falcon9Builder.h

```

1  #ifndef DERIVED_FALCON9BUILDER_H
2  #define DERIVED_FALCON9BUILDER_H
3
4  #include <iostream>
5  #include <vector>
6  #include "RocketBuilder.h"
7  #include "Cargo.h"
8  #include "CargoCollection.h"
9  #include "Falcon9Payload.h"
10 #include "Falcon9Propulsion.h"
11
12 class Falcon9Builder : public RocketBuilder
13 {
14     private:
15
16         Rocket * product;
17         CargoCollection * c;
18
19     public:
20
26         Falcon9Builder(CargoCollection * c);
27
32         ~Falcon9Builder();
33
38         void reset();
39
44         void addPayload();
45
50         void addPropulsion();
51
57         Rocket * getRocket();
58
59 };
60
61 #endif //DERIVED_FALCON9BUILDER_H

```

5.17 Falcon9Core.h

```

1  #ifndef __Falcon9Core_h__
2  #define __Falcon9Core_h__
3
4  #include <vector>
5  #include <iostream>
6  #include "Core.h"
7  #include "MerlinEngine.h"
8
9  using namespace std;
10
11 class Falcon9Core: public Core
12 {
13     private:
14
15         string name;
16         vector< MerlinEngine* > coreEngines;
17
18     public:
19
25         Falcon9Core(string name);
26
31         ~Falcon9Core();
32
38         string getName();
39
40 };
41
42 #endif
43

```

5.18 Falcon9Payload.h

```

1 #ifndef DERIVED_FALCON9PAYLOAD_H
2 #define DERIVED_FALCON9PAYLOAD_H
3
4 #include <iostream>
5 #include <vector>
6 #include "Cargo.h"
7 #include "Payload.h"
8 #include "Falcon9Core.h"
9 #include "MerlinEngine.h"
10 #include "Dragon.h"
11
12 using namespace std;
13
14 class Falcon9Payload : public Payload
15 {
16     public:
17
18     Falcon9Payload(SpaceCraft * s, VacuumMerlinEngine * vme);
19
20     ~Falcon9Payload();
21
22     void launch();
23 };
24
25 #endif //DERIVED_FALCON9PAYLOAD_H

```

5.19 Falcon9Propulsion.h

```

1 #ifndef DERIVED_FALCON9PROPULSION_H
2 #define DERIVED_FALCON9PROPULSION_H
3
4 #include "Payload.h"
5 #include "Propulsion.h"
6 #include "PropulsionMemento.h"
7 #include "Falcon9Core.h"
8
9 class Falcon9Propulsion : public Propulsion
10 {
11     private:
12
13         Falcon9Core * core1;
14         Payload * payload;
15
16     public:
17
18         Falcon9Propulsion(Payload * p);
19
20         ~Falcon9Propulsion();
21
22         PropulsionMemento* backup ();
23
24         void restore(PropulsionMemento * );
25
26         void land();
27
28         void launch();
29 };
30
31 #endif //DERIVED_FALCON9PROPULSION_H

```

5.20 FalconHeavy.h

```

1 #ifndef __FalconHeavy_h__
2 #define __FalconHeavy_h__
3
4 #include "Rocket.h"
5
6 class FalconHeavy: public Rocket
7 {
8     public:
9
10         void nextstage();
11
12         FalconHeavy(Payload * payload);
13
14         ~FalconHeavy();
15 };

```



```

34         string getName();
35
36     };
37
38 #endif

```

5.21 FalconHeavyBuilder.h

```

1 #ifndef DERIVED_FALCONHEAVYBUILDER_H
2 #define DERIVED_FALCONHEAVYBUILDER_H
3
4 #include <iostream>
5 #include <vector>
6 #include "Human.h"
7 #include "Cargo.h"
8 #include "HumanCollection.h"
9 #include "CargoCollection.h"
10 #include "RocketBuilder.h"
11 #include "Payload.h"
12 #include "FalconHeavy.h"
13 #include "FalconHeavyPayload.h"
14 #include "Propulsion.h"
15 #include "FalconHeavyPropulsion.h"
16
17
18 class FalconHeavyBuilder : public RocketBuilder
19 {
20     private:
21
22         FalconHeavy * product;
23         CargoCollection * c;
24         HumanCollection * h;
25
26     public:
27
28         FalconHeavyBuilder(CargoCollection * c, HumanCollection * h);
29
30         ~FalconHeavyBuilder();
31
32         void reset();
33
34         void addPayload();
35
36         void addPropulsion();
37
38         FalconHeavy* getRocket();
39 };
40
41 #endif //DERIVED_FALCONHEAVYBUILDER_H

```

5.22 FalconHeavyPayload.h

```

1 #ifndef DERIVED_FALCONHEAVYPAYLOAD_H
2 #define DERIVED_FALCONHEAVYPAYLOAD_H
3
4 #include "Payload.h"
5 #include "CrewDragon.h"
6 #include "SpaceCraft.h"
7 #include "VacuumMerlinEngine.h"
8 #include "Falcon9Core.h"
9
10 class FalconHeavyPayload : public Payload
11 {
12     public:
13
14         FalconHeavyPayload(SpaceCraft * s, VacuumMerlinEngine * vme);
15
16         ~FalconHeavyPayload();
17
18         void launch();
19 };
20
21 #endif //DERIVED_FALCONHEAVYPAYLOAD_H

```

5.23 FalconHeavyPropulsion.h

```

1 #ifndef DERIVED_FALCONHEAVYPROPULSION_H
2 #define DERIVED_FALCONHEAVYPROPULSION_H
3
4 #include "Payload.h"
5 #include "Propulsion.h"
6 #include "Falcon9Core.h"
7 #include "FalconHeavyPropulsionMemento.h"
8
9 class FalconHeavyPropulsion : public Propulsion
10 {
11     private:
12
13         Falcon9Core * core1;
14         Falcon9Core * core2;
15         Falcon9Core * core3;
16         Payload * payload;
17
18     public:
19
25         FalconHeavyPropulsion(Payload * p);
26
31         ~FalconHeavyPropulsion();
32
38         FalconHeavyPropulsionMemento* backup();
39
44         void restore(FalconHeavyPropulsionMemento * );
45
50         void land();
51
56         void launch();
57
58 };
59
60 #endif //DERIVED_FALCONHEAVYPROPULSION_H

```

5.24 FalconHeavyPropulsionMemento.h

```

1 #ifndef FALCONHEAVYPROPULSIONMEMENTO_H
2 #define FALCONHEAVYPROPULSIONMEMENTO_H
3
4 #include <iostream>
5 #include "Payload.h"
6 #include "Falcon9Core.h"
7
8 class FalconHeavyPropulsionMemento
9 {
10     private:
11
12         Falcon9Core * core1;
13         Falcon9Core * core2;
14         Falcon9Core * core3;
15         Payload * payload;
16
17     public:
18
23         FalconHeavyPropulsionMemento(Falcon9Core * , Falcon9Core * , Falcon9Core * , Payload *);
24
30         Falcon9Core * getCore1();
31
37         Falcon9Core * getCore2();
38
44         Falcon9Core * getCore3();
45
51         Payload * getPay();
52 };
53 #endif

```

5.25 FHCaretaker.h

```

1 #ifndef FHCARETAKER_H
2 #define FHCARETAKER_H
3
4 #include "FalconHeavyPropulsionMemento.h"
5
6 class HPCaretaker
7 {
8     private:

```

```

9
10     FalconHeavyPropulsionMemento * backup;
11
12     public:
13
14     HPCaretaker();
15
16     FalconHeavyPropulsionMemento * getBackup();
17
18     void setBackup(FalconHeavyPropulsionMemento*);
19
20 };
21 #endif

```

5.26 FHStage1.h

```

1 #ifndef DERIVED_FHSTAGE1_H
2 #define DERIVED_FHSTAGE1_H
3
4 #include "RocketState.h"
5
6 class Rocket;
7
8 class FHStage1: public RocketState
9 {
10     public:
11
12     void handleChange(Rocket* r);
13
14     FHStage1();
15
16     ~FHStage1();
17 };
18 #endif //DERIVED_FHSTAGE1_H

```

5.27 FHStage2.h

```

1 #ifndef DERIVED_FHSTAGE2_H
2 #define DERIVED_FHSTAGE2_H
3
4 #include "RocketState.h"
5
6 class Rocket;
7
8 class FHStage2: public RocketState
9 {
10     public:
11
12     void handleChange(Rocket* r);
13
14     FHStage2();
15
16     ~FHStage2();
17 };
18 #endif //DERIVED_FHSTAGE2_H

```

5.28 GroundMissionControl.h

```

1 #ifndef __GroundMissionControl_h__
2 #define __GroundMissionControl_h__
3
4 #include <vector>
5 #include "Satellite.h"
6
7 using namespace std;
8
9 class GroundMissionControl
10 {
11     private:
12
13     vector<Satellite*> satelliteList;

```

```
14
15     public:
16
21         GroundMissionControl();
22
27         void attach(Satellite*);
28
33         void detach(Satellite*);
34
39         void notify();
40
45         ~GroundMissionControl();
46
47 };
48
49 #endif
```

5.29 Human.h

```
1 #ifndef __Human_h__
2 #define __Human_h__
3
4 #include <string>
5 #include "VectorOfHumans.h"
6
7 using namespace std;
8
9 class Human
10 {
11     private:
12
13         string name;
14         double weight;
15         string role;
16
17     public:
18
24         string getName();
25
31         double getWeight();
32
38         string getRole();
39
47         Human(string name, double weight, string role);
48
53         ~Human();
54 };
55 #endif
```

5.30 HumanCollection.h

```
1 #ifndef __HumanCollection_h__
2 #define __HumanCollection_h__
3
4 class Human;
5 class HumanIterator;
6 class HumanCollection;
7
8 class HumanCollection
9 {
10     public:
11
16         HumanCollection();
17
23         virtual HumanIterator* createHumanIterator()=0;
24
29         virtual void addHuman(Human*)=0;
30
35         virtual void removeHuman(Human*)=0;
36
43         virtual bool isEmpty()=0;
44
49         virtual ~HumanCollection();
50
51 };
52
53 #endif
```

5.31 HumanFactory.h

```
1 #ifndef HUMANFACTORY_H
2 #define HUMANFACTORY_H
3
4 #include "Human.h"
5 #include "Colours.h"
6
7 class HumanFactory
8 {
9     private:
10
11     public:
12
13         HumanFactory();
14
15         ~HumanFactory();
16
17         Human* createHuman(string name, double weight, string role);
18 };
19
20 #endif
```

5.32 HumanIterator.h

```
1 #ifndef __HumanIterator_h__
2 #define __HumanIterator_h__
3
4 #include "HumanCollection.h"
5
6 class Human;
7 class HumanIterator;
8
9 class HumanIterator
10 {
11     public:
12
13         HumanIterator();
14
15         virtual Human* first()=0;
16
17         virtual Human* next()=0;
18
19         virtual bool hasNext()=0;
20
21         virtual Human* current()=0;
22
23         virtual ~HumanIterator();
24 };
25
26 #endif
```

5.33 Iterator.h

```
1 #ifndef PROJECT_ITERATOR_H
2 #define PROJECT_ITERATOR_H
3
4 class Iterator {
5
6 };
7
8 #endif //PROJECT_ITERATOR_H
```

5.34 LaunchMode.h

```
1 #ifndef __LaunchMode_h__
2 #define __LaunchMode_h__
3
4 #include "Simulation.h"
5
6 class LaunchMode
7 {
8     private:
```

```

9
10     Simulation * receiver;
11
12     public:
13
19         LaunchMode(Simulation * obj);
20
25         void execute();
26
32         Simulation * getR();
33
34 };
35
36 #endif

```

5.35 MerlinEngine.h

```

1 #ifndef __MerlinEngine_h__
2 #define __MerlinEngine_h__
3
4 #include <iostream>
5 #include "Engine.h"
6
7 class MerlinEngine : public Engine
8 {
9     private:
10
11         string name;
12
13     public:
14
20         MerlinEngine(string name);
21
27         MerlinEngine * clone();
28
33         ~MerlinEngine();
34 };
35
36 #endif

```

5.36 OperationsIteratorCargo.h

```

1 #ifndef DERIVED_OPERATIONSITERATORCARGO_H
2 #define DERIVED_OPERATIONSITERATORCARGO_H
3
4 #include <iostream>
5 #include <vector>
6 #include "Cargo.h"
7
8 using namespace std;
9
10 class Cargo;
11 class CargoIterator;
12
13 class OperationsIteratorCargo: public CargoIterator
14 {
15     private:
16
17         int currentpos{};
18         vector<Cargo*> itlist;
19
20     public:
21
27         explicit OperationsIteratorCargo(vector<Cargo*> vect);
28
34         Cargo* first() override;
35
41         Cargo* next() override;
42
49         bool hasNext() override;
50
56         Cargo* current() override;
57
62         ~OperationsIteratorCargo() override=default;
63
64 };
65
66 #endif //DERIVED_OPERATIONSITERATOR_H

```

5.37 OperationsIteratorHumans.h

```

1 #ifndef DERIVED_OPERATIONSITERATORHUMANS_H
2 #define DERIVED_OPERATIONSITERATORHUMANS_H
3
4 #include <vector>
5 #include "HumanIterator.h"
6
7 using namespace std;
8
9 class Human;
10
11 class OperationsIteratorHumans : public HumanIterator
12 {
13     private:
14
15         int currentpos;
16         vector<Human*> itlist;
17
18     public:
19
20         explicit OperationsIteratorHumans(vector<Human*> vect);
21
22         Human* first() override;
23
24         Human* next() override;
25
26         bool hasNext() override;
27
28         Human* current() override;
29
30         ~OperationsIteratorHumans() override=default;
31
32 };
33
34 #endif //DERIVED_OPERATIONSITERATORHUMANS_H

```

5.38 Payload.h

```

1 #ifndef DERIVED_PAYLOAD_H
2 #define DERIVED_PAYLOAD_H
3
4 #include <iostream>
5 #include <vector>
6 #include "VacuumMerlinEngine.h"
7 #include "SpaceCraft.h"
8
9 using namespace std;
10
11
12 class Payload
13 {
14     private:
15
16         VacuumMerlinEngine * vacmerlinengine;
17         SpaceCraft * spacecraft;
18
19     public:
20
21         VacuumMerlinEngine * getVacuumMerlinEngine();
22
23         void setVacuumMerlinEngine(VacuumMerlinEngine * v);
24
25         SpaceCraft * getSpaceCraft();
26
27         void setSpaceCraft(SpaceCraft * s);
28
29         void print();
30
31         Payload(SpaceCraft * s, VacuumMerlinEngine * v);
32
33         Payload();
34
35         virtual ~Payload();
36
37         bool testVacuumMerlinEngine();
38
39         virtual void launch() = 0;
40
41 };
42
43 #endif //DERIVED_PAYLOAD_H

```

5.39 PropCaretaker.h

```
1 #ifndef PROPCARETAKER_H
2 #define PROPCARETAKER_H
3
4 #include "PropulsionMemento.h"
5
6 class PropCaretaker
7 {
8     private:
9
10         PropulsionMemento * backup;
11
12     public:
13
14         PropCaretaker();
15
16         PropulsionMemento * getBackup();
17
18         void setBackup(PropulsionMemento* );
19 };
20 #endif
```

5.40 Propulsion.h

```
1 #ifndef DERIVED_PROPULSION_H
2 #define DERIVED_PROPULSION_H
3
4 #include "Payload.h"
5
6 class Propulsion : public Payload
7 {
8
9     private:
10
11     public:
12
13         Propulsion();
14
15         virtual ~Propulsion();
16
17         virtual void land()=0;
18
19         bool testPropulsion();
20 };
21 #endif //DERIVED_PROPULSION_H
```

5.41 PropulsionMemento.h

```
1 #ifndef PROPULSIONMEMENTO_H
2 #define PROPULSIONMEMENTO_H
3
4 #include <iostream>
5 #include "Falcon9Core.h"
6 #include "Payload.h"
7
8 class PropulsionMemento
9 {
10     private:
11
12         Falcon9Core * core ;
13         Payload * payload;
14
15     public:
16
17         PropulsionMemento(Falcon9Core * core, Payload * pay);
18
19         Falcon9Core * getCore();
20
21         Payload * getPay();
22 };
23 #endif
```


5.42 Rocket.h

```
1 #ifndef __Rocket_h__
2 #define __Rocket_h__
3
4 #include "Payload.h"
5 #include "RocketState.h"
6 #include "Propulsion.h"
7
8 using namespace std;
9
10 class Rocket
11 {
12
13     private:
14
15         bool readytolaunch;
16         bool hasbeentested;
17         string name;
18
19     protected:
20
21         Payload * payload;
22         RocketState* stage;
23         Propulsion * propulsion;
24
25     public:
26
27         Rocket(Payload * payload);
28
29         virtual ~Rocket();
30
31         virtual void nextstage()=0;
32
33         void setPayload(Payload * p);
34
35         void print();
36
37         Payload * getPayload();
38
39         void setState(RocketState *state);
40
41         void setPropulsion(Propulsion * p);
42
43         void launch();
44
45         void testRocket();
46
47         bool testPropulsion();
48
49         bool testVacuumMerlinEngine();
50
51         void setName(string name);
52
53         virtual string getName()=0;
54
55         void testThatFails();
56 };
57
58 #endif
```

5.43 RocketBuilder.h

```
1 #ifndef DERIVED_ROCKETBUILDER_H
2 #define DERIVED_ROCKETBUILDER_H
3
4 #include "Rocket.h"
5
6 class RocketBuilder
7 {
8     public:
9
10         RocketBuilder();
11
12         virtual ~RocketBuilder();
13
14         virtual void reset() = 0;
15
16         virtual void addPayload() = 0;
17
18         virtual void addPropulsion() = 0;
19
20         virtual Rocket * getRocket() = 0;
21 }
```

```

46
47 };
48
49 #endif //DERIVED_ROCKETBUILDER_H

```

5.44 RocketState.h

```

1 #ifndef DERIVED_ROCKETSTATE_H
2 #define DERIVED_ROCKETSTATE_H
3
4 class Rocket;
5
6 class RocketState
7 {
8     private:
9
10    public:
11
12        virtual void handleChange(Rocket* r)=0;
13
14        RocketState();
15
16        virtual ~RocketState();
17 };
18
19 #endif //DERIVED_ROCKETSTATE_H

```

5.45 Satellite.h

```

1 #ifndef __Satellite_h__
2 #define __Satellite_h__
3
4 #include <iostream>
5 #include <string>
6 #include "VacuumMerlinEngine.h"
7 #include "SatelliteTransmission.h"
8 #include "Cargo.h"
9
10 class SatelliteTransmission;
11
12 using namespace std;
13
14 class Satellite : public Cargo
15 {
16     private:
17
18         VacuumMerlinEngine* vacuumMerlinEngine;
19         SatelliteTransmission* mediator;
20         string name;
21
22     public:
23
24         Satellite();
25
26         void ConnectionChanged();
27
28         virtual string getConnection() = 0;
29
30         virtual void setConnection(string) = 0;
31
32         virtual void update() = 0;
33
34         virtual void print() = 0;
35
36         ~Satellite();
37 };
38
39 #endif

```

5.46 SatelliteTransmission.h

```

1 #ifndef SATELLITETRANSMISSION_H
2 #define SATELLITETRANSMISSION_H

```

```

3
4 #include <iostream>
5 #include "Satellite.h"
6
7 class Satellite;
8
9 using namespace std;
10 //Mediator - Controls Communication between Satellites
11 class SatelliteTransmission
12 {
13     private:
14
15     public:
16
17         SatelliteTransmission();
18
19         virtual void notify(Satellite*) = 0;
20
21         ~SatelliteTransmission();
22 };
23 #endif

```

5.47 SatelliteTransmissionController.h

```

1 #ifndef SATELLITETRANSMISSIONCONTROLLER_H
2 #define SATELLITETRANSMISSIONCONTROLLER_H
3
4 #include <iostream>
5 #include "SatelliteTransmission.h"
6
7 using namespace std;
8
9 class SatelliteTransmissionController : public SatelliteTransmission
10 {
11     private:
12
13     public:
14
15         SatelliteTransmissionController();
16
17         void notify(Satellite*);
18
19         ~SatelliteTransmissionController();
20 };
21 #endif

```

5.48 Simulation.h

```

1 #ifndef __Simulation_h__
2 #define __Simulation_h__
3
4 #include "AllHeaders.h"
5
6 class Simulation
7 {
8     private:
9
10         HumanCollection * voh ; //vector of humans.
11         CargoCollection * voc ; //vecotre of cargor.
12         ConcreteGroundMissionControl * groundMissionControl;
13         Rocket* dogeToMoon;
14         int send;
15
16     public:
17
18         Simulation();
19
20         ~Simulation();
21
22         void setup();
23
24         void liftOff();
25
26         void staticFire();
27 }

```

```
52         void testThatFails();
53
54     };
55
56 #endif
```

5.49 SpaceCraft.h

```
1 #ifndef __SpaceCraft_h__
2 #define __SpaceCraft_h__
3
4 #include <string>
5 #include "VacuumMerlinEngine.h"
6 #include "Human.h"
7 #include "CargoCollection.h"
8
9 class SpaceCraft;
10
11 class SpaceCraft
12 {
13     private:
14
15         static int counter;
16         int id;
17         string mission;
18         Human* captain;
19         CargoCollection * c;
20
21     public:
22
23         SpaceCraft(CargoCollection * c);
24
25         virtual ~SpaceCraft();
26
27         void setCargoCollection(CargoCollection * c);
28
29         CargoCollection * getCargoCollection();
30
31         virtual void printSpaceCraft()=0; //to be re-impliment in the CrewDragon
32 };
33
34 #endif
```

5.50 SpaceCraftFactory.h

```
1 #ifndef __SpaceCraftFactory_h__
2 #define __SpaceCraftFactory_h__
3
4 class SpaceCraftFactory;
5
6 class SpaceCraftFactory
7 {
8
9 };
10
11 #endif
```

5.51 StarLinkSatellite.h

```
1 #ifndef __StarLinkSatellite_h__
2 #define __StarLinkSatellite_h__
3
4 #include <iostream>
5 // #include "Colours.h"
6 #include "Satellite.h"
7 #include "ConcreteGroundMissionControl.h"
8
9 using namespace std;
10
11 // class ConcreteGroundMissionControl;
12
13 class StarLinkSatellite: public Satellite
14 {
15     private:
16
```

```

17     ConcreteGroundMissionControl* missionControl;
18     bool communicating;
19     string connection;
20     string name;
21
22     public:
23
24         StarLinkSatellite(ConcreteGroundMissionControl*);
25
26         StarLinkSatellite(string);
27
28         StarLinkSatellite * clone();
29
30         string getName();
31
32         void setName(string);
33
34         void update();
35
36         void print();
37
38         string getConnection();
39
40         void setConnection(string);
41
42         ~StarLinkSatellite();
43
44 };
45 #endif

```

5.52 VacuumMerlinEngine.h

```

1 #ifndef __VacuumMerlinEngine_h__
2 #define __VacuumMerlinEngine_h__
3
4 #include "Engine.h"
5
6 // class Engine;
7 class VacuumMerlinEngine;
8
9 class VacuumMerlinEngine: public Engine
10 {
11
12     private:
13
14         static int counter;
15         int id;
16
17     public:
18
19         VacuumMerlinEngine();
20
21         VacuumMerlinEngine * clone();
22
23         ~VacuumMerlinEngine();
24
25 };
26 #endif

```

5.53 VectorOfCargo.h

```

1 #ifndef DERIVED_VECTOROFCARGO_H
2 #define DERIVED_VECTOROFCARGO_H
3
4 #include <iostream>
5 #include <vector>
6 #include "CargoIterator.h"
7 #include "OperationsIteratorCargo.h"
8
9
10 using namespace std;
11
12 class CargoCollection;
13
14 class VectorOfCargo: public CargoCollection
15 {
16     private:

```

```
17
18     vector<Cargo*> cargo;
19
20     public:
21
22     VectorOfCargo();
23
24     CargoIterator* createCargoIterator() override;
25
26     void addCargo(Cargo*) override;
27
28     void removeCargo(Cargo*) override;
29
30     bool isEmpty() override;
31
32     ~VectorOfCargo() ;
33 };
34 #endif //DERIVED_VECTOROFCARGO_H
```

5.54 VectorOfHumans.h

```
1 #ifndef DERIVED_VECTOROFHUMANS_H
2 #define DERIVED_VECTOROFHUMANS_H
3
4 #include <iostream>
5 #include <vector>
6 #include "HumanCollection.h"
7 #include "HumanIterator.h"
8 #include "OperationsIteratorHumans.h"
9
10 using namespace std;
11
12 class VectorOfHumans : public HumanCollection
13 {
14     private:
15
16     vector<Human*> people;
17
18     public:
19
20     VectorOfHumans();
21
22     HumanIterator* createHumanIterator() override;
23
24     void addHuman(Human*) override;
25
26     void removeHuman(Human*) override;
27
28     bool isEmpty() override;
29
30     ~VectorOfHumans() override;
31 };
32 #endif //DERIVED_VECTOROFHUMANS_H
```

Index

- ~Cargo
 - Cargo, 8
- ~CargoCollection
 - CargoCollection, 10
- ~CargoFactory
 - CargoFactory, 12
- ~CargoIterator
 - CargoIterator, 13
- ~ConcreteGroundMissionControl
 - ConcreteGroundMissionControl, 19
- ~ConfigurationManager
 - ConfigurationManager, 21
- ~CrewDragon
 - CrewDragon, 22
- ~Dragon
 - Dragon, 24
- ~F9Stage1
 - F9Stage1, 26
- ~F9Stage2
 - F9Stage2, 27
- ~FHStage1
 - FHStage1, 47
- ~FHStage2
 - FHStage2, 49
- ~Falcon9
 - Falcon9, 28
- ~Falcon9Builder
 - Falcon9Builder, 30
- ~Falcon9Core
 - Falcon9Core, 33
- ~Falcon9Payload
 - Falcon9Payload, 34
- ~Falcon9Propulsion
 - Falcon9Propulsion, 36
- ~FalconHeavy
 - FalconHeavy, 38
- ~FalconHeavyBuilder
 - FalconHeavyBuilder, 40
- ~FalconHeavyPayload
 - FalconHeavyPayload, 42
- ~FalconHeavyPropulsion
 - FalconHeavyPropulsion, 43
- ~GroundMissionControl
 - GroundMissionControl, 50
- ~Human
 - Human, 54
- ~HumanCollection
 - HumanCollection, 56
- ~HumanFactory
 - HumanFactory, 58
- ~HumanIterator
 - HumanIterator, 59
- ~MerlinEngine
 - MerlinEngine, 64
- ~OperationsIteratorCargo
 - OperationsIteratorCargo, 65
- ~OperationsIteratorHumans
 - OperationsIteratorHumans, 68
- ~Payload
 - Payload, 71
- ~Propulsion
 - Propulsion, 75
- ~Rocket
 - Rocket, 79
- ~RocketBuilder
 - RocketBuilder, 84
- ~RocketState
 - RocketState, 86
- ~Satellite
 - Satellite, 87
- ~SatelliteTransmission
 - SatelliteTransmission, 90
- ~SatelliteTransmissionController
 - SatelliteTransmissionController, 91
- ~Simulation
 - Simulation, 92
- ~SpaceCraft
 - SpaceCraft, 95
- ~StarLinkSatellite
 - StarLinkSatellite, 97
- ~VacuumMerlinEngine
 - VacuumMerlinEngine, 100
- ~VectorOfCargo
 - VectorOfCargo, 101
- ~VectorOfHumans
 - VectorOfHumans, 103
- addCargo
 - CargoCollection, 10
 - VectorOfCargo, 102
- addHuman
 - HumanCollection, 56
 - VectorOfHumans, 104
- addPayload
 - Falcon9Builder, 30
 - FalconHeavyBuilder, 40
 - RocketBuilder, 84
- addPropulsion
 - Falcon9Builder, 31

- FalconHeavyBuilder, 40
- RocketBuilder, 84
- attach
 - GroundMissionControl, 51
- backup
 - Falcon9Propulsion, 36
 - FalconHeavyPropulsion, 44
- black
 - Colours, 16
- blue
 - Colours, 16
- BuildAndDecorateRocket
 - ConfigurationManager, 21
- Cargo, 7
 - ~Cargo, 8
 - Cargo, 7
 - getDescription, 8
 - getName, 8
 - getWeight, 8
- CargoCollection, 9
 - ~CargoCollection, 10
 - addCargo, 10
 - CargoCollection, 9
 - createCargoIterator, 10
 - isEmpty, 10
 - removeCargo, 11
- CargoFactory, 11
 - ~CargoFactory, 12
 - CargoFactory, 12
 - createCargo, 12
- CargoIterator, 13
 - ~CargoIterator, 13
 - CargoIterator, 13
 - current, 14
 - first, 14
 - hasNext, 14
 - next, 15
- clone
 - MerlinEngine, 64
 - StarLinkSatellite, 97
 - VacuumMerlinEngine, 100
- Colours, 15
 - black, 16
 - blue, 16
 - cyan, 16
 - green, 17
 - purple, 17
 - red, 18
 - yellow, 18
- ConcreteGroundMissionControl, 19
 - ~ConcreteGroundMissionControl, 19
 - ConcreteGroundMissionControl, 19
 - hasConnected, 20
 - setConnection, 20
- ConfigurationManager, 20
 - ~ConfigurationManager, 21
 - BuildAndDecorateRocket, 21
 - ConfigurationManager, 21
 - ConnectionChanged
 - Satellite, 88
 - Core, 22
 - createCargo
 - CargoFactory, 12
 - createCargoIterator
 - CargoCollection, 10
 - VectorOfCargo, 102
 - createHuman
 - HumanFactory, 58
 - createHumanIterator
 - HumanCollection, 56
 - VectorOfHumans, 104
 - CrewDragon, 22
 - ~CrewDragon, 22
 - CrewDragon, 22
 - printSpaceCraft, 23
 - current
 - CargoIterator, 14
 - HumanIterator, 60
 - OperationsIteratorCargo, 65
 - OperationsIteratorHumans, 68
 - cyan
 - Colours, 16
 - detach
 - GroundMissionControl, 51
 - Dragon, 23
 - ~Dragon, 24
 - Dragon, 23
 - printSpaceCraft, 24
 - Engine, 25
 - Engine, 25
 - execute
 - LaunchMode, 62
 - F9Stage1, 25
 - ~F9Stage1, 26
 - F9Stage1, 25
 - handleChange, 26
 - F9Stage2, 27
 - ~F9Stage2, 27
 - F9Stage2, 27
 - handleChange, 27
 - Falcon9, 28
 - ~Falcon9, 28
 - Falcon9, 28
 - getName, 29
 - nextstage, 29
 - Falcon9Builder, 29
 - ~Falcon9Builder, 30
 - addPayload, 30
 - addPropulsion, 31
 - Falcon9Builder, 30
 - getRocket, 31
 - reset, 31
 - Falcon9Core, 32

- ~Falcon9Core, 33
- Falcon9Core, 32
- getName, 33
- Falcon9Payload, 33
 - ~Falcon9Payload, 34
 - Falcon9Payload, 34
 - launch, 34
- Falcon9Propulsion, 35
 - ~Falcon9Propulsion, 36
 - backup, 36
 - Falcon9Propulsion, 35
 - land, 36
 - launch, 36
 - restore, 37
- FalconHeavy, 37
 - ~FalconHeavy, 38
 - FalconHeavy, 38
 - getName, 38
 - nextstage, 38
- FalconHeavyBuilder, 39
 - ~FalconHeavyBuilder, 40
 - addPayload, 40
 - addPropulsion, 40
 - FalconHeavyBuilder, 39
 - getRocket, 40
 - reset, 41
- FalconHeavyPayload, 41
 - ~FalconHeavyPayload, 42
 - FalconHeavyPayload, 42
 - launch, 42
- FalconHeavyPropulsion, 43
 - ~FalconHeavyPropulsion, 43
 - backup, 44
 - FalconHeavyPropulsion, 43
 - land, 44
 - launch, 44
 - restore, 44
- FalconHeavyPropulsionMemento, 45
 - FalconHeavyPropulsionMemento, 45
 - getCore1, 46
 - getCore2, 46
 - getCore3, 46
 - getPay, 46
- FHStage1, 47
 - ~FHStage1, 47
 - FHStage1, 47
 - handleChange, 48
- FHStage2, 48
 - ~FHStage2, 49
 - FHStage2, 49
 - handleChange, 49
- first
 - CargoIterator, 14
 - HumanIterator, 60
 - OperationsIteratorCargo, 66
 - OperationsIteratorHumans, 68
- getBackup
 - HPCaretaker, 52
- PropCaretaker, 74
- getCargoCollection
 - SpaceCraft, 95
- getConnection
 - Satellite, 88
 - StarLinkSatellite, 97
- getCore
 - PropulsionMemento, 77
- getCore1
 - FalconHeavyPropulsionMemento, 46
- getCore2
 - FalconHeavyPropulsionMemento, 46
- getCore3
 - FalconHeavyPropulsionMemento, 46
- getDescription
 - Cargo, 8
- getName
 - Cargo, 8
 - Falcon9, 29
 - Falcon9Core, 33
 - FalconHeavy, 38
 - Human, 54
 - Rocket, 79
 - StarLinkSatellite, 98
- getPay
 - FalconHeavyPropulsionMemento, 46
 - PropulsionMemento, 77
- getPayload
 - Rocket, 79
- getR
 - LaunchMode, 62
- getRocket
 - Falcon9Builder, 31
 - FalconHeavyBuilder, 40
 - RocketBuilder, 84
- getRole
 - Human, 54
- getSpaceCraft
 - Payload, 71
- getVacuumMerlinEngine
 - Payload, 71
- getWeight
 - Cargo, 8
 - Human, 54
- green
 - Colours, 17
- GroundMissionControl, 50
 - ~GroundMissionControl, 50
 - attach, 51
 - detach, 51
 - GroundMissionControl, 50
 - notify, 51
- handleChange
 - F9Stage1, 26
 - F9Stage2, 27
 - FHStage1, 48
 - FHStage2, 49
 - RocketState, 86

- hasConnected
 - ConcreteGroundMissionControl, 20
- hasNext
 - CargoIterator, 14
 - HumanIterator, 60
 - OperationsIteratorCargo, 66
 - OperationsIteratorHumans, 68
- HPCCaretaker, 52
 - getBackup, 52
 - HPCCaretaker, 52
 - setBackup, 52
- Human, 53
 - ~Human, 54
 - getName, 54
 - getRole, 54
 - getWeight, 54
 - Human, 53
- HumanCollection, 55
 - ~HumanCollection, 56
 - addHuman, 56
 - createHumanIterator, 56
 - HumanCollection, 55
 - isEmpty, 56
 - removeHuman, 57
- HumanFactory, 57
 - ~HumanFactory, 58
 - createHuman, 58
 - HumanFactory, 58
- HumanIterator, 59
 - ~HumanIterator, 59
 - current, 60
 - first, 60
 - hasNext, 60
 - HumanIterator, 59
 - next, 61
- isEmpty
 - CargoCollection, 10
 - HumanCollection, 56
 - VectorOfCargo, 102
 - VectorOfHumans, 104
- Iterator, 61
- land
 - Falcon9Propulsion, 36
 - FalconHeavyPropulsion, 44
 - Propulsion, 75
- launch
 - Falcon9Payload, 34
 - Falcon9Propulsion, 36
 - FalconHeavyPayload, 42
 - FalconHeavyPropulsion, 44
 - Payload, 71
 - Rocket, 79
- LaunchMode, 62
 - execute, 62
 - getR, 62
 - LaunchMode, 62
- liftOff
 - Simulation, 93
- MerlinEngine, 63
 - ~MerlinEngine, 64
 - clone, 64
 - MerlinEngine, 63
- next
 - CargoIterator, 15
 - HumanIterator, 61
 - OperationsIteratorCargo, 66
 - OperationsIteratorHumans, 69
- nextstage
 - Falcon9, 29
 - FalconHeavy, 38
 - Rocket, 80
- notify
 - GroundMissionControl, 51
 - SatelliteTransmission, 90
 - SatelliteTransmissionController, 91
- OperationsIteratorCargo, 64
 - ~OperationsIteratorCargo, 65
 - current, 65
 - first, 66
 - hasNext, 66
 - next, 66
 - OperationsIteratorCargo, 65
- OperationsIteratorHumans, 67
 - ~OperationsIteratorHumans, 68
 - current, 68
 - first, 68
 - hasNext, 68
 - next, 69
 - OperationsIteratorHumans, 67
- Payload, 69
 - ~Payload, 71
 - getSpaceCraft, 71
 - getVacuumMerlinEngine, 71
 - launch, 71
 - Payload, 70
 - print, 72
 - setSpaceCraft, 72
 - setVacuumMerlinEngine, 72
 - testVacuumMerlinEngine, 73
- print
 - Payload, 72
 - Rocket, 80
 - Satellite, 88
 - StarLinkSatellite, 98
- printSpaceCraft
 - CrewDragon, 23
 - Dragon, 24
 - SpaceCraft, 95
- PropCaretaker, 73
 - getBackup, 74
 - PropCaretaker, 73
 - setBackup, 74

- Propulsion, 75
 - ~Propulsion, 75
 - land, 75
 - Propulsion, 75
 - testPropulsion, 76
- PropulsionMemento, 76
 - getCore, 77
 - getPay, 77
 - PropulsionMemento, 76
- purple
 - Colours, 17
- red
 - Colours, 18
- removeCargo
 - CargoCollection, 11
 - VectorOfCargo, 102
- removeHuman
 - HumanCollection, 57
 - VectorOfHumans, 105
- reset
 - Falcon9Builder, 31
 - FalconHeavyBuilder, 41
 - RocketBuilder, 85
- restore
 - Falcon9Propulsion, 37
 - FalconHeavyPropulsion, 44
- Rocket, 77
 - ~Rocket, 79
 - getName, 79
 - getPayload, 79
 - launch, 79
 - nextstage, 80
 - print, 80
 - Rocket, 78
 - setName, 80
 - setPayload, 81
 - setPropulsion, 81
 - setState, 81
 - testPropulsion, 82
 - testRocket, 82
 - testThatFails, 82
 - testVacuumMerlinEngine, 82
- RocketBuilder, 83
 - ~RocketBuilder, 84
 - addPayload, 84
 - addPropulsion, 84
 - getRocket, 84
 - reset, 85
 - RocketBuilder, 83
- RocketState, 85
 - ~RocketState, 86
 - handleChange, 86
 - RocketState, 86
- Satellite, 87
 - ~Satellite, 87
 - ConnectionChanged, 88
 - getConnection, 88
 - print, 88
 - Satellite, 87
 - setConnection, 89
 - update, 89
- SatelliteTransmission, 89
 - ~SatelliteTransmission, 90
 - notify, 90
 - SatelliteTransmission, 90
- SatelliteTransmissionController, 91
 - ~SatelliteTransmissionController, 91
 - notify, 91
 - SatelliteTransmissionController, 91
- setBackup
 - HPCaretaker, 52
 - PropCaretaker, 74
- setCargoCollection
 - SpaceCraft, 95
- setConnection
 - ConcreteGroundMissionControl, 20
 - Satellite, 89
 - StarLinkSatellite, 98
- setName
 - Rocket, 80
 - StarLinkSatellite, 99
- setPayload
 - Rocket, 81
- setPropulsion
 - Rocket, 81
- setSpaceCraft
 - Payload, 72
- setState
 - Rocket, 81
- setup
 - Simulation, 93
- setVacuumMerlinEngine
 - Payload, 72
- Simulation, 92
 - ~Simulation, 92
 - liftOff, 93
 - setup, 93
 - Simulation, 92
 - staticFire, 93
 - testThatFails, 93
- SpaceCraft, 94
 - ~SpaceCraft, 95
 - getCargoCollection, 95
 - printSpaceCraft, 95
 - setCargoCollection, 95
 - SpaceCraft, 94
- SpaceCraftFactory, 96
- StarLinkSatellite, 96
 - ~StarLinkSatellite, 97
 - clone, 97
 - getConnection, 97
 - getName, 98
 - print, 98
 - setConnection, 98
 - setName, 99

- StarLinkSatellite, 96, 97
 - update, 99
- staticFire
 - Simulation, 93
- testPropulsion
 - Propulsion, 76
 - Rocket, 82
- testRocket
 - Rocket, 82
- testThatFails
 - Rocket, 82
 - Simulation, 93
- testVacuumMerlinEngine
 - Payload, 73
 - Rocket, 82
- update
 - Satellite, 89
 - StarLinkSatellite, 99
- VacuumMerlinEngine, 99
 - ~VacuumMerlinEngine, 100
 - clone, 100
 - VacuumMerlinEngine, 100
- VectorOfCargo, 101
 - ~VectorOfCargo, 101
 - addCargo, 102
 - createCargoIterator, 102
 - isEmpty, 102
 - removeCargo, 102
 - VectorOfCargo, 101
- VectorOfHumans, 103
 - ~VectorOfHumans, 103
 - addHuman, 104
 - createHumanIterator, 104
 - isEmpty, 104
 - removeHuman, 105
 - VectorOfHumans, 103
- yellow
 - Colours, 18