

## Transformation

0.1

Erzeugt von Doxygen 1.8.17



<b>1 Coordinate Transformation</b>	<b>1</b>
<b>2 Trafo</b>	<b>3</b>
<b>3 Verzeichnis der Namensbereiche</b>	<b>5</b>
3.1 Pakete . . . . .	5
<b>4 Hierarchie-Verzeichnis</b>	<b>7</b>
4.1 Klassenhierarchie . . . . .	7
<b>5 Klassen-Verzeichnis</b>	<b>9</b>
5.1 Auflistung der Klassen . . . . .	9
<b>6 Datei-Verzeichnis</b>	<b>11</b>
6.1 Auflistung der Dateien . . . . .	11
<b>7 Dokumentation der Namensbereiche</b>	<b>13</b>
7.1 Paket coordinates . . . . .	13
7.2 Paket datumstrategy . . . . .	13
7.3 Paket params . . . . .	14
7.4 Paket test . . . . .	14
7.5 Paket test.coordinates . . . . .	14
7.6 Paket test.datumstrategy . . . . .	15
7.7 Paket test.params . . . . .	15
<b>8 Klassen-Dokumentation</b>	<b>17</b>
8.1 test.coordinates.AllTests Klassenreferenz . . . . .	17
8.2 datumstrategy.CardansTransformation Klassenreferenz . . . . .	17
8.2.1 Ausführliche Beschreibung . . . . .	18
8.2.2 Dokumentation der Elementfunktionen . . . . .	18
8.2.2.1 transform() . . . . .	18
8.2.3 Dokumentation der Datenelemente . . . . .	18
8.2.3.1 e1 . . . . .	19
8.2.3.2 e2 . . . . .	19
8.2.3.3 e3 . . . . .	19
8.2.3.4 e4 . . . . .	19
8.2.3.5 e5 . . . . .	19
8.2.3.6 e6 . . . . .	19
8.2.3.7 e7 . . . . .	19
8.2.3.8 e8 . . . . .	19
8.2.3.9 e9 . . . . .	20
8.3 params.ControlParms Klassenreferenz . . . . .	20
8.3.1 Ausführliche Beschreibung . . . . .	23
8.3.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	23
8.3.2.1 ControlParms() . . . . .	23

8.3.3 Dokumentation der Elementfunktionen	23
8.3.3.1 getAusgabebetyp()	23
8.3.3.2 getDa()	23
8.3.3.3 getDestinationCoordinateX()	24
8.3.3.4 getDestinationCoordinateY()	24
8.3.3.5 getDestinationCoordinateZ()	24
8.3.3.6 getDf()	24
8.3.3.7 getDx()	24
8.3.3.8 getDy()	24
8.3.3.9 getDz()	24
8.3.3.10 getFileinput()	24
8.3.3.11 getFileoutput()	25
8.3.3.12 getFromawert()	25
8.3.3.13 getFrombwert()	25
8.3.3.14 getFromdatum()	25
8.3.3.15 getFromprojection()	25
8.3.3.16 getInstance()	25
8.3.3.17 getKindofhoehe()	25
8.3.3.18 getKindoftrafo()	26
8.3.3.19 getMassstab()	26
8.3.3.20 getNorthhem()	26
8.3.3.21 getPoint_number()	26
8.3.3.22 getPointid()	26
8.3.3.23 getQupkt()	26
8.3.3.24 getQutausch()	27
8.3.3.25 getRx2()	27
8.3.3.26 getRz1()	27
8.3.3.27 getRz3()	27
8.3.3.28 getSourceCoordinateX()	27
8.3.3.29 getSourceCoordinateY()	27
8.3.3.30 getSourceCoordinateZ()	27
8.3.3.31 getToawert()	27
8.3.3.32 getTobwert()	28
8.3.3.33 getTodatum()	28
8.3.3.34 getTopprojection()	28
8.3.3.35 getWx()	28
8.3.3.36 getWy()	28
8.3.3.37 getWz()	28
8.3.3.38 getZihoehe()	28
8.3.3.39 getZipkt()	28
8.3.3.40 getZitausch()	29
8.3.3.41 isChangedatum()	29

8.3.3.42 print()	29
8.3.3.43 setAusgabetyt()	29
8.3.3.44 setChangedatum()	29
8.3.3.45 setDa()	29
8.3.3.46 setDestinationCoordinateX()	30
8.3.3.47 setDestinationCoordinateY()	30
8.3.3.48 setDestinationCoordinateZ()	30
8.3.3.49 setDf()	30
8.3.3.50 setDx()	30
8.3.3.51 setDy()	30
8.3.3.52 setDz()	30
8.3.3.53 setFileinput()	31
8.3.3.54 setFileoutput()	31
8.3.3.55 setFromawert()	31
8.3.3.56 setFrombwert()	31
8.3.3.57 setFromdatum()	31
8.3.3.58 setFromprojection()	31
8.3.3.59 setKindofhoehe()	31
8.3.3.60 setKindoftrafo()	32
8.3.3.61 setMassstab()	32
8.3.3.62 setNorthhem()	32
8.3.3.63 setPoint_number()	32
8.3.3.64 setPointid()	32
8.3.3.65 setQupkt()	33
8.3.3.66 setQutausch()	33
8.3.3.67 setRx2()	33
8.3.3.68 setRz1()	33
8.3.3.69 setRz3()	33
8.3.3.70 setSourceCoordinateX()	33
8.3.3.71 setSourceCoordinateY()	33
8.3.3.72 setSourceCoordinateZ()	34
8.3.3.73 setToawert()	34
8.3.3.74 setTobwert()	34
8.3.3.75 setTodatum()	34
8.3.3.76 setTopprojection()	34
8.3.3.77 setWx()	34
8.3.3.78 setWy()	34
8.3.3.79 setWz()	35
8.3.3.80 setZihoehe()	35
8.3.3.81 setZipkt()	35
8.3.3.82 setZitausch()	35
8.3.4 Dokumentation der Datenelemente	35

8.3.4.1	ausgabetyp	35
8.3.4.2	changedatum	35
8.3.4.3	da	35
8.3.4.4	destinationCoordinateX	36
8.3.4.5	destinationCoordinateY	36
8.3.4.6	destinationCoordinateZ	36
8.3.4.7	df	36
8.3.4.8	dx	36
8.3.4.9	dy	36
8.3.4.10	dz	36
8.3.4.11	fileinput	36
8.3.4.12	fileoutput	37
8.3.4.13	fromawert	37
8.3.4.14	frombwert	37
8.3.4.15	fromdatum	37
8.3.4.16	fromprojection	37
8.3.4.17	instance	37
8.3.4.18	kindofhoehe	37
8.3.4.19	kindoftrafo	37
8.3.4.20	massstab	38
8.3.4.21	northhem	38
8.3.4.22	point_number	38
8.3.4.23	pointid	38
8.3.4.24	qupkt	38
8.3.4.25	qutausch	38
8.3.4.26	Rx2	38
8.3.4.27	Rz1	38
8.3.4.28	Rz3	39
8.3.4.29	sourceCoordinateX	39
8.3.4.30	sourceCoordinateY	39
8.3.4.31	sourceCoordinateZ	39
8.3.4.32	toawert	39
8.3.4.33	tobwert	39
8.3.4.34	todatum	39
8.3.4.35	toprojection	39
8.3.4.36	wx	40
8.3.4.37	wy	40
8.3.4.38	wz	40
8.3.4.39	zihoehe	40
8.3.4.40	zipkt	40
8.3.4.41	zitausch	40
8.4	coordinates.Coordinate Klassenreferenz	40

8.4.1 Ausführliche Beschreibung	41
8.4.2 Dokumentation der Elementfunktionen	41
8.4.2.1 getAsGeographicInterface()	42
8.4.2.2 getAsTargetCoordinate()	42
8.4.2.3 getPointid()	42
8.4.2.4 print()	43
8.4.2.5 setPointid()	43
8.4.3 Dokumentation der Datenelemente	43
8.4.3.1 pointid	43
8.5 test.coordinates.CoordinateFactoryTest Klassenreferenz	43
8.5.1 Ausführliche Beschreibung	44
8.5.2 Dokumentation der Elementfunktionen	44
8.5.2.1 testGetCoordGeographic()	44
8.5.2.2 testGetCoordGK()	44
8.5.2.3 testGetCoordSoldner()	44
8.5.2.4 testGetCoordUTM()	44
8.5.2.5 testGetXYZ()	45
8.6 params.EllipsoidParms Klassenreferenz	45
8.6.1 Ausführliche Beschreibung	46
8.6.2 Beschreibung der Konstruktoren und Destruktoren	47
8.6.2.1 EllipsoidParms() [1/2]	47
8.6.2.2 EllipsoidParms() [2/2]	47
8.6.3 Dokumentation der Elementfunktionen	47
8.6.3.1 calculatelongitudeParms()	47
8.6.3.2 getC()	47
8.6.3.3 getFirstEccentricity()	47
8.6.3.4 getFlattening()	48
8.6.3.5 getGK_refmer()	48
8.6.3.6 getId()	48
8.6.3.7 getM()	48
8.6.3.8 getN()	48
8.6.3.9 getSecondEccentricity()	48
8.6.3.10 getSemiMajorAxis()	48
8.6.3.11 getSemiMinorAxis()	48
8.6.3.12 getSoldner_b0()	49
8.6.3.13 getSoldner_l0()	49
8.6.3.14 getToa()	49
8.6.3.15 getTob()	49
8.6.3.16 getToc()	49
8.6.3.17 getToe2()	49
8.6.3.18 getToes2()	49
8.6.3.19 getTof()	49

8.6.3.20	getToid()	50
8.6.3.21	print()	50
8.6.3.22	setC()	50
8.6.3.23	setFirstEccentricity()	50
8.6.3.24	setFlattening()	50
8.6.3.25	setGK_refmer()	50
8.6.3.26	setId()	50
8.6.3.27	setM()	51
8.6.3.28	setN()	51
8.6.3.29	setSecondEccentricity()	51
8.6.3.30	setSemiMajorAxis()	51
8.6.3.31	setSemiMinorAxis()	51
8.6.3.32	setSoldner_b0()	51
8.6.3.33	setSoldner_l0()	51
8.6.3.34	setToa()	52
8.6.3.35	setTob()	52
8.6.3.36	setToc()	52
8.6.3.37	setToe2()	52
8.6.3.38	setToes2()	52
8.6.3.39	setTof()	52
8.6.3.40	setToid()	52
8.6.4	Dokumentation der Datenelemente	53
8.6.4.1	c	53
8.6.4.2	firstEccentricity	53
8.6.4.3	flattening	53
8.6.4.4	GK_refmer	53
8.6.4.5	id	53
8.6.4.6	M	53
8.6.4.7	N	53
8.6.4.8	secondEccentricity	54
8.6.4.9	semiMajorAxis	54
8.6.4.10	semiMinorAxis	54
8.6.4.11	Soldner_b0	54
8.6.4.12	Soldner_l0	54
8.6.4.13	toa	54
8.6.4.14	tob	54
8.6.4.15	toc	54
8.6.4.16	toe2	55
8.6.4.17	toes2	55
8.6.4.18	tof	55
8.6.4.19	toid	55
8.7	datumstrategy.EulersTransformation Klassenreferenz	55



8.7.1 Ausführliche Beschreibung	56
8.7.2 Dokumentation der Elementfunktionen	56
8.7.2.1 transform()	56
8.7.3 Dokumentation der Datenelemente	57
8.7.3.1 e1	57
8.7.3.2 e2	57
8.7.3.3 e3	57
8.7.3.4 e4	57
8.7.3.5 e5	57
8.7.3.6 e6	57
8.7.3.7 e7	57
8.7.3.8 e8	58
8.7.3.9 e9	58
8.8 coordinates.Gauss Klassenreferenz	58
8.8.1 Ausführliche Beschreibung	59
8.8.2 Dokumentation der Elementfunktionen	60
8.8.2.1 CalculateAbszisseOrdinate()	60
8.8.2.2 getAbszisse()	60
8.8.2.3 getAsGeographicInterface()	60
8.8.2.4 getEllipsoidal()	61
8.8.2.5 getHeight()	61
8.8.2.6 getOrdinate()	61
8.8.2.7 getScale()	62
8.8.2.8 meridianLength()	62
8.8.2.9 print()	62
8.8.2.10 setAbszisse()	62
8.8.2.11 setEllipsoidal()	62
8.8.2.12 setHeight()	63
8.8.2.13 setOrdinate()	63
8.8.2.14 setScale()	63
8.8.3 Dokumentation der Datenelemente	64
8.8.3.1 abszisse	64
8.8.3.2 ellipsoidal	64
8.8.3.3 height	64
8.8.3.4 ordinate	64
8.8.3.5 scale	64
8.9 coordinates.GaussKrueger Klassenreferenz	64
8.9.1 Ausführliche Beschreibung	65
8.9.2 Beschreibung der Konstruktoren und Destruktoren	66
8.9.2.1 GaussKrueger()	66
8.9.3 Dokumentation der Elementfunktionen	66
8.9.3.1 getAsGeographicInterface()	66

8.9.3.2 getAsTargetCoordinate()	66
8.9.3.3 getHoch()	68
8.9.3.4 getLlao()	68
8.9.3.5 getRechts()	68
8.9.3.6 print()	69
8.9.3.7 setHoch()	69
8.9.3.8 setLlao()	70
8.9.3.9 setRechts()	70
8.9.4 Dokumentation der Datenelemente	70
8.9.4.1 centralMeridian	70
8.9.4.2 hoch	70
8.9.4.3 rechts	71
8.10 test.coordinates.GaussKruegerTest Klassenreferenz	71
8.10.1 Ausführliche Beschreibung	71
8.10.2 Dokumentation der Elementfunktionen	71
8.10.2.1 getAsGaussKrueger()	72
8.10.2.2 getAsGeographic()	72
8.10.3 Dokumentation der Datenelemente	72
8.10.3.1 EPSILON_IN_DEGREE	72
8.10.3.2 EPSILON_IN_METER	72
8.10.3.3 RHO	72
8.11 test.coordinates.GaussTest Klassenreferenz	72
8.11.1 Ausführliche Beschreibung	73
8.11.2 Dokumentation der Elementfunktionen	73
8.11.2.1 testMeridianLength()	73
8.11.3 Dokumentation der Datenelemente	73
8.11.3.1 EPSILON_IN_METER	73
8.11.3.2 RHO	73
8.12 coordinates.GeographicCoordinate Klassenreferenz	74
8.12.1 Ausführliche Beschreibung	75
8.12.2 Beschreibung der Konstruktoren und Destruktoren	76
8.12.2.1 GeographicCoordinate() [1/3]	76
8.12.2.2 GeographicCoordinate() [2/3]	76
8.12.2.3 GeographicCoordinate() [3/3]	76
8.12.3 Dokumentation der Elementfunktionen	76
8.12.3.1 getAsGeographicInterface()	76
8.12.3.2 getAsTargetCoordinate()	77
8.12.3.3 getEllipsoidal()	77
8.12.3.4 getHeight()	77
8.12.3.5 getInstance() [1/2]	78
8.12.3.6 getInstance() [2/2]	78
8.12.3.7 getLatitude()	78

8.12.3.8 getLongitude()	79
8.12.3.9 getPoint_number()	79
8.12.3.10 print()	79
8.12.3.11 resetState()	79
8.12.3.12 setEllipsoidal()	79
8.12.3.13 setHeight()	80
8.12.3.14 setLatitude()	80
8.12.3.15 setLongitude()	80
8.12.3.16 setPoint_number()	81
8.12.4 Dokumentation der Datenelemente	81
8.12.4.1 ellipsoidal	81
8.12.4.2 height	81
8.12.4.3 latitude	81
8.12.4.4 longitude	81
8.12.4.5 myInstance	81
8.12.4.6 point_number	82
8.13 coordinates.GeographicCoordinateInterface Klassenreferenz	82
8.13.1 Ausführliche Beschreibung	84
8.13.2 Beschreibung der Konstruktoren und Destruktoren	84
8.13.2.1 GeographicCoordinateInterface() [1/3]	84
8.13.2.2 GeographicCoordinateInterface() [2/3]	85
8.13.2.3 GeographicCoordinateInterface() [3/3]	85
8.13.3 Dokumentation der Elementfunktionen	85
8.13.3.1 copy()	85
8.13.3.2 fupubre()	85
8.13.3.3 GeographicLongitudeLatitude()	86
8.13.3.4 getAsGeographicInterface()	86
8.13.3.5 getAsTargetCoordinate()	86
8.13.3.6 getEllipsoidal()	87
8.13.3.7 getHeight()	87
8.13.3.8 getInstance() [1/3]	87
8.13.3.9 getInstance() [2/3]	87
8.13.3.10 getInstance() [3/3]	88
8.13.3.11 getLatitude()	88
8.13.3.12 getLongitude()	88
8.13.3.13 getPoint_number()	89
8.13.3.14 molodenskytrafo()	89
8.13.3.15 print()	89
8.13.3.16 resetState()	89
8.13.3.17 setEllipsoidal()	90
8.13.3.18 setHeight()	90
8.13.3.19 setLatitude()	90

8.13.3.20 setLongitude()	90
8.13.3.21 setPoint_number()	91
8.13.4 Dokumentation der Datenelemente	91
8.13.4.1 ellipsoidal	91
8.13.4.2 height	91
8.13.4.3 latitude	91
8.13.4.4 longitude	91
8.13.4.5 myInstance	92
8.13.4.6 point_number	92
8.14 test.coordinates.GeographicCoordinateTest Klassenreferenz	92
8.14.1 Dokumentation der Elementfunktionen	92
8.14.1.1 testFromGeographicInterface()	92
8.14.1.2 testGetAsGeographicInterface()	92
8.14.2 Dokumentation der Datenelemente	92
8.14.2.1 EPSILONFORDEGREE	93
8.15 coordinates.GetAppropriateCoordinate Klassenreferenz	93
8.15.1 Ausführliche Beschreibung	93
8.15.2 Dokumentation der Elementfunktionen	94
8.15.2.1 getCoord()	94
8.15.2.2 getGeographicCoordinateInterface()	94
8.16 datumstrategy.GetAppropriateTransformationAlgorithm Klassenreferenz	94
8.16.1 Ausführliche Beschreibung	94
8.16.2 Dokumentation der Elementfunktionen	94
8.16.2.1 getStrategy()	95
8.17 test.datumstrategy.GetAppropriateTransformationAlgorithmTest Klassenreferenz	95
8.17.1 Dokumentation der Elementfunktionen	95
8.17.1.1 testSpatialSimilarityTransformationInfin()	95
8.18 params.LatParm Klassenreferenz	95
8.18.1 Ausführliche Beschreibung	96
8.18.2 Beschreibung der Konstruktoren und Destruktoren	96
8.18.2.1 LatParm()	96
8.18.3 Dokumentation der Elementfunktionen	96
8.18.3.1 Constant()	96
8.18.3.2 getCbr()	96
8.18.3.3 getEtabr()	97
8.18.3.4 getRadg()	97
8.18.3.5 getRadm()	97
8.18.3.6 getRadn()	97
8.18.3.7 getTbr()	97
8.18.3.8 getTbr2()	97
8.18.3.9 getVbr()	97
8.18.3.10 getVbr2()	97

8.18.3.11 setCbr()	98
8.18.3.12 setEtabr()	98
8.18.3.13 setRadg()	98
8.18.3.14 setRadm()	98
8.18.3.15 setRadn()	98
8.18.3.16 setTbr()	98
8.18.3.17 setTbr2()	98
8.18.3.18 setVbr()	99
8.18.3.19 setVbr2()	99
8.18.4 Dokumentation der Datenelemente	99
8.18.4.1 cbr	99
8.18.4.2 etabr	99
8.18.4.3 radg	99
8.18.4.4 radm	99
8.18.4.5 radn	99
8.18.4.6 tbr	100
8.18.4.7 vbr	100
8.19 test.params.LatParmTest Klassenreferenz	100
8.19.1 Dokumentation der Elementfunktionen	100
8.19.1.1 test()	100
8.20 datumstrategy.MockStrategy Klassenreferenz	101
8.20.1 Dokumentation der Elementfunktionen	101
8.20.1.1 transform()	101
8.21 test.datumstrategy.MockStrategyTest Klassenreferenz	101
8.21.1 Ausführliche Beschreibung	102
8.21.2 Dokumentation der Elementfunktionen	102
8.21.2.1 testTransformGaussKruegerStrip3ToStrip4()	102
8.21.2.2 testTransformGaussKruegerStrip4ToStrip3()	103
8.21.3 Dokumentation der Datenelemente	103
8.21.3.1 EPSILON_IN_METER	103
8.22 datumstrategy.MolodenskiiTransformationAbridged Klassenreferenz	103
8.22.1 Ausführliche Beschreibung	104
8.22.2 Beschreibung der Konstruktoren und Destruktoren	104
8.22.2.1 MolodenskiiTransformationAbridged()	104
8.22.3 Dokumentation der Elementfunktionen	104
8.22.3.1 getInstance()	104
8.22.3.2 transform()	104
8.22.4 Dokumentation der Datenelemente	105
8.22.4.1 myInstance	105
8.23 test.datumstrategy.MolodenskiiTransformationAbridgedTest Klassenreferenz	105
8.23.1 Ausführliche Beschreibung	105
8.23.2 Dokumentation der Elementfunktionen	106

8.23.2.1 testMolodenskiiTransformationAbridged()	106
8.23.3 Dokumentation der Datenelemente	106
8.23.3.1 EPSILON_meter	106
8.23.3.2 EPSILON_rad	106
8.23.3.3 RHO	106
8.24 datumstrategy.MolodenskiiTransformationStandard Klassenreferenz	107
8.24.1 Ausführliche Beschreibung	107
8.24.2 Beschreibung der Konstruktoren und Destruktoren	108
8.24.2.1 MolodenskiiTransformationStandard()	108
8.24.3 Dokumentation der Elementfunktionen	108
8.24.3.1 getInstance()	108
8.24.3.2 transform()	108
8.24.4 Dokumentation der Datenelemente	108
8.24.4.1 myInstance	108
8.25 test.datumstrategy.MolodenskiiTransformationStandardTest Klassenreferenz	109
8.25.1 Ausführliche Beschreibung	109
8.25.2 Dokumentation der Elementfunktionen	109
8.25.2.1 testTransform()	109
8.25.3 Dokumentation der Datenelemente	109
8.25.3.1 EPSILON_IN_METER	110
8.25.3.2 EPSILON_IN_RAD	110
8.25.3.3 RHO	110
8.26 params.MolodenskyParm Klassenreferenz	110
8.26.1 Ausführliche Beschreibung	111
8.26.2 Beschreibung der Konstruktoren und Destruktoren	111
8.26.2.1 MolodenskyParm()	111
8.26.3 Dokumentation der Elementfunktionen	111
8.26.3.1 getDa()	111
8.26.3.2 getDf()	111
8.26.3.3 getTransx()	111
8.26.3.4 getTransy()	112
8.26.3.5 getTransz()	112
8.26.3.6 setDa()	112
8.26.3.7 setDf()	112
8.26.3.8 setTransx()	112
8.26.3.9 setTransy()	112
8.26.3.10 setTransz()	112
8.26.4 Dokumentation der Datenelemente	113
8.26.4.1 da	113
8.26.4.2 transx	113
8.27 coordinates.Soldner Klassenreferenz	113
8.27.1 Ausführliche Beschreibung	114

8.27.2 Beschreibung der Konstruktoren und Destruktoren	115
8.27.2.1 Soldner() [1/3]	115
8.27.2.2 Soldner() [2/3]	115
8.27.2.3 Soldner() [3/3]	115
8.27.3 Dokumentation der Elementfunktionen	115
8.27.3.1 getAbszisse()	115
8.27.3.2 getAsGeographicInterface()	115
8.27.3.3 getAsTargetCoordinate()	116
8.27.3.4 getB0()	116
8.27.3.5 getEllipsoidal()	116
8.27.3.6 getHeight()	117
8.27.3.7 getL0()	117
8.27.3.8 getOrdinate()	117
8.27.3.9 print()	117
8.27.3.10 setAbszisse()	117
8.27.3.11 setB0()	118
8.27.3.12 setEllipsoidal()	118
8.27.3.13 setHeight()	118
8.27.3.14 setL0()	119
8.27.3.15 setOrdinate()	119
8.27.4 Dokumentation der Datenelemente	119
8.27.4.1 abszisse	119
8.27.4.2 b0	119
8.27.4.3 ellipsoidal	119
8.27.4.4 height	120
8.27.4.5 l0	120
8.27.4.6 ordinate	120
8.28 datumstrategy.SpatialSimilarityTransformationInfin Klassenreferenz	120
8.28.1 Ausführliche Beschreibung	121
8.28.2 Dokumentation der Elementfunktionen	122
8.28.2.1 transform()	122
8.29 test.datumstrategy.SpatialSimilarityTransformationInfinTest Klassenreferenz	122
8.29.1 Ausführliche Beschreibung	122
8.29.2 Dokumentation der Elementfunktionen	123
8.29.2.1 testTransform()	123
8.29.3 Dokumentation der Datenelemente	123
8.29.3.1 EPSILON_IN_METER	123
8.29.3.2 EPSILON_IN_RAD	123
8.29.3.3 RHO	123
8.30 datumstrategy.SpatialSimilarityTransformationTrig Klassenreferenz	123
8.30.1 Dokumentation der Elementfunktionen	124
8.30.1.1 transform()	124

8.31 test.datumstrategy.SpatialSimilarityTransformationTrigTest Klassenreferenz	124
8.31.1 Ausführliche Beschreibung	125
8.31.2 Dokumentation der Elementfunktionen	125
8.31.2.1 testTransform()	125
8.31.3 Dokumentation der Datenelemente	125
8.31.3.1 EPSILON_IN_METER	125
8.31.3.2 EPSILON_IN_RAD	125
8.31.3.3 RHO	126
8.32 params.SpatialSimilarityTransformParm Klassenreferenz	126
8.32.1 Ausführliche Beschreibung	126
8.32.2 Beschreibung der Konstruktoren und Destruktoren	127
8.32.2.1 SpatialSimilarityTransformParm()	127
8.32.3 Dokumentation der Elementfunktionen	127
8.32.3.1 getAlpha()	127
8.32.3.2 getBeta()	127
8.32.3.3 getFormula()	127
8.32.3.4 getGamma()	127
8.32.3.5 getScale()	127
8.32.3.6 getTransx()	128
8.32.3.7 getTransy()	128
8.32.3.8 getTransz()	128
8.32.3.9 setAlpha()	128
8.32.3.10 setBeta()	128
8.32.3.11 setFormula()	128
8.32.3.12 setGamma()	128
8.32.3.13 setScale()	129
8.32.3.14 setTransx()	129
8.32.3.15 setTransy()	129
8.32.3.16 setTransz()	129
8.32.4 Dokumentation der Datenelemente	129
8.32.4.1 alpha	129
8.32.4.2 formula	129
8.32.4.3 scale	129
8.32.4.4 transx	130
8.33 Transformation Klassenreferenz	130
8.33.1 Ausführliche Beschreibung	130
8.33.2 Dokumentation der Elementfunktionen	130
8.33.2.1 main()	131
8.34 datumstrategy.TransformationStrategy Klassenreferenz	131
8.34.1 Ausführliche Beschreibung	131
8.34.2 Dokumentation der Elementfunktionen	132
8.34.2.1 transform()	132



8.35 coordinates.UTM Klassenreferenz	132
8.35.1 Ausführliche Beschreibung	133
8.35.2 Beschreibung der Konstruktoren und Destruktoren	133
8.35.2.1 UTM()	134
8.35.3 Dokumentation der Elementfunktionen	134
8.35.3.1 getAsGeographicInterface()	134
8.35.3.2 getAsTargetCoordinate()	134
8.35.3.3 getEast()	134
8.35.3.4 getNorth()	135
8.35.3.5 getZone()	135
8.35.3.6 init()	135
8.35.3.7 isNorthhem()	135
8.35.3.8 print()	135
8.35.3.9 setEast()	135
8.35.3.10 setNorth()	135
8.35.3.11 setNorthhem()	136
8.35.3.12 setZone()	136
8.35.4 Dokumentation der Datenelemente	136
8.35.4.1 east	136
8.35.4.2 north	136
8.35.4.3 northhem	136
8.35.4.4 zone	136
8.36 test.coordinates.UTMTest Klassenreferenz	137
8.36.1 Dokumentation der Elementfunktionen	137
8.36.1.1 testGetAsGeographicInterface()	137
8.36.1.2 testGetAsUTM()	137
8.37 coordinates.XYZCoordinate Klassenreferenz	137
8.37.1 Ausführliche Beschreibung	138
8.37.2 Beschreibung der Konstruktoren und Destruktoren	139
8.37.2.1 XYZCoordinate() [1/3]	139
8.37.2.2 XYZCoordinate() [2/3]	139
8.37.2.3 XYZCoordinate() [3/3]	139
8.37.3 Dokumentation der Elementfunktionen	140
8.37.3.1 getAsGeographicInterface()	140
8.37.3.2 getAsTargetCoordinate()	140
8.37.3.3 getX()	141
8.37.3.4 getY()	141
8.37.3.5 getZ()	141
8.37.3.6 init()	142
8.37.3.7 print()	142
8.37.3.8 setX()	142
8.37.3.9 setY()	142

8.37.3.10 setZ() . . . . .	142
8.37.4 Dokumentation der Datenelemente . . . . .	143
8.37.4.1 x . . . . .	143
8.38 test.coordinates.XYZCoordinateTest Klassenreferenz . . . . .	143
8.38.1 Dokumentation der Elementfunktionen . . . . .	143
8.38.1.1 testGetAsGeographicInterface() . . . . .	143
8.38.1.2 testGetAsTargetCoordinate() . . . . .	143
<b>9 Datei-Dokumentation</b> . . . . .	<b>145</b>
9.1 C:/Users/YangXiaofei/Desktop/Workplace0115/Transformation1/bin/README.md-Dateireferenz . . . . .	145
9.2 coordinates/Coordinate.java-Dateireferenz . . . . .	145
9.3 coordinates/Gauss.java-Dateireferenz . . . . .	145
9.4 coordinates/GaussKrueger.java-Dateireferenz . . . . .	146
9.5 coordinates/GeographicCoordinate.java-Dateireferenz . . . . .	146
9.6 coordinates/GeographicCoordinateInterface.java-Dateireferenz . . . . .	146
9.7 coordinates/GetAppropriateCoordinate.java-Dateireferenz . . . . .	146
9.8 coordinates/Soldner.java-Dateireferenz . . . . .	147
9.9 coordinates/UTM.java-Dateireferenz . . . . .	147
9.10 coordinates/XYZCoordinate.java-Dateireferenz . . . . .	147
9.11 datumstrategy/CardansTransformation.java-Dateireferenz . . . . .	147
9.12 datumstrategy/EulersTransformation.java-Dateireferenz . . . . .	148
9.13 datumstrategy/GetAppropriateTransformationAlgorithm.java-Dateireferenz . . . . .	148
9.14 datumstrategy/MockStrategy.java-Dateireferenz . . . . .	148
9.15 datumstrategy/MolodenskiiTransformationAbridged.java-Dateireferenz . . . . .	148
9.16 datumstrategy/MolodenskiiTransformationStandard.java-Dateireferenz . . . . .	149
9.17 datumstrategy/SpatialSimilarityTransformationInfin.java-Dateireferenz . . . . .	149
9.18 datumstrategy/SpatialSimilarityTransformationTrig.java-Dateireferenz . . . . .	149
9.19 datumstrategy/TransformationStrategy.java-Dateireferenz . . . . .	149
9.19.1 Ausführliche Beschreibung . . . . .	150
9.20 params/ControlParms.java-Dateireferenz . . . . .	150
9.21 params/EllipsoidParms.java-Dateireferenz . . . . .	150
9.22 params/LatParm.java-Dateireferenz . . . . .	151
9.23 params/MolodenskyParm.java-Dateireferenz . . . . .	151
9.24 params/SpatialSimilarityTransformParm.java-Dateireferenz . . . . .	151
9.25 test/coordinates/AllTests.java-Dateireferenz . . . . .	151
9.26 test/coordinates/CoordinateFactoryTest.java-Dateireferenz . . . . .	152
9.27 test/coordinates/GaussKruegerTest.java-Dateireferenz . . . . .	152
9.28 test/coordinates/GaussTest.java-Dateireferenz . . . . .	152
9.29 test/coordinates/GeographicCoordinateTest.java-Dateireferenz . . . . .	152
9.30 test/coordinates/UTMTest.java-Dateireferenz . . . . .	153
9.31 test/coordinates/XYZCoordinateTest.java-Dateireferenz . . . . .	153
9.32 test/datumstrategy/GetAppropriateTransformationAlgorithmTest.java-Dateireferenz . . . . .	153

---

9.33 test/datumstrategy/MockStrategyTest.java-Dateireferenz . . . . .	153
9.34 test/datumstrategy/MolodenskiiTransformationAbridgedTest.java-Dateireferenz . . . . .	154
9.35 test/datumstrategy/MolodenskiiTransformationStandardTest.java-Dateireferenz . . . . .	154
9.36 test/datumstrategy/SpatialSimilarityTransformationInfinTest.java-Dateireferenz . . . . .	154
9.37 test/datumstrategy/SpatialSimilarityTransformationTrigTest.java-Dateireferenz . . . . .	154
9.38 test/params/LatParmTest.java-Dateireferenz . . . . .	155
9.39 Transformation.java-Dateireferenz . . . . .	155
<b>Index</b>	<b>157</b>



# Kapitel 1

## Coordinate Transformation

### Zu beachten

For educational purposes only

### Copyright

No licence restrictions yet

### Version

0.1

### Autor

Norbert Rösch  
Steven Landgraf  
Jessica Palka  
Wentao Lu  
Svea Krikau  
Xiaofei Yang

### Bemerkungen

Steven Landgraf

- `classSpatialSimilarityTransformationInfinTest`
- Tests the Spatial Similarity [Transformation](#) as a Datum Strategy.
- `class MolokiiTransformationStandardTestdens`
- Tests the Molodenskii [Transformation](#) Standard as a Datum Strategy.

Jessica Palka

- `class MockStrategyTest`

- Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used.

Svea Krikau

- class MolodenskiiTransformationAbridged
- Algorithmic realization of the abridged molodensky transformation
- class MolodenskiiTransformationAbridgedTest
- Test of the algorithmic realization of the abridged molodensky transformation

Wentao Lu

- class SpatialSimilarityTransformationTrigTest
- Tests the Spatial Similarity [Transformation](#) as a Datum Strategy

Xiaofei Yang

- Update documentations,make web pages and Latex

# Kapitel 2

## Trafo

This is the repository of the course Fortgeschrittene Konzepte in GIS (2019/2020) at the Geodetic Institute in Karlsruhe. It contains Java code for several geodetic transformations integrated in a single programm. The course focuses on learning software engineering skills and design patterns such as singleton or factory method (creational patterns). The program is based on a structure design pattern of a Model View Controller.

This code is open source and everyone is invited to participate to this repository. Please feel free to pick open Issues and send Pull Requests (see section Contributing below).

### Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes.

### Prerequisites

Java-compiler or -IDE (e.g. Eclipse: <https://www.eclipse.org/>).

JUnit (at least Version 3 is recommended) to run the implemented tests.

### Installing

Just clone the repository.

```
git clone repo@gik-ubuntu-09.gik.kit.edu/Trafo
```

And start as a Java project with your favorite compiler/IDE (e.g. Eclipse).

The project should run without any Errors.

### Running the tests

Several Unit tests are already implemented.

## Break down into end to end tests (Todo: Delete this section?)

Explain what these tests test and why  
Give an example

## And coding style tests (Todo: Delete this section?)

Explain what these tests test and why  
Give an example

## Contributing

When contributing to this repository, please first discuss the change you wish to make via issue, email, or any other method with the owners of this repository before making a change and submit pull requests.

## Versioning

For the versions available, see the [tags on this repository](#).

## Authors

- **Norbert Roesch** - *Initial work* -
- **Wentao Lu**
- **Jessica Palka**
- **Xiaofei Yang**
- **Steven Landgraf**
- **Svea Krikau**

See also the list of [contributors](#) who participated in this project.

## License

This project is licensed under the GNU License - see the LICENSE.md file for details

## Acknowledgments

- We acknowledge all authors and participants who contributed to this project. Thank You.



## Kapitel 3

# Verzeichnis der Namensbereiche

### 3.1 Pakete

Hier folgen die Pakete mit einer Kurzbeschreibung (wenn verfügbar):

<a href="#">coordinates</a>	13
<a href="#">datumstrategy</a>	13
<a href="#">params</a>	14
<a href="#">test</a>	14
<a href="#">test.coordinates</a>	14
<a href="#">test.datumstrategy</a>	15
<a href="#">test.params</a>	15



## Kapitel 4

# Hierarchie-Verzeichnis

### 4.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

test.coordinates.AllTests . . . . .	17
coordinates.Coordinate . . . . .	40
coordinates.Gauss . . . . .	58
coordinates.GaussKrueger . . . . .	64
coordinates.UTM . . . . .	132
test.coordinates.UTMTest . . . . .	137
coordinates.GeographicCoordinate . . . . .	74
coordinates.GeographicCoordinateInterface . . . . .	82
coordinates.Soldner . . . . .	113
coordinates.XYZCoordinate . . . . .	137
test.coordinates.CoordinateFactoryTest . . . . .	43
params.EllipsoidParms . . . . .	45
params.ControlParms . . . . .	20
test.coordinates.GaussKruegerTest . . . . .	71
test.coordinates.GaussTest . . . . .	72
test.coordinates.GeographicCoordinateTest . . . . .	92
coordinates.GetAppropriateCoordinate . . . . .	93
datumstrategy.GetAppropriateTransformationAlgorithm . . . . .	94
test.datumstrategy.GetAppropriateTransformationAlgorithmTest . . . . .	95
params.LatParm . . . . .	95
test.params.LatParmTest . . . . .	100
test.datumstrategy.MockStrategyTest . . . . .	101
test.datumstrategy.MolodenskiiTransformationAbridgedTest . . . . .	105
test.datumstrategy.MolodenskiiTransformationStandardTest . . . . .	109
params.MolodenskyParm . . . . .	110
test.datumstrategy.SpatialSimilarityTransformationInfinTest . . . . .	122
test.datumstrategy.SpatialSimilarityTransformationTrigTest . . . . .	124
params.SpatialSimilarityTransformParm . . . . .	126
Transformation . . . . .	130
datumstrategy.TransformationStrategy . . . . .	131
datumstrategy.CardansTransformation . . . . .	17
datumstrategy.EulersTransformation . . . . .	55
datumstrategy.MockStrategy . . . . .	101
datumstrategy.MolodenskiiTransformationAbridged . . . . .	103
datumstrategy.MolodenskiiTransformationStandard . . . . .	107
datumstrategy.SpatialSimilarityTransformationInfin . . . . .	120
datumstrategy.SpatialSimilarityTransformationTrig . . . . .	123
test.coordinates.XYZCoordinateTest . . . . .	143



## Kapitel 5

# Klassen-Verzeichnis

### 5.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

<a href="#">test.coordinates.AllTests</a>	17
<a href="#">datumstrategy.CardansTransformation</a>	
Algorithmic realization of Translation and the Cardan's Rotation	17
<a href="#">params.ControlParms</a>	
Class <a href="#">ControlParms</a> extends class <a href="#">EllipsoidParms</a>	20
<a href="#">coordinates.Coordinate</a>	
Abstract class that represents a coordinate. Enables any concrete <a href="#">Coordinate</a> class be used as input for TransformationStrategy which relies on geographic coordinates as input/output	
40	
<a href="#">test.coordinates.CoordinateFactoryTest</a>	
Coordinate factory for plane coordinates and geographic coordinates	43
<a href="#">params.EllipsoidParms</a>	
Ellipsoid parameters	45
<a href="#">datumstrategy.EulersTransformation</a>	
Algorithmic realization of the Euler's transformation	55
<a href="#">coordinates.Gauss</a>	
Class <a href="#">Gauss</a> extends class <a href="#">Coordinate</a>	58
<a href="#">coordinates.GaussKrueger</a>	
<a href="#">Coordinate</a> point relating to a certain Gauss-Krueger reference meridian	64
<a href="#">test.coordinates.GaussKruegerTest</a>	
Test class with unit tests to test behavior class GaussKrueger	71
<a href="#">test.coordinates.GaussTest</a>	
Test class with unit tests to test behavior class Gauss	72
<a href="#">coordinates.GeographicCoordinate</a>	
Concrete implementation of abstract class <a href="#">Coordinate</a> , represents a geographic coordinate (as input coordinate)	
74	
<a href="#">coordinates.GeographicCoordinateInterface</a>	
Concrete implementation of abstract class <a href="#">Coordinate</a> , represents a geographic coordinate (As interface for strategy)	
82	
<a href="#">test.coordinates.GeographicCoordinateTest</a>	92
<a href="#">coordinates.GetAppropriateCoordinate</a>	
<a href="#">Coordinate</a> factory for plane coordinates and geographic coordinates	93
<a href="#">datumstrategy.GetAppropriateTransformationAlgorithm</a>	94

<a href="#">test.datumstrategy.GetAppropriateTransformationAlgorithmTest</a>	95
<a href="#">params.LatParm</a>	
Latitude Parameters	95
<a href="#">test.params.LatParmTest</a>	100
<a href="#">datumstrategy.MockStrategy</a>	101
<a href="#">test.datumstrategy.MockStrategyTest</a>	
Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used	101
<a href="#">datumstrategy.MolodenskiiTransformationAbridged</a>	
Algorithmic realization of the abridged molodensky transformation	103
<a href="#">test.datumstrategy.MolodenskiiTransformationAbridgedTest</a>	
Test of the algorithmic realization of the abridged molodensky transformation	105
<a href="#">datumstrategy.MolodenskiiTransformationStandard</a>	
Algorithmic realization of the molodensky transformation	107
<a href="#">test.datumstrategy.MolodenskiiTransformationStandardTest</a>	
Tests the Molodenskii <a href="#">Transformation</a> Standard as a Datum Strategy. Input values and expected values were calculated by hand	109
<a href="#">params.MolodenskyParm</a>	
Molodensky Parameters	110
<a href="#">coordinates.Soldner</a>	
Class <a href="#">Soldner</a> extends class <a href="#">Coordinate</a>	113
<a href="#">datumstrategy.SpatialSimilarityTransformationInfin</a>	
Algorithmic realization of the 3D similarity transformation also known as 7-parameter transformation	120
<a href="#">test.datumstrategy.SpatialSimilarityTransformationInfinTest</a>	
Tests the Spatial Similarity <a href="#">Transformation</a> as a Datum Strategy. Input values and expected values were calculated by hand	122
<a href="#">datumstrategy.SpatialSimilarityTransformationTrig</a>	123
<a href="#">test.datumstrategy.SpatialSimilarityTransformationTrigTest</a>	
Tests the Spatial Similarity <a href="#">Transformation</a> as a Datum Strategy. Input values and expected values were calculated by hand	124
<a href="#">params.SpatialSimilarityTransformParm</a>	
Spatial similarity transform parameters	126
<a href="#">Transformation</a>	
Main Transformatin class	130
<a href="#">datumstrategy.TransformationStrategy</a>	
Abstract class that encapsulates transformation algorithms for different coordinates	
the concrete transformation algorithms that are implemented by the child classes of <a href="#">TransformationStrategy</a> implement the abstract method <a href="#">transform(GeographicCoordinateInterface)</a> that performs the transformation with geographic coordinates as input/output	131
<a href="#">coordinates.UTM</a>	
Class <a href="#">UTM</a> extends class <a href="#">Gauss</a>	132
<a href="#">test.coordinates.UTMTest</a>	137
<a href="#">coordinates.XYZCoordinate</a>	
Class <a href="#">XYZCoordinate</a>	137
<a href="#">test.coordinates.XYZCoordinateTest</a>	143

# Kapitel 6

## Datei-Verzeichnis

### 6.1 Auflistung der Dateien

Hier folgt die Aufzählung aller Dateien mit einer Kurzbeschreibung:

Transformation.java	155
coordinates/Coordinate.java	145
coordinates/Gauss.java	145
coordinates/GaussKrueger.java	146
coordinates/GeographicCoordinate.java	146
coordinates/GeographicCoordinateInterface.java	146
coordinates/GetAppropriateCoordinate.java	146
coordinates/Soldner.java	147
coordinates/UTM.java	147
coordinates/XYZCoordinate.java	147
datumstrategy/CardansTransformation.java	147
datumstrategy/EulersTransformation.java	148
datumstrategy/GetAppropriateTransformationAlgorithm.java	148
datumstrategy/MockStrategy.java	148
datumstrategy/MolodenskiiTransformationAbridged.java	148
datumstrategy/MolodenskiiTransformationStandard.java	149
datumstrategy/SpatialSimilarityTransformationInfin.java	149
datumstrategy/SpatialSimilarityTransformationTrig.java	149
datumstrategy/TransformationStrategy.java	
Abstract class TransformationStrategy	149
params/ControlParms.java	150
params/EllipsoidParms.java	150
params/LatParm.java	151
params/MolodenskyParm.java	151
params/SpatialSimilarityTransformParm.java	151
test/coordinates/AllTests.java	151
test/coordinates/CoordinateFactoryTest.java	152
test/coordinates/GaussKruegerTest.java	152
test/coordinates/GaussTest.java	152
test/coordinates/GeographicCoordinateTest.java	152
test/coordinates/UTMTest.java	153
test/coordinates/XYZCoordinateTest.java	153
test/datumstrategy/GetAppropriateTransformationAlgorithmTest.java	153
test/datumstrategy/MockStrategyTest.java	153
test/datumstrategy/MolodenskiiTransformationAbridgedTest.java	154

test/datumstrategy/ <a href="#">MolodenskiiTransformationStandardTest.java</a>	154
test/datumstrategy/ <a href="#">SpatialSimilarityTransformationInfinTest.java</a>	154
test/datumstrategy/ <a href="#">SpatialSimilarityTransformationTrigTest.java</a>	154
test/params/ <a href="#">LatParmTest.java</a>	155



# Kapitel 7

## Dokumentation der Namensbereiche

### 7.1 Paket coordinates

#### Klassen

- class [Coordinate](#)  
*Abstract class that represents a coordinate. Enables any concrete [Coordinate](#) class be used as input for TransformationStrategy which relies on geographic coordinates as input/output*
- class [Gauss](#)  
*Class [Gauss](#) extends class [Coordinate](#).*
- class [GaussKrueger](#)  
*[Coordinate](#) point relating to a certain Gauss-Krueger reference meridian.*
- class [GeographicCoordinate](#)  
*concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (as input coordinate)*
- class [GeographicCoordinateInterface](#)  
*concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (As interface for strategy)*
- class [GetAppropriateCoordinate](#)  
*[Coordinate](#) factory for plane coordinates and geographic coordinates.*
- class [Soldner](#)  
*Class [Soldner](#) extends class [Coordinate](#).*
- class [UTM](#)  
*Class [UTM](#) extends class [Gauss](#).*
- class [XYZCoordinate](#)  
*Class [XYZCoordinate](#).*

### 7.2 Paket datumstrategy

#### Klassen

- class [CardansTransformation](#)  
*Algorithmic realization of Translation and the Cardan's Rotation.*
- class [EulersTransformation](#)

- Algorithmic realization of the Euler's transformation.*
  - class [GetAppropriateTransformationAlgorithm](#)
  - class [MockStrategy](#)
  - class [MolodenskiiTransformationAbridged](#)
    - Algorithmic realization of the abridged molodensky transformation.*
  - class [MolodenskiiTransformationStandard](#)
    - Algorithmic realization of the molodensky transformation.*
  - class [SpatialSimilarityTransformationInfin](#)
    - Algorithmic realization of the 3D similarity transformation also known as 7-parameter transformation.*
  - class [SpatialSimilarityTransformationTrig](#)
  - class [TransformationStrategy](#)
    - abstract class that encapsulates transformation algorithms for different coordinates*
    - the concrete transformation algorithms that are implemented by the child classes of [TransformationStrategy](#) implement the abstract method [transform\(GeographicCoordinateInterface\)](#) that performs the transformation with geographic coordinates as input/output*

## 7.3 Paket params

### Klassen

- class [ControlParms](#)
  - Class [ControlParms](#) extends class [EllipsoidParms](#).
- class [EllipsoidParms](#)
  - Ellipsoid parameters.*
- class [LatParm](#)
  - Latitude Parameters.*
- class [MolodenskyParm](#)
  - Molodensky Parameters.*
- class [SpatialSimilarityTransformParm](#)
  - Spatial similarity transform parameters.*

## 7.4 Paket test

### Pakete

- package [coordinates](#)
- package [datumstrategy](#)
- package [params](#)

## 7.5 Paket test.coordinates

### Klassen

- class [AllTests](#)
- class [CoordinateFactoryTest](#)
  - Coordinate factory for plane coordinates and geographic coordinates.*
- class [GaussKruegerTest](#)
  - Test class with unit tests to test behavior class [GaussKrueger](#).*
- class [GaussTest](#)
  - Test class with unit tests to test behavior class [Gauss](#).*
- class [GeographicCoordinateTest](#)
- class [UTMTest](#)
- class [XYZCoordinateTest](#)

## 7.6 Paket test.datumstrategy

### Klassen

- class [GetAppropriateTransformationAlgorithmTest](#)
- class [MockStrategyTest](#)  
*Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used.*
- class [MolodenskiiTransformationAbridgedTest](#)  
*Test of the algorithmic realization of the abridged molodensky transformation.*
- class [MolodenskiiTransformationStandardTest](#)  
*Tests the Molodenskii [Transformation](#) Standard as a Datum Strategy. Input values and expected values were calculated by hand.*
- class [SpatialSimilarityTransformationInfinTest](#)  
*Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.*
- class [SpatialSimilarityTransformationTrigTest](#)  
*Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.*

## 7.7 Paket test.params

### Klassen

- class [LatParmTest](#)



## Kapitel 8

# Klassen-Dokumentation

### 8.1 test.coordinates.AllTests Klassenreferenz

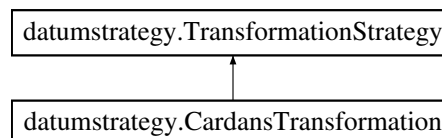
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/coordinates/[AllTests.java](#)

### 8.2 datumstrategy.CardansTransformation Klassenreferenz

Algorithmic realization of Translation and the Cardan's Rotation.

Klassendiagramm für datumstrategy.CardansTransformation:



#### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*Use Cardan's Rotation to transform the geographic coordinate.*

#### Private Attribute

- double [e1](#)
- double [e2](#)
- double [e3](#)
- double [e4](#)
- double [e5](#)
- double [e6](#)
- double [e7](#)
- double [e8](#)
- double [e9](#)

## 8.2.1 Ausführliche Beschreibung

Algorithmic realization of Translation and the Cardan's Rotation.

The method takes all needed parameters from the class ControlParms that extends the class EllipsoidParms. Needed parameter for the [Transformation](#): 3 translation-, 1 scale- and 3 rotation -parameters. The start coordinates are taken from the class GeographicCoordinate and have to be converted into cartesian coordinates to realize the transformation. The Cardan's Rotation is done with trigonometric function. After the [Transformation](#), the new coordinates are converted into ellipsoidal coordinates by an iterative process.

source of cardan's rotation: <http://www.itm.uni-stuttgart.de/courses/madyn/Merkblaetter/M08.pdf> (in Germany) Concept explanation : <http://n.ethz.ch/~roclaudi/download/4.Semster/glossarium%20h%F7ge.pdf>

### Autor

Yucheng Luo

### Bemerkungen

Cardan's rotation matrix added on 01.02.2018 by Yucheng Luo

### Version

0.1

## 8.2.2 Dokumentation der Elementfunktionen

### 8.2.2.1 transform()

```
void datumstrategy.CardansTransformation.transform (
    GeographicCoordinateInterface geo )
```

Use Cardan's Rotation to transform the geographic coordinate.

Translation : Cartesian coordinates are used. Rotation : rotate with principle of conventional Cardan's Rotation ( $W_x, W_y, W_z$ ). Algorithm explanation : Rotate first about X-axis ,after first step rotate about current Y-axis ,at end rotate about current Z-axis.

### Autor

Yucheng Luo

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

## 8.2.3 Dokumentation der Datenelemente

**8.2.3.1 e1**

```
double datumstrategy.CardansTransformation.e1 [private]
```

**8.2.3.2 e2**

```
double datumstrategy.CardansTransformation.e2 [private]
```

**8.2.3.3 e3**

```
double datumstrategy.CardansTransformation.e3 [private]
```

**8.2.3.4 e4**

```
double datumstrategy.CardansTransformation.e4 [private]
```

**8.2.3.5 e5**

```
double datumstrategy.CardansTransformation.e5 [private]
```

**8.2.3.6 e6**

```
double datumstrategy.CardansTransformation.e6 [private]
```

**8.2.3.7 e7**

```
double datumstrategy.CardansTransformation.e7 [private]
```

**8.2.3.8 e8**

```
double datumstrategy.CardansTransformation.e8 [private]
```

### 8.2.3.9 e9

```
double datumstrategy.CardansTransformation.e9 [private]
```

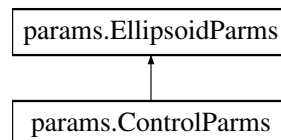
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[CardansTransformation.java](#)

## 8.3 params.ControlParms Klassenreferenz

Class [ControlParms](#) extends class [EllipsoidParms](#).

Klassendiagramm für params.ControlParms:



### Öffentliche Methoden

- void [print](#) ()  
*print outputs of class [params.ControlParms](#)*
- String [getPointid](#) ()  
*getter method that returns the id of the particular Coordinate point*
- void [setPointid](#) (String [pointid](#))  
*setter method that accepts id for the particular Coordinate point*
- long [getPoint\\_number](#) ()  
*getter method that returns point number of the particular Coordinate point*
- void [setPoint\\_number](#) (long [point\\_number](#))  
*setter method that accepts point number for the particular Coordinate point*
- String [getFromprojection](#) ()
- void [setFromprojection](#) (String [fromprojection](#))
- String [getFromdatum](#) ()
- void [setFromdatum](#) (String [fromdatum](#))
- double [getFromawert](#) ()
- void [setFromawert](#) (double [fromawert](#))
- double [getFrombwert](#) ()
- void [setFrombwert](#) (double [frombwert](#))
- double [getSourceCoordinateX](#) ()
- void [setSourceCoordinateX](#) (double [sourceCoordinateX](#))
- double [getSourceCoordinateY](#) ()
- void [setSourceCoordinateY](#) (double [sourceCoordinateY](#))
- double [getSourceCoordinateZ](#) ()
- void [setSourceCoordinateZ](#) (double [sourceCoordinateZ](#))
- String [getTopprojection](#) ()
- void [setTopprojection](#) (String [topprojection](#))
- String [getTodatum](#) ()
- void [setTodatum](#) (String [todatum](#))



- double [getToawert](#) ()
- void [setToawert](#) (double [toawert](#))
- double [getTobwert](#) ()
- void [setTobwert](#) (double [tobwert](#))
- boolean [isChangedatum](#) ()
- void [setChangedatum](#) (boolean [changedatum](#))
- String [getNorthhem](#) ()
- void [setNorthhem](#) (String [northhem](#))
- String [getKindofhoehe](#) ()
- void [setKindofhoehe](#) (String [kindofhoehe](#))
- String [getKindoftrafo](#) ()
- void [setKindoftrafo](#) (String [kindoftrafo](#))
- double [getDx](#) ()
- void [setDx](#) (double [dx](#))
- double [getDy](#) ()
- void [setDy](#) (double [dy](#))
- double [getDz](#) ()
- void [setDz](#) (double [dz](#))
- double [getDa](#) ()
- void [setDa](#) (double [da](#))
- double [getDf](#) ()
- void [setDf](#) (double [df](#))
- double [getWx](#) ()
- void [setWx](#) (double [wx](#))
- double [getWy](#) ()
- void [setWy](#) (double [wy](#))
- double [getWz](#) ()
- void [setWz](#) (double [wz](#))
- double [getRz1](#) ()
- void [setRz1](#) (double [Rz1](#))
- double [getRx2](#) ()
- void [setRx2](#) (double [Rx2](#))
- double [getRz3](#) ()
- void [setRz3](#) (double [Rz3](#))
- double [getMassstab](#) ()
- void [setMassstab](#) (double [massstab](#))
- double [getDestinationCoordinateX](#) ()
- void [setDestinationCoordinateX](#) (double [destinationCoordinateX](#))
- double [getDestinationCoordinateY](#) ()
- void [setDestinationCoordinateY](#) (double [destinationCoordinateY](#))
- double [getDestinationCoordinateZ](#) ()
- void [setDestinationCoordinateZ](#) (double [destinationCoordinateZ](#))
- String [getFileinput](#) ()
- void [setFileinput](#) (String [fileinput](#))
- String [getFileoutput](#) ()
- void [setFileoutput](#) (String [fileoutput](#))
- String [getQupkt](#) ()
- void [setQupkt](#) (String [qupkt](#))
- String [getQutausch](#) ()
- void [setQutausch](#) (String [qutausch](#))
- String [getAusgabetyt](#) ()
- void [setAusgabetyt](#) (String [ausgabetyt](#))
- String [getZipkt](#) ()
- void [setZipkt](#) (String [zipkt](#))
- String [getZihoehe](#) ()
- void [setZihoehe](#) (String [zihoehe](#))
- String [getZitausch](#) ()
- void [setZitausch](#) (String [zitausch](#))

## Öffentliche, statische Methoden

- static [ControlParms getInstance](#) ()  
*Singleton getInstance.*

## Private Methoden

- [ControlParms](#) ()  
*private constructor for singleton*

## Private Attribute

- String [pointid](#)
- long [point\\_number](#)
- String [fromprojection](#)
- String [fromdatum](#)
- double [fromawert](#)
- double [frombwert](#)
- double [sourceCoordinateX](#)
- double [sourceCoordinateY](#)
- double [sourceCoordinateZ](#)
- String [toprojection](#)
- String [todatum](#)
- double [toawert](#)
- double [tobwert](#)
- boolean [changedatum](#)
- String [northhem](#)
- String [kindofhoehe](#)
- String [kindoftrafo](#)
- double [dx](#)
- double [dy](#)
- double [dz](#)
- double [da](#)
- double [df](#)
- double [wx](#)
- double [wy](#)
- double [wz](#)
- double [Rz1](#)
- double [Rx2](#)
- double [Rz3](#)
- double [massstab](#)
- double [destinationCoordinateX](#)
- double [destinationCoordinateY](#)
- double [destinationCoordinateZ](#)
- String [fileinput](#)
- String [fileoutput](#)
- String [qupkt](#)
- String [qutausch](#)
- String [ausgabety](#)
- String [zipkt](#)
- String [zihoehe](#)
- String [zitausch](#)

## Statische, private Attribute

- static [ControlParms instance](#) = null

### 8.3.1 Ausführliche Beschreibung

Class [ControlParms](#) extends class [EllipsoidParms](#).

#### Bemerkungen

last refactored 11.12.2017 by Eva Majer

- adaption for [ControlParms](#) Singleton
- Translation of some comments
- Translation of print method output

add some comments on 2018-01-06 by Yunhao Huang

### 8.3.2 Beschreibung der Konstruktoren und Destruktoren

#### 8.3.2.1 ControlParms()

```
params.ControlParms.ControlParms ( ) [private]
```

private constructor for singleton

### 8.3.3 Dokumentation der Elementfunktionen

#### 8.3.3.1 getAusgabety()

```
String params.ControlParms.getAusgabety ( )
```

#### 8.3.3.2 getDa()

```
double params.ControlParms.getDa ( )
```

#### 8.3.3.3 getDestinationCoordinateX()

```
double params.ControlParms.getDestinationCoordinateX ( )
```

#### 8.3.3.4 getDestinationCoordinateY()

```
double params.ControlParms.getDestinationCoordinateY ( )
```

#### 8.3.3.5 getDestinationCoordinateZ()

```
double params.ControlParms.getDestinationCoordinateZ ( )
```

#### 8.3.3.6 getDf()

```
double params.ControlParms.getDf ( )
```

#### 8.3.3.7 getDx()

```
double params.ControlParms.getDx ( )
```

#### 8.3.3.8 getDy()

```
double params.ControlParms.getDy ( )
```

#### 8.3.3.9 getDz()

```
double params.ControlParms.getDz ( )
```

#### 8.3.3.10 getFileinput()

```
String params.ControlParms.getFileinput ( )
```

**8.3.3.11 getFileoutput()**

```
String params.ControlParms.getFileoutput ( )
```

**8.3.3.12 getFromawert()**

```
double params.ControlParms.getFromawert ( )
```

**8.3.3.13 getFrombwert()**

```
double params.ControlParms.getFrombwert ( )
```

**8.3.3.14 getFromdatum()**

```
String params.ControlParms.getFromdatum ( )
```

**8.3.3.15 getFromprojection()**

```
String params.ControlParms.getFromprojection ( )
```

**8.3.3.16 getInstance()**

```
static ControlParms params.ControlParms.getInstance ( ) [static]
```

Singleton getinstance.

**8.3.3.17 getKindofhoehe()**

```
String params.ControlParms.getKindofhoehe ( )
```

#### 8.3.3.18 getKindoftrafo()

```
String params.ControlParms.getKindoftrafo ( )
```

#### 8.3.3.19 getMassstab()

```
double params.ControlParms.getMassstab ( )
```

#### 8.3.3.20 getNorthhem()

```
String params.ControlParms.getNorthhem ( )
```

#### 8.3.3.21 getPoint\_number()

```
long params.ControlParms.getPoint_number ( )
```

getter method that returns point number of the particular Coordinate point

##### Rückgabe

point\_number - returns the point number of the particular Coordinate point as long value

#### 8.3.3.22 getPointid()

```
String params.ControlParms.getPointid ( )
```

getter method that returns the id of the particular Coordinate point

##### Rückgabe

pointid - string representing the point identification number

#### 8.3.3.23 getQupkt()

```
String params.ControlParms.getQupkt ( )
```

**8.3.3.24 getQutausch()**

```
String params.ControlParms.getQutausch ( )
```

**8.3.3.25 getRx2()**

```
double params.ControlParms.getRx2 ( )
```

**8.3.3.26 getRz1()**

```
double params.ControlParms.getRz1 ( )
```

**8.3.3.27 getRz3()**

```
double params.ControlParms.getRz3 ( )
```

**8.3.3.28 getSourceCoordinateX()**

```
double params.ControlParms.getSourceCoordinateX ( )
```

**8.3.3.29 getSourceCoordinateY()**

```
double params.ControlParms.getSourceCoordinateY ( )
```

**8.3.3.30 getSourceCoordinateZ()**

```
double params.ControlParms.getSourceCoordinateZ ( )
```

**8.3.3.31 getToawert()**

```
double params.ControlParms.getToawert ( )
```

**8.3.3.32 getTobwert()**

```
double params.ControlParms.getTobwert ( )
```

**8.3.3.33 getTodatum()**

```
String params.ControlParms.getTodatum ( )
```

**8.3.3.34 getTopprojection()**

```
String params.ControlParms.getTopprojection ( )
```

**8.3.3.35 getWx()**

```
double params.ControlParms.getWx ( )
```

**8.3.3.36 getWy()**

```
double params.ControlParms.getWy ( )
```

**8.3.3.37 getWz()**

```
double params.ControlParms.getWz ( )
```

**8.3.3.38 getZihoehe()**

```
String params.ControlParms.getZihoehe ( )
```

**8.3.3.39 getZipkt()**

```
String params.ControlParms.getZipkt ( )
```



#### 8.3.3.40 getZitausch()

```
String params.ControlParms.getZitausch ( )
```

#### 8.3.3.41 isChangedatum()

```
boolean params.ControlParms.isChangedatum ( )
```

#### 8.3.3.42 print()

```
void params.ControlParms.print ( )
```

print outputs of class [params.ControlParms](#)

##### Autor

unknown

##### Bemerkungen

updated header 2018-01-06 by Yunhao Huang

Erneute Implementation von [params.EllipsoidParms](#).

#### 8.3.3.43 setAusgabetyt()

```
void params.ControlParms.setAusgabetyt (
    String ausgabetyt )
```

#### 8.3.3.44 setChangedatum()

```
void params.ControlParms.setChangedatum (
    boolean changedatum )
```

#### 8.3.3.45 setDa()

```
void params.ControlParms.setDa (
    double da )
```

**8.3.3.46 setDestinationCoordinateX()**

```
void params.ControlParms.setDestinationCoordinateX (
    double destinationCoordinateX )
```

**8.3.3.47 setDestinationCoordinateY()**

```
void params.ControlParms.setDestinationCoordinateY (
    double destinationCoordinateY )
```

**8.3.3.48 setDestinationCoordinateZ()**

```
void params.ControlParms.setDestinationCoordinateZ (
    double destinationCoordinateZ )
```

**8.3.3.49 setDf()**

```
void params.ControlParms.setDf (
    double df )
```

**8.3.3.50 setDx()**

```
void params.ControlParms.setDx (
    double dx )
```

**8.3.3.51 setDy()**

```
void params.ControlParms.setDy (
    double dy )
```

**8.3.3.52 setDz()**

```
void params.ControlParms.setDz (
    double dz )
```

**8.3.3.53 setFileinput()**

```
void params.ControlParms.setFileinput (
    String fileinput )
```

**8.3.3.54 setFileoutput()**

```
void params.ControlParms.setFileoutput (
    String fileoutput )
```

**8.3.3.55 setFromawert()**

```
void params.ControlParms.setFromawert (
    double fromawert )
```

**8.3.3.56 setFrombwert()**

```
void params.ControlParms.setFrombwert (
    double frombwert )
```

**8.3.3.57 setFromdatum()**

```
void params.ControlParms.setFromdatum (
    String fromdatum )
```

**8.3.3.58 setFromprojection()**

```
void params.ControlParms.setFromprojection (
    String fromprojection )
```

**8.3.3.59 setKindofhoehe()**

```
void params.ControlParms.setKindofhoehe (
    String kindofhoehe )
```

#### 8.3.3.60 setKindoftrafo()

```
void params.ControlParms.setKindoftrafo (
    String kindoftrafo )
```

#### 8.3.3.61 setMassstab()

```
void params.ControlParms.setMassstab (
    double massstab )
```

#### 8.3.3.62 setNorthhem()

```
void params.ControlParms.setNorthhem (
    String northhem )
```

#### 8.3.3.63 setPoint\_number()

```
void params.ControlParms.setPoint_number (
    long point_number )
```

setter method that accepts point number for the particular Coordinate point

Parameter

<i>point_number</i>	- a long value representing the point number for the particular Coordinate point
---------------------	--

#### 8.3.3.64 setPointid()

```
void params.ControlParms.setPointid (
    String pointid )
```

setter method that accepts id for the particular Coordinate point

Parameter

<i>pointid</i>	- string representing the point identification number
----------------	---

**8.3.3.65 setQupkt()**

```
void params.ControlParms.setQupkt (
    String qupkt )
```

**8.3.3.66 setQutausch()**

```
void params.ControlParms.setQutausch (
    String qutausch )
```

**8.3.3.67 setRx2()**

```
void params.ControlParms.setRx2 (
    double Rx2 )
```

**8.3.3.68 setRz1()**

```
void params.ControlParms.setRz1 (
    double Rz1 )
```

**8.3.3.69 setRz3()**

```
void params.ControlParms.setRz3 (
    double Rz3 )
```

**8.3.3.70 setSourceCoordinateX()**

```
void params.ControlParms.setSourceCoordinateX (
    double sourceCoordinateX )
```

**8.3.3.71 setSourceCoordinateY()**

```
void params.ControlParms.setSourceCoordinateY (
    double sourceCoordinateY )
```

**8.3.3.72 setSourceCoordinateZ()**

```
void params.ControlParms.setSourceCoordinateZ (
    double sourceCoordinateZ )
```

**8.3.3.73 setToawert()**

```
void params.ControlParms.setToawert (
    double toawert )
```

**8.3.3.74 setTobwert()**

```
void params.ControlParms.setTobwert (
    double tobwert )
```

**8.3.3.75 setTodatum()**

```
void params.ControlParms.setTodatum (
    String todatum )
```

**8.3.3.76 setToprojection()**

```
void params.ControlParms.setToprojection (
    String toprojection )
```

**8.3.3.77 setWx()**

```
void params.ControlParms.setWx (
    double wx )
```

**8.3.3.78 setWy()**

```
void params.ControlParms.setWy (
    double wy )
```

**8.3.3.79 setWz()**

```
void params.ControlParms.setWz (
    double wz )
```

**8.3.3.80 setZihoehe()**

```
void params.ControlParms.setZihoehe (
    String zihoehe )
```

**8.3.3.81 setZipkt()**

```
void params.ControlParms.setZipkt (
    String zipkt )
```

**8.3.3.82 setZitausch()**

```
void params.ControlParms.setZitausch (
    String zitausch )
```

**8.3.4 Dokumentation der Datenelemente****8.3.4.1 ausgabetyt**

```
String params.ControlParms.ausgabetyt [private]
```

**8.3.4.2 changedatum**

```
boolean params.ControlParms.changedatum [private]
```

True/False means with/without datum changement

**8.3.4.3 da**

```
double params.ControlParms.da [private]
```

Difference in large semi axis (For Molodenskii-Transformation)

#### 8.3.4.4 destinationCoordinateX

```
double params.ControlParms.destinationCoordinateX [private]
```

Currently not used

#### 8.3.4.5 destinationCoordinateY

```
double params.ControlParms.destinationCoordinateY [private]
```

Currently not used

#### 8.3.4.6 destinationCoordinateZ

```
double params.ControlParms.destinationCoordinateZ [private]
```

Currently not used

#### 8.3.4.7 df

```
double params.ControlParms.df [private]
```

Difference flattening (For Molodenskii-Transformation)

#### 8.3.4.8 dx

```
double params.ControlParms.dx [private]
```

Translation X (For Molodenskii und 3D similarity transformation)

#### 8.3.4.9 dy

```
double params.ControlParms.dy [private]
```

Translation Y (For Molodenskii und 3D similarity transformation)

#### 8.3.4.10 dz

```
double params.ControlParms.dz [private]
```

Translation Z (For Molodenskii und 3D similarity transformation)

#### 8.3.4.11 fileinput

```
String params.ControlParms.fileinput [private]
```



**8.3.4.12 fileoutput**

```
String params.ControlParms.fileoutput [private]
```

**8.3.4.13 fromawert**

```
double params.ControlParms.fromawert [private]
```

**8.3.4.14 frombwert**

```
double params.ControlParms.frombwert [private]
```

**8.3.4.15 fromdatum**

```
String params.ControlParms.fromdatum [private]
```

**8.3.4.16 fromprojection**

```
String params.ControlParms.fromprojection [private]
```

**8.3.4.17 instance**

```
ControlParms params.ControlParms.instance = null [static], [private]
```

Singleton

**8.3.4.18 kindofhoehe**

```
String params.ControlParms.kindofhoehe [private]
```

Height calculation

**8.3.4.19 kindoftrafo**

```
String params.ControlParms.kindoftrafo [private]
```

Kind of transformation (e.g. 3D similarity transformation)

**8.3.4.20 massstab**

```
double params.ControlParms.massstab [private]
```

Scale (For 3D similarity transformation)

**8.3.4.21 northhem**

```
String params.ControlParms.northhem [private]
```

Hemisphere

**8.3.4.22 point\_number**

```
long params.ControlParms.point_number [private]
```

**8.3.4.23 pointid**

```
String params.ControlParms.pointid [private]
```

**8.3.4.24 qupkt**

```
String params.ControlParms.qupkt [private]
```

**8.3.4.25 qutausch**

```
String params.ControlParms.qutausch [private]
```

**8.3.4.26 Rx2**

```
double params.ControlParms.Rx2 [private]
```

second Rotation about current X (For Eulerstransformation)

**8.3.4.27 Rz1**

```
double params.ControlParms.Rz1 [private]
```

first Rotation about current Z (For Eulerstransformation)

**8.3.4.28 Rz3**

```
double params.ControlParms.Rz3 [private]
```

third Rotation about current Z (For Eulerstransformation)

**8.3.4.29 sourceCoordinateX**

```
double params.ControlParms.sourceCoordinateX [private]
```

Input coordinate X

**8.3.4.30 sourceCoordinateY**

```
double params.ControlParms.sourceCoordinateY [private]
```

Input coordinate Y

**8.3.4.31 sourceCoordinateZ**

```
double params.ControlParms.sourceCoordinateZ [private]
```

Input coordinate Z

**8.3.4.32 toawert**

```
double params.ControlParms.toawert [private]
```

**8.3.4.33 tobwert**

```
double params.ControlParms.tobwert [private]
```

**8.3.4.34 todatum**

```
String params.ControlParms.todatum [private]
```

**8.3.4.35 topjection**

```
String params.ControlParms.topjection [private]
```

#### 8.3.4.36 wx

```
double params.ControlParms.wx [private]
```

Rotation X (For 3D similarity transformation)

#### 8.3.4.37 wy

```
double params.ControlParms.wy [private]
```

Rotation Y (For 3D similarity transformation)

#### 8.3.4.38 wz

```
double params.ControlParms.wz [private]
```

Rotation Z (For 3D similarity transformation)

#### 8.3.4.39 zihoehe

```
String params.ControlParms.zihoehe [private]
```

#### 8.3.4.40 zipkt

```
String params.ControlParms.zipkt [private]
```

#### 8.3.4.41 zitausch

```
String params.ControlParms.zitausch [private]
```

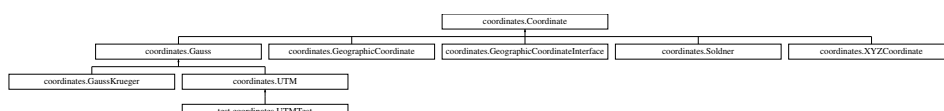
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [params/ControlParms.java](#)

## 8.4 coordinates.Coordinate Klassenreferenz

Abstract class that represents a coordinate. Enables any concrete [Coordinate](#) class be used as input for TransformationStrategy which relies on geographic coordinates as input/output

Klassendiagramm für coordinates.Coordinate:



## Öffentliche Methoden

- abstract [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ellipsoidParms)  
*abstract method that allows the concrete [Coordinate](#) instance to be convertible to [GeographicCoordinate](#)*
- abstract void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ellipsoidParms, [GeographicCoordinateInterface](#) geographicCoordinate)  
*abstract method that allows the concrete [Coordinate](#) instance to be convertible from [GeographicCoordinate](#)*
- abstract void [print](#) ()  
*abstract method that allows the concrete [Coordinate](#) to be output on screen*
- String [getPointid](#) ()  
*getter method that returns the id of the particular [Coordinate](#) point*
- void [setPointid](#) (String pointid)  
*setter method that accepts id for the particular [Coordinate](#) point*

## Private Attribute

- String [pointid](#)

### 8.4.1 Ausführliche Beschreibung

Abstract class that represents a coordinate. Enables any concrete [Coordinate](#) class be used as input for TransformationStrategy which relies on geographic coordinates as input/output

#### Autor

unknown

#### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

renamed to [Coordinate](#) (was Coordinates) on 2016-11-26 by Patrick Huebner

added abstract methods [getAsGeographic](#) and [fromGeographic](#) to enable any concrete [Coordinate](#) class be used as input for TransformationStrategy which relies on geographic coordinates as input/output on 2016-11-26 by Patrick Huebner

#### Version

0.1

#### Parameter

<i>pointid</i>	- string representing the point identification number
----------------	---

### 8.4.2 Dokumentation der Elementfunktionen

#### 8.4.2.1 getAsGeographicInterface()

```
abstract GeographicCoordinateInterface coordinates.Coordinate.getAsGeographicInterface (
    EllipsoidParms ellipsoidParms ) [abstract]
```

abstract method that allows the concrete [Coordinate](#) instance to be convertible to [GeographicCoordinate](#)

##### Parameter

<i>ellipsoidParms</i>	- an object of type <a href="#">EllipsoidParms</a> that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> should refer to
-----------------------	---

Erneute Implementation in [coordinates.GeographicCoordinateInterface](#), [coordinates.GeographicCoordinate](#), [coordinates.Soldner](#), [coordinates.XYZCoordinate](#), [coordinates.GaussKrueger](#), [coordinates.UTM](#) und [coordinates.Gauss](#).

#### 8.4.2.2 getAsTargetCoordinate()

```
abstract void coordinates.Coordinate.getAsTargetCoordinate (
    EllipsoidParms ellipsoidParms,
    GeographicCoordinateInterface geographicCoordinate ) [abstract]
```

abstract method that allows the concrete [Coordinate](#) instance to be convertible from [GeographicCoordinate](#)

##### Parameter

<i>geographicCoordinate</i>	- an object of type <a href="#">GeographicCoordinate</a> that represents the geographic coordinate the concrete <a href="#">Coordinate</a> instance should be converted from
<i>ellipsoidParms</i>	- an object of type <a href="#">EllipsoidParms</a> that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> refers to

Erneute Implementation in [coordinates.GeographicCoordinateInterface](#), [coordinates.GeographicCoordinate](#), [coordinates.XYZCoordinate](#), [coordinates.Soldner](#), [coordinates.GaussKrueger](#) und [coordinates.UTM](#).

#### 8.4.2.3 getPointid()

```
String coordinates.Coordinate.getPointid ( )
```

getter method that returns the id of the particular [Coordinate](#) point

##### Rückgabe

pointid - string representing the point identification number

#### 8.4.2.4 print()

```
abstract void coordinates.Coordinate.print ( ) [abstract]
```

abstract methnd that allows the concrete [Coordinate](#) to be output on screen

Erneute Implementation in [coordinates.GeographicCoordinateInterface](#), [coordinates.XYZCoordinate](#), [coordinates.GeographicCoordinate](#), [coordinates.Soldner](#), [coordinates.Gauss](#), [coordinates.GaussKrueger](#) und [coordinates.UTM](#).

#### 8.4.2.5 setPointid()

```
void coordinates.Coordinate.setPointid (
    String pointid )
```

setter method that accepts id for the particular [Coordinate](#) point

Parameter

<i>pointid</i>	- string representing the point identification number
----------------	---

### 8.4.3 Dokumentation der Datenelemente

#### 8.4.3.1 pointid

```
String coordinates.Coordinate.pointid [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/Coordinate.java](#)

## 8.5 test.coordinates.CoordinateFactoryTest Klassenreferenz

Coordinate factory for plane coordinates and geographic coordinates.

### Öffentliche Methoden

- void [testGetCoordGK](#) ()
- void [testGetCoordUTM](#) ()
- void [testGetCoordSoldner](#) ()
- void [testGetCoordGeographic](#) ()
- void [testGetXYZ](#) ()

### 8.5.1 Ausführliche Beschreibung

Coordinate factory for plane coordinates and geographic coordinates.

#### Autor

Norbert Roesch

#### Bemerkungen

January 2019 ensure the factory produces the right objects

#### Version

0.1

### 8.5.2 Dokumentation der Elementfunktionen

#### 8.5.2.1 testGetCoordGeographic()

```
void test.coordinates.CoordinateFactoryTest.testGetCoordGeographic ( )
```

#### 8.5.2.2 testGetCoordGK()

```
void test.coordinates.CoordinateFactoryTest.testGetCoordGK ( )
```

#### 8.5.2.3 testGetCoordSoldner()

```
void test.coordinates.CoordinateFactoryTest.testGetCoordSoldner ( )
```

#### 8.5.2.4 testGetCoordUTM()

```
void test.coordinates.CoordinateFactoryTest.testGetCoordUTM ( )
```



### 8.5.2.5 testGetXYZ()

```
void test.coordinates.CoordinateFactoryTest.testGetXYZ ( )
```

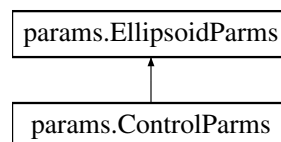
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/coordinates/[CoordinateFactoryTest.java](#)

## 8.6 params.EllipsoidParms Klassenreferenz

Ellipsoid parameters.

Klassendiagramm für params.EllipsoidParms:



### Öffentliche Methoden

- [EllipsoidParms](#) ()  
*Constructor for the class [params.EllipsoidParms](#).*
- [EllipsoidParms](#) (double es2, double c)  
*General constructor for the class [params.EllipsoidParms](#).*
- void [calculatelongitudeParms](#) ([GeographicCoordinateInterface](#) geo)
- void [print](#) ()
- String [getId](#) ()
- void [setId](#) (String id)
- double [getC](#) ()
- void [setC](#) (double c)
- double [getSecondEccentricity](#) ()
- void [setSecondEccentricity](#) (double es2)
- double [getSemiMajorAxis](#) ()
- void [setSemiMajorAxis](#) (double a)
- double [getSemiMinorAxis](#) ()
- void [setSemiMinorAxis](#) (double b)
- double [getFlattening](#) ()
- void [setFlattening](#) (double f)
- double [getFirstEccentricity](#) ()
- void [setFirstEccentricity](#) (double e2)
- String [getToid](#) ()
- void [setToid](#) (String toid)
- double [getToc](#) ()
- void [setToc](#) (double toc)
- double [getToes2](#) ()
- void [setToes2](#) (double toes2)
- double [getToa](#) ()
- void [setToa](#) (double toa)

- double [getTob](#) ()
- void [setTob](#) (double [tob](#))
- double [getTof](#) ()
- void [setTof](#) (double [tof](#))
- double [getToe2](#) ()
- void [setToe2](#) (double [toe2](#))
- double [getN](#) ()
- void [setN](#) (double [n](#))
- double [getM](#) ()
- void [setM](#) (double [m](#))
- double [getSoldner\\_I0](#) ()
- void [setSoldner\\_I0](#) (double [soldner\\_I0](#))
- double [getSoldner\\_b0](#) ()
- void [setSoldner\\_b0](#) (double [soldner\\_b0](#))
- double [getGK\\_refmer](#) ()
- void [setGK\\_refmer](#) (double [GK\\_refmer](#))

### Private Attribute

- String [id](#)
- double [c](#)
- double [secondEccentricity](#)
- double [semiMajorAxis](#)
- double [semiMinorAxis](#)
- double [flattening](#)
- double [firstEccentricity](#)
- String [toid](#)
- double [toc](#)
- double [toes2](#)
- double [toa](#)
- double [tob](#)
- double [tof](#)
- double [toe2](#)
- double [N](#)
- double [M](#)
- double [Soldner\\_I0](#)
- double [Soldner\\_b0](#)
- double [GK\\_refmer](#)

### 8.6.1 Ausführliche Beschreibung

Ellipsoid parameters.

#### Autor

unknown

#### Bemerkungen

add header and some comments on 2018-01-06 by Yunhao Huang

## 8.6.2 Beschreibung der Konstruktoren und Destruktoren

### 8.6.2.1 EllipsoidParms() [1/2]

```
params.EllipsoidParms.EllipsoidParms ( )
```

Constructor for the class [params.EllipsoidParms](#).

### 8.6.2.2 EllipsoidParms() [2/2]

```
params.EllipsoidParms.EllipsoidParms (
    double es2,
    double c )
```

General constructor for the class [params.EllipsoidParms](#).

Transfer of two shape parameters

## 8.6.3 Dokumentation der Elementfunktionen

### 8.6.3.1 calculatelongitudeParms()

```
void params.EllipsoidParms.calculatelongitudeParms (
    GeographicCoordinateInterface geo )
```

### 8.6.3.2 getC()

```
double params.EllipsoidParms.getC ( )
```

### 8.6.3.3 getFirstEccentricity()

```
double params.EllipsoidParms.getFirstEccentricity ( )
```

#### 8.6.3.4 getFlattening()

```
double params.EllipsoidParms.getFlattening ( )
```

#### 8.6.3.5 getGK\_refmer()

```
double params.EllipsoidParms.getGK_refmer ( )
```

#### 8.6.3.6 getId()

```
String params.EllipsoidParms.getId ( )
```

#### 8.6.3.7 getM()

```
double params.EllipsoidParms.getM ( )
```

#### 8.6.3.8 getN()

```
double params.EllipsoidParms.getN ( )
```

#### 8.6.3.9 getSecondEccentricity()

```
double params.EllipsoidParms.getSecondEccentricity ( )
```

#### 8.6.3.10 getSemiMajorAxis()

```
double params.EllipsoidParms.getSemiMajorAxis ( )
```

#### 8.6.3.11 getSemiMinorAxis()

```
double params.EllipsoidParms.getSemiMinorAxis ( )
```

**8.6.3.12 getSoldner\_b0()**

```
double params.EllipsoidParms.getSoldner_b0 ( )
```

**8.6.3.13 getSoldner\_l0()**

```
double params.EllipsoidParms.getSoldner_l0 ( )
```

**8.6.3.14 getToa()**

```
double params.EllipsoidParms.getToa ( )
```

**8.6.3.15 getTob()**

```
double params.EllipsoidParms.getTob ( )
```

**8.6.3.16 getToc()**

```
double params.EllipsoidParms.getToc ( )
```

**8.6.3.17 getToe2()**

```
double params.EllipsoidParms.getToe2 ( )
```

**8.6.3.18 getToes2()**

```
double params.EllipsoidParms.getToes2 ( )
```

**8.6.3.19 getTof()**

```
double params.EllipsoidParms.getTof ( )
```

**8.6.3.20 getToid()**

```
String params.EllipsoidParms.getToid ( )
```

**8.6.3.21 print()**

```
void params.EllipsoidParms.print ( )
```

Erneute Implementation in [params.ControlParms](#).

**8.6.3.22 setC()**

```
void params.EllipsoidParms.setC (
    double c )
```

**8.6.3.23 setFirstEccentricity()**

```
void params.EllipsoidParms.setFirstEccentricity (
    double e2 )
```

**8.6.3.24 setFlattening()**

```
void params.EllipsoidParms.setFlattening (
    double f )
```

**8.6.3.25 setGK\_refmer()**

```
void params.EllipsoidParms.setGK_refmer (
    double GK_refmer )
```

**8.6.3.26 setId()**

```
void params.EllipsoidParms.setId (
    String id )
```

**8.6.3.27 setM()**

```
void params.EllipsoidParms.setM (
    double m )
```

**8.6.3.28 setN()**

```
void params.EllipsoidParms.setN (
    double n )
```

**8.6.3.29 setSecondEccentricity()**

```
void params.EllipsoidParms.setSecondEccentricity (
    double es2 )
```

**8.6.3.30 setSemiMajorAxis()**

```
void params.EllipsoidParms.setSemiMajorAxis (
    double a )
```

**8.6.3.31 setSemiMinorAxis()**

```
void params.EllipsoidParms.setSemiMinorAxis (
    double b )
```

**8.6.3.32 setSoldner\_b0()**

```
void params.EllipsoidParms.setSoldner_b0 (
    double soldner_b0 )
```

**8.6.3.33 setSoldner\_l0()**

```
void params.EllipsoidParms.setSoldner_l0 (
    double soldner_l0 )
```

**8.6.3.34 setToa()**

```
void params.EllipsoidParms.setToa (
    double toa )
```

**8.6.3.35 setTob()**

```
void params.EllipsoidParms.setTob (
    double tob )
```

**8.6.3.36 setToc()**

```
void params.EllipsoidParms.setToc (
    double toc )
```

**8.6.3.37 setToe2()**

```
void params.EllipsoidParms.setToe2 (
    double toe2 )
```

**8.6.3.38 setToes2()**

```
void params.EllipsoidParms.setToes2 (
    double toes2 )
```

**8.6.3.39 setTof()**

```
void params.EllipsoidParms.setTof (
    double tof )
```

**8.6.3.40 setToid()**

```
void params.EllipsoidParms.setToid (
    String toid )
```



## 8.6.4 Dokumentation der Datenelemente

### 8.6.4.1 c

```
double params.EllipsoidParms.c [private]
```

Radius of curvature

### 8.6.4.2 firstEccentricity

```
double params.EllipsoidParms.firstEccentricity [private]
```

### 8.6.4.3 flattening

```
double params.EllipsoidParms.flattening [private]
```

### 8.6.4.4 GK\_refmer

```
double params.EllipsoidParms.GK_refmer [private]
```

Parameters for converting geographic coordinates -> [coordinates.GaussKrueger](#)

### 8.6.4.5 id

```
String params.EllipsoidParms.id [private]
```

ID

### 8.6.4.6 M

```
double params.EllipsoidParms.M [private]
```

Radius M (only if necessary)

### 8.6.4.7 N

```
double params.EllipsoidParms.N [private]
```

Radius N (only if necessary)

#### 8.6.4.8 secondEccentricity

```
double params.EllipsoidParms.secondEccentricity [private]
```

#### 8.6.4.9 semiMajorAxis

```
double params.EllipsoidParms.semiMajorAxis [private]
```

#### 8.6.4.10 semiMinorAxis

```
double params.EllipsoidParms.semiMinorAxis [private]
```

#### 8.6.4.11 Soldner\_b0

```
double params.EllipsoidParms.Soldner_b0 [private]
```

Parameters for converting geographic coordinates -> [coordinates.Soldner](#)

#### 8.6.4.12 Soldner\_l0

```
double params.EllipsoidParms.Soldner_l0 [private]
```

Parameters for converting geographic coordinates -> [coordinates.Soldner](#)

#### 8.6.4.13 toa

```
double params.EllipsoidParms.toa [private]
```

Semi-major axis (if change of datum)

#### 8.6.4.14 tob

```
double params.EllipsoidParms.tob [private]
```

Semi-minor axis (if change of datum)

#### 8.6.4.15 toc

```
double params.EllipsoidParms.toc [private]
```

Radius of curvature (if change of datum)

#### 8.6.4.16 toe2

```
double params.EllipsoidParms.toe2 [private]
```

First eccentricity (if change of datum)

#### 8.6.4.17 toes2

```
double params.EllipsoidParms.toes2 [private]
```

Second eccentricity (if change of datum)

#### 8.6.4.18 tof

```
double params.EllipsoidParms.tof [private]
```

Flattening (if change of datum)

#### 8.6.4.19 toid

```
String params.EllipsoidParms.toid [private]
```

ID (if change of datum)

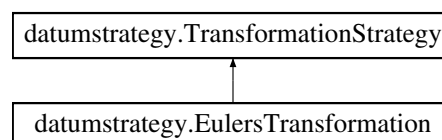
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [params/EllipsoidParms.java](#)

## 8.7 datumstrategy.EulersTransformation Klassenreferenz

Algorithmic realization of the Euler's transformation.

Klassendiagramm für datumstrategy.EulersTransformation:



### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*Use Euler angles to transform the geographic coordinate.*

## Private Attribute

- double [e1](#)
- double [e2](#)
- double [e3](#)
- double [e4](#)
- double [e5](#)
- double [e6](#)
- double [e7](#)
- double [e8](#)
- double [e9](#)

### 8.7.1 Ausführliche Beschreibung

Algorithmic realization of the Euler's transformation.

The method takes all needed parameters from the class `ControlParms` that extends the class `EllipsoidParms`. Needed parameter for the [EulersTransformation](#): 3 translation-, 1 scale- and 3 rotation -parameters. The start coordinates are taken from the class `GeographicCoordinate` and have to be converted into cartesian coordinates to realize the transformation. The Translation is done with cartesian coordinates and the rotation is implemented with trigonometric function. After the [Transformation](#), the new cartesian coordinates are converted into ellipsoidal coordinates by an iterative process.

source of [EulersTransformation](#): [https://en.wikipedia.org/wiki/Euler\\_angles](https://en.wikipedia.org/wiki/Euler_angles)

#### Autor

Yucheng Luo

#### Bemerkungen

Euler's angle added on 01.02.2018 by Yucheng Luo

#### Version

0.1

### 8.7.2 Dokumentation der Elementfunktionen

#### 8.7.2.1 transform()

```
void datumstrategy.EulersTransformation.transform (
    GeographicCoordinateInterface geo )
```

Use Euler angles to transform the geographic coordinate.

Translation : Cartesian coordinates are used. Rotation : rotate with principle of conventional Euler's Standard-x-convention (Rz1,Rx2,Rz3). Algorithm explanation : Rotate first about Z-axis ,after first step rotate about current X-axis, at end rotate about current Z-axis. Link to the Algorithm : [https://de.wikipedia.org/wiki/Eulersche\\_Winkel](https://de.wikipedia.org/wiki/Eulersche_Winkel)

#### Autor

Yucheng Luo

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

### 8.7.3 Dokumentation der Datenelemente

#### 8.7.3.1 e1

```
double datumstrategy.EulersTransformation.e1 [private]
```

#### 8.7.3.2 e2

```
double datumstrategy.EulersTransformation.e2 [private]
```

#### 8.7.3.3 e3

```
double datumstrategy.EulersTransformation.e3 [private]
```

#### 8.7.3.4 e4

```
double datumstrategy.EulersTransformation.e4 [private]
```

#### 8.7.3.5 e5

```
double datumstrategy.EulersTransformation.e5 [private]
```

#### 8.7.3.6 e6

```
double datumstrategy.EulersTransformation.e6 [private]
```

#### 8.7.3.7 e7

```
double datumstrategy.EulersTransformation.e7 [private]
```

### 8.7.3.8 e8

```
double datumstrategy.EulersTransformation.e8 [private]
```

### 8.7.3.9 e9

```
double datumstrategy.EulersTransformation.e9 [private]
```

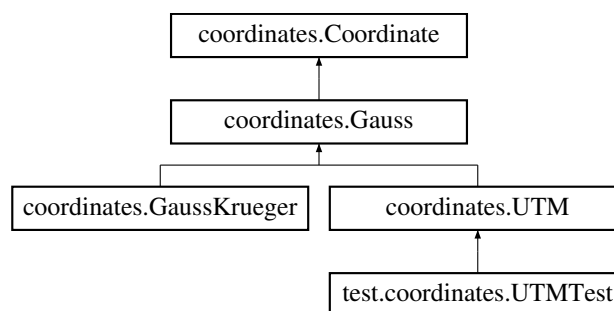
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[EulersTransformation.java](#)

## 8.8 coordinates.Gauss Klassenreferenz

Class [Gauss](#) extends class [Coordinate](#).

Klassendiagramm für coordinates.Gauss:



### Öffentliche Methoden

- abstract void [print](#) ()  
*abstract method that allows the concrete [Coordinate](#) to be output on screen*
- abstract [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ell)  
*abstract method that allows the concrete [Coordinate](#) instance to be convertable to [GeographicCoordinate](#)*
- double [getAbszisse](#) ()  
*getter method that returns the the abscissa of the particular [Gauss](#) coordinate*
- void [setAbszisse](#) (double [abszisse](#))  
*setter method that accepts the abscissa of the particular [Gauss](#) coordinate*
- double [getOrdinate](#) ()  
*getter method that returns the the ordinate of the particular [Gauss](#) coordinate*
- void [setOrdinate](#) (double [ordinate](#))  
*setter method that accepts the ordinate of the particular [Gauss](#) coordinate*
- double [getScale](#) ()  
*getter method that returns the the scale of the particular [Gauss](#) coordinate*
- void [setScale](#) (double [scale](#))  
*setter method that accepts the scale of the particular [Gauss](#) coordinate*

- double `getHeight` ()  
*getter method that returns the the height of the particular `Gauss` coordinate*
- void `setHeight` (double `height`)  
*setter method that accepts the height of the particular `Gauss` coordinate*
- int `getEllipsoidal` ()  
*getter method that returns the the ellipsoidal of the particular `Gauss` coordinate*
- void `setEllipsoidal` (int `ellipsoidal`)  
*setter method that accepts the ellipsoidal of the particular `Gauss` coordinate*

## Öffentliche, statische Methoden

- static double `meridianLength` (double latitude, `EllipsoidParms` ell)

## Geschützte Methoden

- void `CalculateAbszisseOrdinate` (double cdl, double hnull, `LatParm` latparm)  
*CalculateAbszisseOrdinate calculates the abscissa and ordinate.*

## Private Attribute

- double `abszisse`
- double `ordinate`
- double `scale`
- double `height`
- int `ellipsoidal`

## 8.8.1 Ausführliche Beschreibung

Class `Gauss` extends class `Coordinate`.

### Autor

unknown

### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

### Version

0.1

### Parameter

<code>abszisse</code>	- x-value as double
<code>ordinate</code>	- y-value as double
<code>scale</code>	- scale of central median as double
<code>height</code>	- height as double
<code>ellipsoidal</code>	height above ellipsoid as integer

## 8.8.2 Dokumentation der Elementfunktionen

### 8.8.2.1 CalculateAbszisseOrdinate()

```
void coordinates.Gauss.CalculateAbszisseOrdinate (
    double cdl,
    double hnull,
    LatParm latparm ) [protected]
```

CalculateAbszisseOrdinate calculates the abscissa and ordinate.

#### Autor

unknown

#### Bemerkungen

updated header 29.11.2017 by Markus Mueller

#### Parameter

<i>cdl</i>	- a double
<i>hnull</i>	- a double
<i>latparm</i>	- a LatParm object

### 8.8.2.2 getAbszisse()

```
double coordinates.Gauss.getAbszisse ( )
```

getter method that returns the the abscissa of the particular Gauss coordinate

#### Rückgabe

abszisse - returns the abscissa of the particular Gauss coordinate as double

### 8.8.2.3 getAsGeographicInterface()

```
abstract GeographicCoordinateInterface coordinates.Gauss.getAsGeographicInterface (
    EllipsoidParms ellipsoidParms ) [abstract]
```

abstract method that allows the concrete Coordinate instance to be convertable to GeographicCoordinate



**Parameter**

<i>ellipsoidParms</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> should refer to
-----------------------	---

Erneute Implementation von [coordinates.Coordinate](#).

Erneute Implementation in [coordinates.GaussKrueger](#) und [coordinates.UTM](#).

**8.8.2.4 getEllipsoidal()**

```
int coordinates.Gauss.getEllipsoidal ( )
```

getter method that returns the the ellipsoidal of the particular [Gauss](#) coordinate

**Rückgabe**

ellipsoidal - returns the ellipsoidal of the particular [Gauss](#) coordinate as double

**8.8.2.5 getHeight()**

```
double coordinates.Gauss.getHeight ( )
```

getter method that returns the the height of the particular [Gauss](#) coordinate

**Rückgabe**

height - returns the height of the particular [Gauss](#) coordinate as double

**8.8.2.6 getOrdinate()**

```
double coordinates.Gauss.getOrdinate ( )
```

getter method that returns the the ordinate of the particular [Gauss](#) coordinate

**Rückgabe**

ordinate - returns the ordinate of the particular [Gauss](#) coordinate as double

### 8.8.2.7 `getScale()`

```
double coordinates.Gauss.getScale ( )
```

getter method that returns the the scale of the particular [Gauss](#) coordinate

#### Rückgabe

scale - returns the ordinate of the particular [Gauss](#) coordinate as double as double

### 8.8.2.8 `meridianLength()`

```
static double coordinates.Gauss.meridianLength (
    double latitude,
    EllipsoidParms ell ) [static]
```

### 8.8.2.9 `print()`

```
abstract void coordinates.Gauss.print ( ) [abstract]
```

abstract methnd that allows the concrete [Coordinate](#) to be output on screen

Erneute Implementation von [coordinates.Coordinate](#).

Erneute Implementation in [coordinates.GaussKrueger](#) und [coordinates.UTM](#).

### 8.8.2.10 `setAbszisse()`

```
void coordinates.Gauss.setAbszisse (
    double abszisse )
```

setter method that accepts the abscissa of the particular [Gauss](#) coordinate

#### Parameter

<i>abszisse</i>	- sets a double representing the abscissa for the particular <a href="#">Gauss</a> coordinate
-----------------	---

### 8.8.2.11 `setEllipsoidal()`

```
void coordinates.Gauss.setEllipsoidal (
    int ellipsoidal )
```

setter method that accepts the ellipsoidal of the particular [Gauss](#) coordinate

**Parameter**

<i>ellipsoidal</i>	- sets a double representing the ellipsoidal for the particular <a href="#">Gauss</a> coordinate
--------------------	--

### 8.8.2.12 setHeight()

```
void coordinates.Gauss.setHeight (
    double height )
```

setter method that accepts the height of the particular [Gauss](#) coordinate

**Parameter**

<i>height</i>	- sets a double representing the height for the particular <a href="#">Gauss</a> coordinate
---------------	---

### 8.8.2.13 setOrdinate()

```
void coordinates.Gauss.setOrdinate (
    double ordinate )
```

setter method that accepts the ordinate of the particular [Gauss](#) coordinate

**Parameter**

<i>ordinate</i>	- sets a double representing the ordinate for the particular <a href="#">Gauss</a> coordinate
-----------------	---

### 8.8.2.14 setScale()

```
void coordinates.Gauss.setScale (
    double scale )
```

setter method that accepts the scale of the particular [Gauss](#) coordinate

**Parameter**

<i>scale</i>	- sets a double representing the scale for the particular <a href="#">Gauss</a> coordinate
--------------	--

### 8.8.3 Dokumentation der Datenelemente

#### 8.8.3.1 abszisse

```
double coordinates.Gauss.abszisse [private]
```

x-value

#### 8.8.3.2 ellipsoidal

```
int coordinates.Gauss.ellipsoidal [private]
```

Height above ellipsoid:, 0 => no height, 1 => ellipsoidal, 2 => geoid

#### 8.8.3.3 height

```
double coordinates.Gauss.height [private]
```

Height

#### 8.8.3.4 ordinate

```
double coordinates.Gauss.ordinate [private]
```

y-value

#### 8.8.3.5 scale

```
double coordinates.Gauss.scale [private]
```

scale of central meridian

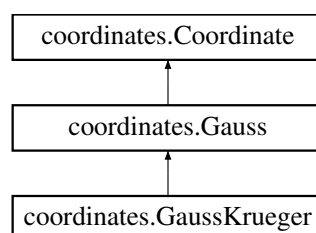
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/Gauss.java](#)

## 8.9 coordinates.GaussKrueger Klassenreferenz

[Coordinate](#) point relating to a certain Gauss-Krueger reference meridian.

Klassendiagramm für coordinates.GaussKrueger:



## Öffentliche Methoden

- [GaussKrueger](#) ()  
*main constructor to set the coordinates and reference meridian*
- void [print](#) ()  
*prints a string representation of the Gauss-Krueger object into the console*
- [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ell)  
*transforms the coordinates of the current [GaussKrueger](#) instance into geographic coordinates*
- void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ell, [GeographicCoordinateInterface](#) geocoord)  
*transforms geographic coordinates into [GaussKrueger](#) coordinates and stores them in the current [GaussKrueger](#) instance*
- double [getRechts](#) ()  
*getter method that returns the the northing of the particular [GaussKrueger](#) instance*
- void [setRechts](#) (double rechts)  
*setter method that accepts the easting of the particular [GaussKrueger](#) instance*
- double [getHoch](#) ()  
*getter method for the northing of the current [GaussKrueger](#) instance*
- void [setHoch](#) (double hoch)  
*setter method that accepts the northing of the particular [GaussKrueger](#) instance*
- int [getIlao](#) ()  
*getter method for the reference meridian of the current [GaussKrueger](#) instance*
- void [setIlao](#) (int ilao)  
*setter method for the number of the reference meridian of the current [GaussKrueger](#) instance*

## Private Attribute

- double rechts
- double hoch
- int centralMeridian

## Weitere Geerbte Elemente

### 8.9.1 Ausführliche Beschreibung

[Coordinate](#) point relating to a certain Gauss-Krueger reference meridian.

#### Autor

unknown

#### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

change comment of attribute 'hoch' 2017-12-19 by Markus Hillemann

change attribute name ilao to centralMeridian 2017-12-19 by Markus Hillemann

#### Version

0.1

## Parameter

<i>rechts</i>	- Easting as double
<i>hoch</i>	- Northing as double
<i>centralMeridian</i>	- Central meridian as integer

## 8.9.2 Beschreibung der Konstruktoren und Destruktoren

### 8.9.2.1 GaussKrueger()

```
coordinates.GaussKrueger.GaussKrueger ( )
```

main constructor to set the coordinates and reference meridian

## 8.9.3 Dokumentation der Elementfunktionen

### 8.9.3.1 getAsGeographicInterface()

```
GeographicCoordinateInterface coordinates.GaussKrueger.getAsGeographicInterface (
    EllipsoidParams ell )
```

transforms the coordinates of the current [GaussKrueger](#) instance into geographic coordinates

## Parameter

<i>ell</i>	- an object of type <a href="#">EllipsoidParams</a> that stores information about the reference ellipsoid used for the geographic coordinates
------------	---

## Rückgabe

geographic - returns an object of type [GeographicCoordinate](#) that represents the current [GaussKrueger](#) instance transformed to geographic coordinates

Erneute Implementation von [coordinates.Gauss](#).

### 8.9.3.2 getAsTargetCoordinate()

```
void coordinates.GaussKrueger.getAsTargetCoordinate (
    EllipsoidParams ell,
    GeographicCoordinateInterface geocoord )
```

transforms geographic coordinates into [GaussKrueger](#) coordinates and stores them in the current [GaussKrueger](#) instance

## Parameter

<i>ell</i>	- an object of type <code>EllipsoidParams</code> that stores information about the reference ellipsoid used for the geographic coordinates
<i>geocoord</i>	- an object of type <code>GeographicCoordinate</code> that contains the geographic coordinates to be transformed in <code>GaussKrueger</code> coordinates

Erneute Implementation von `coordinates.Coordinate`.

### 8.9.3.3 `getHoch()`

```
double coordinates.GaussKrueger.getHoch ( )
```

getter method for the northing of the current `GaussKrueger` instance

## Rückgabe

hoch - returns the northing of the current `GaussKrueger` instance as double

### 8.9.3.4 `getIlao()`

```
int coordinates.GaussKrueger.getIlao ( )
```

getter method for the reference meridian of the current `GaussKrueger` instance

## Rückgabe

centralMeridian - returns the reference meridian of the current `GaussKrueger` instance as integer

### 8.9.3.5 `getRechts()`

```
double coordinates.GaussKrueger.getRechts ( )
```

getter method that returns the the northing of the particular `GaussKrueger` instance

## Rückgabe

rechts - returns the northing of the particular `GaussKrueger` instance as double



#### 8.9.3.6 print()

```
void coordinates.GaussKrueger.print ( )
```

prints a string representation of the Gauss-Krueger object into the console

Erneute Implementation von [coordinates.Gauss](#).

#### 8.9.3.7 setHoch()

```
void coordinates.GaussKrueger.setHoch (
    double hoch )
```

setter method that accepts the northing of the particular [GaussKrueger](#) instance

## Parameter

<i>hoch</i>	- sets a double representing the northing for the particular <a href="#">GaussKrueger</a> instance
-------------	--

**8.9.3.8 setIlao()**

```
void coordinates.GaussKrueger.setIlao (  
    int ilao )
```

setter method for the number of the reference meridian of the current [GaussKrueger](#) instance

## Parameter

<i>ilao</i>	- sets an integer as the number of the reference meridian of the current <a href="#">GaussKrueger</a> instance
-------------	--

**8.9.3.9 setRechts()**

```
void coordinates.GaussKrueger.setRechts (  
    double rechts )
```

setter method that accepts the easting of the particular [GaussKrueger](#) instance

## Parameter

<i>rechts</i>	- sets a double representing the easting for the particular <a href="#">GaussKrueger</a> instance
---------------	---

**8.9.4 Dokumentation der Datenelemente****8.9.4.1 centralMeridian**

```
int coordinates.GaussKrueger.centralMeridian [private]
```

central meridian

**8.9.4.2 hoch**

```
double coordinates.GaussKrueger.hoch [private]
```

'hoch' is the official term used by the German survey administration ('northing')

### 8.9.4.3 rechts

```
double coordinates.GaussKrueger.rechts [private]
```

'rechts' is the official term used by the German survey administration ('easting')

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/GaussKrueger.java](#)

## 8.10 test.coordinates.GaussKruegerTest Klassenreferenz

Test class with unit tests to test behavior class GaussKrueger.

### Öffentliche Methoden

- void [getAsGeographic](#) ()  
*tests method getAsGeographic from class GaussKrueger, overwritten from abstract class coordinate expected values and input values are taken from book "Rechenformeln und Rechenbeispiele zur Landesvermessung - Teil 2" from Prof.Dr.-Ing. Albert Schoedlbauer, Robert Wichmann Verlag Karlsruhe, page 88*
- void [getAsGaussKrueger](#) ()

### Private Attribute

- final double [RHO](#) = 180/Math.PI
- final double [EPSILON\\_IN\\_METER](#) = 0.001
- final double [EPSILON\\_IN\\_DEGREE](#) = 0.00000001

### 8.10.1 Ausführliche Beschreibung

Test class with unit tests to test behavior class GaussKrueger.

#### Bemerkungen

last refactored 11.12.2017 by Eva Majer

- adaption for ControlParms Singleton

### 8.10.2 Dokumentation der Elementfunktionen

### 8.10.2.1 getAsGaussKrueger()

```
void test.coordinates.GaussKruegerTest.getAsGaussKrueger ( )
```

### 8.10.2.2 getAsGeographic()

```
void test.coordinates.GaussKruegerTest.getAsGeographic ( )
```

tests method getAsGeographic from class GaussKrueger, overwritten from abstract class coordinate expected values and input values are taken from book "Rechenformeln und Rechenbeispiele zur Landesvermessung - Teil 2" from Prof.Dr.-Ing. Albert Schoedlbauer, Robert Wichmann Verlag Karlsruhe, page 88

## 8.10.3 Dokumentation der Datenelemente

### 8.10.3.1 EPSILON\_IN\_DEGREE

```
final double test.coordinates.GaussKruegerTest.EPSILON_IN_DEGREE = 0.00000001 [private]
```

### 8.10.3.2 EPSILON\_IN\_METER

```
final double test.coordinates.GaussKruegerTest.EPSILON_IN_METER = 0.001 [private]
```

### 8.10.3.3 RHO

```
final double test.coordinates.GaussKruegerTest.RHO = 180/Math.PI [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/coordinates/[GaussKruegerTest.java](#)

## 8.11 test.coordinates.GaussTest Klassenreferenz

Test class with unit tests to test behavior class Gauss.

## Öffentliche Methoden

- void [testMeridianLength](#) ()

## Private Attribute

- final double [RHO](#) = 180/Math.PI
- final double [EPSILON\\_IN\\_METER](#) = 0.0004

### 8.11.1 Ausführliche Beschreibung

Test class with unit tests to test behavior class Gauss.

#### Bemerkungen

implemented 7.02.2019 by Norbert Rösch

this test should be run before [UTMTest](#) and [GaussKruegerTest](#)

### 8.11.2 Dokumentation der Elementfunktionen

#### 8.11.2.1 testMeridianLength()

```
void test.coordinates.GaussTest.testMeridianLength ( )
```

### 8.11.3 Dokumentation der Datenelemente

#### 8.11.3.1 EPSILON\_IN\_METER

```
final double test.coordinates.GaussTest.EPSILON_IN_METER = 0.0004 [private]
```

#### 8.11.3.2 RHO

```
final double test.coordinates.GaussTest.RHO = 180/Math.PI [private]
```

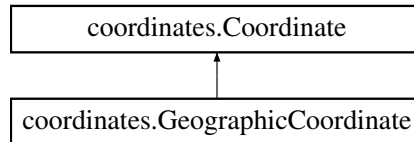
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/coordinates/[GaussTest.java](#)

## 8.12 coordinates.GeographicCoordinate Klassenreferenz

concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (as input coordinate)

Klassendiagramm für coordinates.GeographicCoordinate:



### Öffentliche Methoden

- [GeographicCoordinate](#) ()
- [GeographicCoordinate](#) (double [longitude](#), double [latitude](#))
- [GeographicCoordinate](#) (double [longitude](#), double [latitude](#), double [height](#))
- void [print](#) ()
  - prints a string representation of the particular [GeographicCoordinate](#) instance into the console*
- [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ellipsoidParms)
  - allows the particular [GeographicCoordinate](#) instance to be converted to [GeographicCoordinate](#). Assigns all parameters of [GeographicCoordinateInterface](#) the values of the parameters of this object.*
- void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ellipsoidParms, [GeographicCoordinateInterface](#) geoCoordInt)
  - allows the the particular [GeographicCoordinate](#) instance to be converted from [GeographicCoordinate](#). Assigns all values of the parameters of this object the parameters of this object.*
- long [getPoint\\_number](#) ()
  - getter method that returns point number of the particular Geographic [Coordinate](#) point*
- void [setPoint\\_number](#) (long [point\\_number](#))
  - setter method that accepts point number for the particular Geographic [Coordinate](#) point*
- double [getLongitude](#) ()
  - getter method that returns the geographical longitude*
- void [setLongitude](#) (double [longitude](#))
  - setter method that accepts the geographical longitude*
- double [getLatitude](#) ()
  - getter method that returns the geographical latitude*
- void [setLatitude](#) (double [latitude](#))
  - setter method that accepts the geographical latitude*
- double [getHeight](#) ()
  - getter method that returns the height*
- void [setHeight](#) (double [height](#))
  - setter method accepts the height*
- int [getEllipsoidal](#) ()
  - getter method that returns the height above ellipsoid*
- void [setEllipsoidal](#) (int [ellipsoidal](#))
  - setter method accepts the height above ellipsoid*

## Öffentliche, statische Methoden

- static [GeographicCoordinate getInstance](#) (double [longitude](#), double [latitude](#))  
*factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude and latitude*
- static [GeographicCoordinate getInstance](#) (double [longitude](#), double [latitude](#), double [height](#))  
*factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude, latitude and height*

## Private Methoden

- void [resetState](#) ()  
*sets all parameters of the instance to default value 0, needed because only one instance of the class can exist (singleton class)*

## Private Attribute

- long [point\\_number](#)
- double [longitude](#)
- double [latitude](#)
- double [height](#)
- int [ellipsoidal](#)

## Statische, private Attribute

- static [GeographicCoordinate myInstance](#) = null

### 8.12.1 Ausführliche Beschreibung

concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (as input coordinate)

#### Autor

unknown as of 2016-11

#### Bemerkungen

Split former class [GeographicCoordinate](#) into [GeographicCoordinate](#) (only input coordinate) and [GeographicCoordinateInterface](#) (Interface for strategy) on 2018-01-12 by Johanna Stoetzer

updated header on 2017-11-29 by Markus Mueller

renamed to [GeographicCoordinate](#) (was GeographicCoordinates) on 2016-11-26 by Patrick Huebner

#### Version

0.1

## Parameter

<i>point</i>	number - Point identification number of the <a href="#">GeographicCoordinate</a> as long
<i>longitude</i>	- Longitude of the <a href="#">GeographicCoordinate</a> as double
<i>latitude</i>	- Latitude of the <a href="#">GeographicCoordinate</a> as double
<i>height</i>	- Height of the <a href="#">GeographicCoordinate</a> as double
<i>ellipsoidal</i>	- Height above ellipsoid of the <a href="#">GeographicCoordinate</a> as integer

## 8.12.2 Beschreibung der Konstruktoren und Destruktoren

### 8.12.2.1 [GeographicCoordinate\(\)](#) [1/3]

```
coordinates.GeographicCoordinate.GeographicCoordinate ( )
```

### 8.12.2.2 [GeographicCoordinate\(\)](#) [2/3]

```
coordinates.GeographicCoordinate.GeographicCoordinate (
    double longitude,
    double latitude )
```

### 8.12.2.3 [GeographicCoordinate\(\)](#) [3/3]

```
coordinates.GeographicCoordinate.GeographicCoordinate (
    double longitude,
    double latitude,
    double height )
```

## 8.12.3 Dokumentation der Elementfunktionen

### 8.12.3.1 [getAsGeographicInterface\(\)](#)

```
GeographicCoordinateInterface coordinates.GeographicCoordinate.getAsGeographicInterface (
    EllipsoidParms ellipsoidParms )
```

allows the particular [GeographicCoordinate](#) instance to be converted to [GeographicCoordinate](#). Assigns all parameters of [GeographicCoordinateInterface](#) the values of the parameters of this object.



## Parameter

<i>ellipsoidParms</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> should refer to
-----------------------	---

Erneute Implementation von [coordinates.Coordinate](#).

### 8.12.3.2 getAsTargetCoordinate()

```
void coordinates.GeographicCoordinate.getAsTargetCoordinate (
    EllipsoidParms ellipsoidParms,
    GeographicCoordinateInterface geoCoordInt )
```

allows the the particular [GeographicCoordinate](#) instance to be converted from [GeographicCoordinate](#). Assigns all values of the parameters of this object the parameters of this object.

## Parameter

<i>geoCoordInt</i>	- an object of type <a href="#">GeographicCoordinate</a> that represents the geographic coordinate the concrete <a href="#">Coordinate</a> instance should be converted from
<i>ellipsoidParms</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> refers to

Erneute Implementation von [coordinates.Coordinate](#).

### 8.12.3.3 getEllipsoidal()

```
int coordinates.GeographicCoordinate.getEllipsoidal ( )
```

getter method that returns the height above ellipsoid

## Rückgabe

ellipsoidal - returns a double value representing the height above ellipsoid

### 8.12.3.4 getHeight()

```
double coordinates.GeographicCoordinate.getHeight ( )
```

getter method that returns the height

## Rückgabe

height - returns a double value representing the height

#### 8.12.3.5 getInstance() [1/2]

```
static GeographicCoordinate coordinates.GeographicCoordinate.getInstance (
    double longitude,
    double latitude ) [static]
```

factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude and latitude

##### Parameter

<i>longitude</i>	- double value of the geographical longitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>latitude</i>	- double value of the geographical latitude the <a href="#">GeographicCoordinate</a> instance should be initialized with

#### 8.12.3.6 getInstance() [2/2]

```
static GeographicCoordinate coordinates.GeographicCoordinate.getInstance (
    double longitude,
    double latitude,
    double height ) [static]
```

factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude, latitude and height

##### Parameter

<i>longitude</i>	- double value of the geographical longitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>latitude</i>	- double value of the geographical latitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>height</i>	- double value of the height the <a href="#">GeographicCoordinate</a> instance should be initialized with

#### 8.12.3.7 getLatitude()

```
double coordinates.GeographicCoordinate.getLatitude ( )
```

getter method that returns the geographical latitude

##### Rückgabe

latitude - returns a double value representing the geographical latitude

### 8.12.3.8 getLongitude()

```
double coordinates.GeographicCoordinate.getLongitude ( )
```

getter method that returns the geographical longitude

#### Rückgabe

longitude - returns a double value representing the geographical longitude

### 8.12.3.9 getPoint\_number()

```
long coordinates.GeographicCoordinate.getPoint_number ( )
```

getter method that returns point number of the particular Geographic [Coordinate](#) point

#### Rückgabe

point\_number - returns the point number of the particular Geographic [Coordinate](#) point as long value

### 8.12.3.10 print()

```
void coordinates.GeographicCoordinate.print ( )
```

prints a string representation of the particular [GeographicCoordinate](#) instance into the console

Erneute Implementation von [coordinates.Coordinate](#).

### 8.12.3.11 resetState()

```
void coordinates.GeographicCoordinate.resetState ( ) [private]
```

sets all parameters of the instance to default value 0, needed because only one instance of the class can exist (singleton class)

### 8.12.3.12 setEllipsoidal()

```
void coordinates.GeographicCoordinate.setEllipsoidal (
    int ellipsoidal )
```

setter method accepts the height above ellipsoid

## Parameter

<i>ellipsoidal</i>	- double value representing the height above ellipsoid
--------------------	--

**8.12.3.13 setHeight()**

```
void coordinates.GeographicCoordinate.setHeight (
    double height )
```

setter method accepts the height

## Parameter

<i>height</i>	- double value representing the height
---------------	--

**8.12.3.14 setLatitude()**

```
void coordinates.GeographicCoordinate.setLatitude (
    double latitude )
```

setter method that accepts the geographical latitude

## Parameter

<i>latitude</i>	- double value representing the geographical latitude
-----------------	---

**8.12.3.15 setLongitude()**

```
void coordinates.GeographicCoordinate.setLongitude (
    double longitude )
```

setter method that accepts the geographical longitude

## Parameter

<i>longitude</i>	- double value representing the geographical longitude
------------------	--

### 8.12.3.16 setPoint\_number()

```
void coordinates.GeographicCoordinate.setPoint_number (
    long point_number )
```

setter method that accepts point number for the particular Geographic [Coordinate](#) point

#### Parameter

<i>point_number</i>	- a long value representing the point number for the particular Geographic <a href="#">Coordinate</a> point
---------------------	---

## 8.12.4 Dokumentation der Datenelemente

### 8.12.4.1 ellipsoidal

```
int coordinates.GeographicCoordinate.ellipsoidal [private]
```

Height above ellipsoid: 0 => no height (not defined), 1 => ellipsoidal, 2 => geoid

### 8.12.4.2 height

```
double coordinates.GeographicCoordinate.height [private]
```

### 8.12.4.3 latitude

```
double coordinates.GeographicCoordinate.latitude [private]
```

### 8.12.4.4 longitude

```
double coordinates.GeographicCoordinate.longitude [private]
```

### 8.12.4.5 myInstance

```
GeographicCoordinate coordinates.GeographicCoordinate.myInstance = null [static], [private]
```

#### 8.12.4.6 point\_number

```
long coordinates.GeographicCoordinate.point_number [private]
```

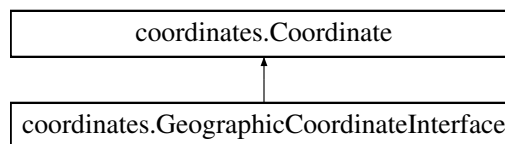
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/GeographicCoordinate.java](#)

### 8.13 coordinates.GeographicCoordinateInterface Klassenreferenz

concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (As interface for strategy)

Klassendiagramm für `coordinates.GeographicCoordinateInterface`:



#### Öffentliche Methoden

- [GeographicCoordinateInterface](#) ()
- [GeographicCoordinateInterface](#) (double [longitude](#), double [latitude](#))
- void [print](#) ()  
*prints a string representation of the particular [GeographicCoordinate](#) instance into the console*
- [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ellipsoidParms)  
*allows the particular [GeographicCoordinate](#) instance to be converted to [GeographicCoordinate](#). simply returns this reference*
- void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ellipsoidParms, [GeographicCoordinateInterface](#) geographic↔  
Coordinate)  
*allows the the particular [GeographicCoordinate](#) instance to be converted from [GeographicCoordinate](#). simply uses copy method*
- [GeographicCoordinateInterface](#) [molodenskytrafo](#) ([EllipsoidParms](#) ell, [MolodenskyParm](#) trafoparameter)  
*calculates a Molodensky transformation*
- double [fupubre](#) (double s, [EllipsoidParms](#) ell)  
*calculates pulse width (Fusspunktbreite) for a given range*
- void [GeographicLongitudeLatitude](#) (double xg, double yg, [EllipsoidParms](#) ell, double scale, double l0)  
*sets longitude and latitude according to given parameters*
- void [copy](#) ([GeographicCoordinateInterface](#) geo)  
*works like a copy constructor, initializes particular instance with the values of the instance passed as parameter*
- long [getPoint\\_number](#) ()  
*getter method that returns point number of the particular Geographic [Coordinate](#) point*
- void [setPoint\\_number](#) (long [point\\_number](#))  
*setter method that accepts point number for the particular Geographic [Coordinate](#) point*
- double [getLongitude](#) ()  
*getter method that returns the geographical longitude*
- void [setLongitude](#) (double [longitude](#))

- setter method that accepts the geographical longitude*
- double `getLatitude ()`  
*getter method that returns the geographical latitude*
- void `setLatitude (double latitude)`  
*setter method that accepts the geographical latitude*
- double `getHeight ()`  
*getter method that returns the height*
- void `setHeight (double height)`  
*setter method accepts the height*
- int `getEllipsoidal ()`  
*getter method that returns the height above ellipsoid*
- void `setEllipsoidal (int ellipsoidal)`  
*setter method accepts the height above ellipsoid*

## Öffentliche, statische Methoden

- static `GeographicCoordinateInterface getInstance ()`  
*factory method that returns the singular instance of the singleton class initialized with default constructor*
- static `GeographicCoordinateInterface getInstance (double longitude, double latitude)`  
*factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude and latitude*
- static `GeographicCoordinateInterface getInstance (double longitude, double latitude, double height)`  
*factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude, latitude and height*

## Private Methoden

- `GeographicCoordinateInterface (double longitude, double latitude, double height)`
- void `resetState ()`  
*sets all parameters of the instance to default value 0, needed because only one instance of the class can exist (singleton class)*

## Private Attribute

- long `point_number`
- double `longitude`
- double `latitude`
- double `height`
- int `ellipsoidal`

## Statische, private Attribute

- static `GeographicCoordinateInterface myInstance = null`

### 8.13.1 Ausführliche Beschreibung

concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (As interface for strategy)

#### Autor

unknown as of 2016-11

#### Bemerkungen

renamed to [GeographicCoordinateInterface](#). Class is now separated into [GeographicCoordinate](#) (only input coordinate) and [GeographicCoordinateInterface](#) (Interface for strategy) on 2018-01-12 by Johanna Stoetzer

updated header on 2017-11-29 by Markus Mueller

renamed to [GeographicCoordinate](#) (was GeographicCoordinates) on 2016-11-26 by Patrick Huebner

#### Version

0.1

#### Parameter

<i>point</i>	number - Point identification number of the <a href="#">GeographicCoordinate</a> as long
<i>longitude</i>	- Longitude of the <a href="#">GeographicCoordinate</a> as double
<i>latitude</i>	- Latitude of the <a href="#">GeographicCoordinate</a> as double
<i>height</i>	- Height of the <a href="#">GeographicCoordinate</a> as double
<i>ellipsoidal</i>	- Height above ellipsoid of the <a href="#">GeographicCoordinate</a> as integer

### 8.13.2 Beschreibung der Konstruktoren und Destruktoren

#### 8.13.2.1 [GeographicCoordinateInterface\(\)](#) [1/3]

```
coordinates.GeographicCoordinateInterface.GeographicCoordinateInterface ( )
```

#### Bemerkungen

public/visible due to testing

#### Parameter

<i>longitude</i>	
<i>latitude</i>	



### 8.13.2.2 GeographicCoordinateInterface() [2/3]

```
coordinates.GeographicCoordinateInterface.GeographicCoordinateInterface (
    double longitude,
    double latitude )
```

### 8.13.2.3 GeographicCoordinateInterface() [3/3]

```
coordinates.GeographicCoordinateInterface.GeographicCoordinateInterface (
    double longitude,
    double latitude,
    double height ) [private]
```

## 8.13.3 Dokumentation der Elementfunktionen

### 8.13.3.1 copy()

```
void coordinates.GeographicCoordinateInterface.copy (
    GeographicCoordinateInterface geo )
```

works like a copy constructor, initializes particular instance with the values of the instance passed as parameter

#### Parameter

<i>geo</i>	- the <a href="#">GeographicCoordinate</a> to initialize the particular instance with
------------	---

### 8.13.3.2 fupubre()

```
double coordinates.GeographicCoordinateInterface.fupubre (
    double s,
    EllipsoidParms ell )
```

calculates pulse width (Fusspunktbreite) for a given range

#### Parameter

<i>s</i>	- double value for the range
<i>ell</i>	- an object of type <a href="#">EllipsoidParms</a> that contains details about the ellipsoid the pulse width calculation refers to

**Rückgabe**

br - returns the pulse width (Fusspunktbreite) as double value

**8.13.3.3 GeographicLongitudeLatitude()**

```
void coordinates.GeographicCoordinateInterface.GeographicLongitudeLatitude (
    double xg,
    double yg,
    EllipsoidParms ell,
    double scale,
    double l0 )
```

sets longitude and latitude according to given parameters

**Parameter**

<i>xg</i>	- double value representing geographic decimal coordinate?
<i>yg</i>	- double value representing geographic decimal coordinate?
<i>ell</i>	- an object of type EllipsoidParms that contains details about the ellipsoid
<i>scale</i>	- scale as double value
<i>l0</i>	- ??

**8.13.3.4 getAsGeographicInterface()**

```
GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.getAsGeographic↵
Interface (
    EllipsoidParms ellipsoidParms )
```

allows the particular [GeographicCoordinate](#) instance to be converted to [GeographicCoordinate](#). simply returns this reference

**Parameter**

<i>ellipsoidParms</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> should refer to
-----------------------	---

Erneute Implementation von [coordinates.Coordinate](#).

**8.13.3.5 getAsTargetCoordinate()**

```
void coordinates.GeographicCoordinateInterface.getAsTargetCoordinate (
    EllipsoidParms ellipsoidParms,
    GeographicCoordinateInterface geographicCoordinate )
```

allows the the particular [GeographicCoordinate](#) instance to be converted from [GeographicCoordinate](#). simply uses copy method

**Parameter**

<i>geographicCoordinate</i>	- an object of type <a href="#">GeographicCoordinate</a> that represents the geographic coordinate the concrete <a href="#">Coordinate</a> instance should be converted from
<i>ellipsoidParms</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the <a href="#">GeographicCoordinate</a> refers to

Erneute Implementation von [coordinates.Coordinate](#).

**8.13.3.6 getEllipsoidal()**

```
int coordinates.GeographicCoordinateInterface.getEllipsoidal ( )
```

getter method that returns the height above ellipsoid

**Rückgabe**

ellipsoidal - returns a double value representing the height above ellipsoid

**8.13.3.7 getHeight()**

```
double coordinates.GeographicCoordinateInterface.getHeight ( )
```

getter method that returns the height

**Rückgabe**

height - returns a double value representing the height

**8.13.3.8 getInstance() [1/3]**

```
static GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.getInstance ( )  
[static]
```

factory method that returns the singular instance of the singleton class initialized with default constructor

**8.13.3.9 getInstance() [2/3]**

```
static GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.getInstance (   
    double longitude,   
    double latitude ) [static]
```

factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude and latitude

## Parameter

<i>longitude</i>	- double value of the geographical longitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>latitude</i>	- double value of the geographical latitude the <a href="#">GeographicCoordinate</a> instance should be initialized with

**8.13.3.10 getInstance() [3/3]**

```
static GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.getInstance (
    double longitude,
    double latitude,
    double height ) [static]
```

factory method that returns the singular instance of the singleton class initialized with constructor that accepts values for longitude, latitude and height

## Parameter

<i>longitude</i>	- double value of the geographical longitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>latitude</i>	- double value of the geographical latitude the <a href="#">GeographicCoordinate</a> instance should be initialized with
<i>height</i>	- double value of the height the <a href="#">GeographicCoordinate</a> instance should be initialized with

**8.13.3.11 getLatitude()**

```
double coordinates.GeographicCoordinateInterface.getLatitude ( )
```

getter method that returns the geographical latitude

## Rückgabe

latitude - returns a double value representing the geographical latitude

**8.13.3.12 getLongitude()**

```
double coordinates.GeographicCoordinateInterface.getLongitude ( )
```

getter method that returns the geographical longitude

## Rückgabe

longitude - returns a double value representing the geographical longitude

#### 8.13.3.13 getPoint\_number()

```
long coordinates.GeographicCoordinateInterface.getPoint_number ( )
```

getter method that returns point number of the particular Geographic [Coordinate](#) point

##### Rückgabe

point\_number - returns the point number of the particular Geographic [Coordinate](#) point as long value

#### 8.13.3.14 molodenskytrafo()

```
GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.molodenskytrafo (
    EllipsoidParms ell,
    MolodenskyParm trafoparameter )
```

calculates a Molodensky transformation

##### Parameter

<i>ell</i>	- an object of type EllipsoidParms that contains details about the ellipsoid the transformation refers to
<i>trafoparameter</i>	- an object of type MolodenskyParm that contains the parameters of the Molodenskij transformation

##### Rückgabe

geo\_p2 - returns a transformed geographic coordinate

#### 8.13.3.15 print()

```
void coordinates.GeographicCoordinateInterface.print ( )
```

prints a string representation of the particular [GeographicCoordinate](#) instance into the console

Erneute Implementation von [coordinates.Coordinate](#).

#### 8.13.3.16 resetState()

```
void coordinates.GeographicCoordinateInterface.resetState ( ) [private]
```

sets all parameters of the instance to default value 0, needed because only one instance of the class can exist (singleton class)

#### 8.13.3.17 setEllipsoidal()

```
void coordinates.GeographicCoordinateInterface.setEllipsoidal (
    int ellipsoidal )
```

setter method accepts the height above ellipsoid

##### Parameter

<i>ellipsoidal</i>	- double value representing the height above ellipsoid
--------------------	--

#### 8.13.3.18 setHeight()

```
void coordinates.GeographicCoordinateInterface.setHeight (
    double height )
```

setter method accepts the height

##### Parameter

<i>height</i>	- double value representing the height
---------------	--

#### 8.13.3.19 setLatitude()

```
void coordinates.GeographicCoordinateInterface.setLatitude (
    double latitude )
```

setter method that accepts the geographical latitude

##### Parameter

<i>latitude</i>	- double value representing the geographical latitude
-----------------	---

#### 8.13.3.20 setLongitude()

```
void coordinates.GeographicCoordinateInterface.setLongitude (
    double longitude )
```

setter method that accepts the geographical longitude

## Parameter

<i>longitude</i>	- double value representing the geographical longitude
------------------	--

**8.13.3.21 setPoint\_number()**

```
void coordinates.GeographicCoordinateInterface.setPoint_number (
    long point_number )
```

setter method that accepts point number for the particular Geographic [Coordinate](#) point

## Parameter

<i>point_number</i>	- a long value representing the point number for the particular Geographic <a href="#">Coordinate</a> point
---------------------	---

**8.13.4 Dokumentation der Datenelemente****8.13.4.1 ellipsoidal**

```
int coordinates.GeographicCoordinateInterface.ellipsoidal [private]
```

Height above ellipsoid: 0 => no height (not defined), 1 => ellipsoidal, 2 => geoid

**8.13.4.2 height**

```
double coordinates.GeographicCoordinateInterface.height [private]
```

**8.13.4.3 latitude**

```
double coordinates.GeographicCoordinateInterface.latitude [private]
```

**8.13.4.4 longitude**

```
double coordinates.GeographicCoordinateInterface.longitude [private]
```

#### 8.13.4.5 myInstance

```
GeographicCoordinateInterface coordinates.GeographicCoordinateInterface.myInstance = null  
[static], [private]
```

#### 8.13.4.6 point\_number

```
long coordinates.GeographicCoordinateInterface.point_number [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/GeographicCoordinateInterface.java](#)

### 8.14 test.coordinates.GeographicCoordinateTest Klassenreferenz

#### Öffentliche Methoden

- void [testGetAsGeographicInterface](#) ()
- void [testFromGeographicInterface](#) ()

#### Statische, private Attribute

- static final double [EPSILONFORDEGREE](#) = 5.e-10

#### 8.14.1 Dokumentation der Elementfunktionen

##### 8.14.1.1 testFromGeographicInterface()

```
void test.coordinates.GeographicCoordinateTest.testFromGeographicInterface ( )
```

##### 8.14.1.2 testGetAsGeographicInterface()

```
void test.coordinates.GeographicCoordinateTest.testGetAsGeographicInterface ( )
```

#### 8.14.2 Dokumentation der Datenelemente



### 8.14.2.1 EPSILONFORDEGREE

```
final double test.coordinates.GeographicCoordinateTest.EPSILONFORDEGREE = 5.e-10 [static],
[private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/coordinates/[GeographicCoordinateTest.java](#)

## 8.15 coordinates.GetAppropriateCoordinate Klassenreferenz

[Coordinate](#) factory for plane coordinates and geographic coordinates.

### Öffentliche, statische Methoden

- static [Coordinate](#) [getCoord](#) (String object)  
*getter method that returns an appropriate coordinate depending on the users input*
- static [GeographicCoordinateInterface](#) [getGeographicCoordinateInterface](#) ()

### 8.15.1 Ausführliche Beschreibung

[Coordinate](#) factory for plane coordinates and geographic coordinates.

#### Autor

Norbert Rösch

#### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

last refactored 11.12.2017 by Eva Majer

- adaption for ControlParms Singleton
- Translation of some comments

2018-01-06 add comments by Yunhao Huang

2018-01-12 added new [GeographicCoordinate](#) as comment by Johanna Stoetzer

2018-01-25 added XYZ-Coordinate to getCoord by Markus Mueller

#### Version

0.1

#### Parameter

<a href="#">Coordinate</a>	- the plane coordinates <a href="#">GaussKrueger</a> , <a href="#">UTM</a> and <a href="#">Soldner</a>
<a href="#">GeographicCoordinateInterface</a>	- the geographic coordinate interface

## 8.15.2 Dokumentation der Elementfunktionen

### 8.15.2.1 getCoord()

```
static Coordinate coordinates.GetAppropriateCoordinate.getCoord (
    String object ) [static]
```

getter method that returns an appropriate coordinate depending on the users input

Parameter

<i>object</i>	- string representing the type of the coordinate
---------------	--

### 8.15.2.2 getGeographicCoordinateInterface()

```
static GeographicCoordinateInterface coordinates.GetAppropriateCoordinate.getGeographic↵
CoordinateInterface ( ) [static]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/GetAppropriateCoordinate.java](#)

## 8.16 datumstrategy.GetAppropriateTransformationAlgorithm Klassenreferenz

### Öffentliche, statische Methoden

- static [TransformationStrategy](#) getStrategy ()

### 8.16.1 Ausführliche Beschreibung

Bemerkungen

- adaption for ControlParms Singleton

### 8.16.2 Dokumentation der Elementfunktionen

### 8.16.2.1 getStrategy()

```
static TransformationStrategy datumstrategy.GetAppropriateTransformationAlgorithm.getStrategy
( ) [static]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[GetAppropriateTransformationAlgorithm.java](#)

## 8.17 test.datumstrategy.GetAppropriateTransformationAlgorithmTest Klassenreferenz

### Öffentliche Methoden

- void [testSpatialSimilarityTransformationInfin](#) ()

### 8.17.1 Dokumentation der Elementfunktionen

#### 8.17.1.1 testSpatialSimilarityTransformationInfin()

```
void test.datumstrategy.GetAppropriateTransformationAlgorithmTest.testSpatialSimilarity↵
TransformationInfin ( )
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/datumstrategy/[GetAppropriateTransformationAlgorithmTest.java](#)

## 8.18 params.LatParm Klassenreferenz

Latitude Parameters.

### Öffentliche Methoden

- [LatParm](#) ()
- void [Constant](#) ([EllipsoidParms](#) ell, double latitude)
- double [getRadm](#) ()
- void [setRadm](#) (double [radm](#))
- double [getRadn](#) ()
- void [setRadn](#) (double [radn](#))
- double [getRadg](#) ()
- void [setRadg](#) (double [radg](#))
- double [getTbr](#) ()
- void [setTbr](#) (double [tbr](#))
- double [getVbr](#) ()
- void [setVbr](#) (double [vbr](#))
- double [getEtabr](#) ()
- void [setEtabr](#) (double [etabr](#))
- double [getCbr](#) ()
- void [setCbr](#) (double [cbr](#))
- double [getVbr2](#) ()
- void [setVbr2](#) (double [vbr2](#))
- double [getTbr2](#) ()
- void [setTbr2](#) (double [tbr2](#))

## Private Attribute

- double `radm`
- double `radn`
- double `radg`
- double `tbr`
- double `vbr`
- double `etabr`
- double `cbr`

### 8.18.1 Ausführliche Beschreibung

Latitude Parameters.

#### Autor

unknown

#### Bemerkungen

add header on 2018-01-06 by Yunhao Huang

### 8.18.2 Beschreibung der Konstruktoren und Destruktoren

#### 8.18.2.1 LatParm()

```
params.LatParm.LatParm ( )
```

### 8.18.3 Dokumentation der Elementfunktionen

#### 8.18.3.1 Constant()

```
void params.LatParm.Constant (
    EllipsoidParms ell,
    double latitude )
```

#### 8.18.3.2 getCbr()

```
double params.LatParm.getCbr ( )
```

**8.18.3.3 getEtabr()**

```
double params.LatParm.getEtabr ( )
```

**8.18.3.4 getRadg()**

```
double params.LatParm.getRadg ( )
```

**8.18.3.5 getRadm()**

```
double params.LatParm.getRadm ( )
```

**8.18.3.6 getRadn()**

```
double params.LatParm.getRadn ( )
```

**8.18.3.7 getTbr()**

```
double params.LatParm.getTbr ( )
```

**8.18.3.8 getTbr2()**

```
double params.LatParm.getTbr2 ( )
```

**8.18.3.9 getVbr()**

```
double params.LatParm.getVbr ( )
```

**8.18.3.10 getVbr2()**

```
double params.LatParm.getVbr2 ( )
```

**8.18.3.11 setCbr()**

```
void params.LatParm.setCbr (
    double cbr )
```

**8.18.3.12 setEtabr()**

```
void params.LatParm.setEtabr (
    double etabr )
```

**8.18.3.13 setRadg()**

```
void params.LatParm.setRadg (
    double radg )
```

**8.18.3.14 setRadm()**

```
void params.LatParm.setRadm (
    double radm )
```

**8.18.3.15 setRadn()**

```
void params.LatParm.setRadn (
    double radn )
```

**8.18.3.16 setTbr()**

```
void params.LatParm.setTbr (
    double tbr )
```

**8.18.3.17 setTbr2()**

```
void params.LatParm.setTbr2 (
    double tbr2 )
```

**8.18.3.18 setVbr()**

```
void params.LatParm.setVbr (
    double vbr )
```

**8.18.3.19 setVbr2()**

```
void params.LatParm.setVbr2 (
    double vbr2 )
```

**8.18.4 Dokumentation der Datenelemente****8.18.4.1 cbr**

```
double params.LatParm.cbr [private]
```

**8.18.4.2 etabr**

```
double params.LatParm.etabr [private]
```

auxiliary value eta2 for the latitude BR

**8.18.4.3 radg**

```
double params.LatParm.radg [private]
```

coordinates. Gaussian curvature radius G

**8.18.4.4 radm**

```
double params.LatParm.radm [private]
```

radius of meridian curvature M

**8.18.4.5 radn**

```
double params.LatParm.radn [private]
```

radius of transverse curvature N

#### 8.18.4.6 tbr

```
double params.LatParm.tbr [private]
```

Tangent of latitude

#### 8.18.4.7 vbr

```
double params.LatParm.vbr [private]
```

auxiliary value V for the latitude BR

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [params/LatParm.java](#)

### 8.19 test.params.LatParmTest Klassenreferenz

#### Öffentliche Methoden

- void [test](#) ()

#### 8.19.1 Dokumentation der Elementfunktionen

##### 8.19.1.1 test()

```
void test.params.LatParmTest.test ( )
```

Test based on parameters of GRS 80

#### Autor

N. Roesch

#### Datum

18.01.2017

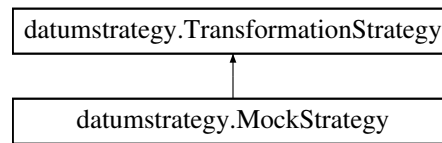
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [test/params/LatParmTest.java](#)



## 8.20 datumstrategy.MockStrategy Klassenreferenz

Klassendiagramm für datumstrategy.MockStrategy:



### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geographicCoordinate)  
*abstract method for the transformation of a geographic coordinate*

### 8.20.1 Dokumentation der Elementfunktionen

#### 8.20.1.1 transform()

```
void datumstrategy.MockStrategy.transform (
    GeographicCoordinateInterface geographicCoordinate )
```

abstract method for the transformation of a geographic coordinate

#### Parameter

<i>geographicCoordinate</i>	- an object of type <a href="#">GeographicCoordinateInterface</a> which is used for input/output for the transformation algorithm
-----------------------------	---

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[MockStrategy.java](#)

## 8.21 test.datumstrategy.MockStrategyTest Klassenreferenz

Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used.

## Öffentliche Methoden

- void `testTransformGaussKruegerStrip3ToStrip4()`  
*Test method "testTransformGaussKruegerStrip3ToStrip4" to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip without changing the geodetic datum (DHDN).*
- void `testTransformGaussKruegerStrip4ToStrip3()`  
*Test method "testTransformGaussKruegerStrip4ToStrip3" to test the transition of a Gauss-Krueger-coordinate from the fourth to the third strip without changing the geodetic datum (DHDN).*

## Private Attribute

- final double `EPSILON_IN_METER` = 0.001

### 8.21.1 Ausführliche Beschreibung

Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used.

#### Autor

Jessica Palka

#### Datum

December 2019

### 8.21.2 Dokumentation der Elementfunktionen

#### 8.21.2.1 testTransformGaussKruegerStrip3ToStrip4()

```
void test.datumstrategy.MockStrategyTest.testTransformGaussKruegerStrip3ToStrip4 ( )
```

Test method "testTransformGaussKruegerStrip3ToStrip4" to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip without changing the geodetic datum (DHDN).

#### Bemerkungen

Expected values and input values for this test are taken from Prof.Dr.-Ing. Albert Schoedlbauer, "↔  
Rechenformeln und Rechenbeispiele zur Landesvermessung - Teil 2", Robert Wichmann Verlag Karlsruhe,  
page 153

### 8.21.2.2 testTransformGaussKruegerStrip4ToStrip3()

```
void test.datumstrategy.MockStrategyTest.testTransformGaussKruegerStrip4ToStrip3 ( )
```

Test method "testTransformGaussKruegerStrip4ToStrip3" to test the transition of a Gauss-Krueger-coordinate from the fourth to the third strip without changing the geodetic datum (DHDN).

#### Bemerkungen

Expected values and input values for this test are taken from Prof.Dr.-Ing. Albert Schoedlbauer, "↵  
Rechenformeln und Rechenbeispiele zur Landesvermessung - Teil 2", Robert Wichmann Verlag Karlsruhe,  
page 154

## 8.21.3 Dokumentation der Datenelemente

### 8.21.3.1 EPSILON\_IN\_METER

```
final double test.datumstrategy.MockStrategyTest.EPSILON_IN_METER = 0.001 [private]
```

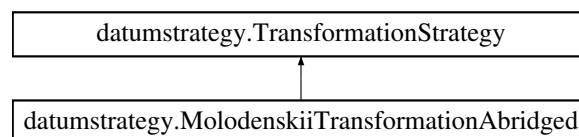
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/datumstrategy/[MockStrategyTest.java](#)

## 8.22 datumstrategy.MolodenskiiTransformationAbridged Klassenreferenz

Algorithmic realization of the abridged molodensky transformation.

Klassendiagramm für datumstrategy.MolodenskiiTransformationAbridged:



### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*abstract method for the transformation of a geographic coordinate*

### Öffentliche, statische Methoden

- static [MolodenskiiTransformationAbridged getInstance](#) ()

## Private Methoden

- `MolodenskiiTransformationAbridged ()`

## Statische, private Attribute

- static `MolodenskiiTransformationAbridged myInstance` = null

### 8.22.1 Ausführliche Beschreibung

Algorithmic realization of the abridged molodensky transformation.

#### Autor

Svea Krikau

#### Bemerkungen

created 7. December 2019

#### Version

0.1 Quelle: <http://www.mygeodesy.id.au/documents/Molodensky%20V2.pdf>

### 8.22.2 Beschreibung der Konstruktoren und Destruktoren

#### 8.22.2.1 MolodenskiiTransformationAbridged()

```
datumstrategy.MolodenskiiTransformationAbridged.MolodenskiiTransformationAbridged ( ) [private]
```

### 8.22.3 Dokumentation der Elementfunktionen

#### 8.22.3.1 getInstance()

```
static MolodenskiiTransformationAbridged datumstrategy.MolodenskiiTransformationAbridged.get←  
Instance ( ) [static]
```

#### 8.22.3.2 transform()

```
void datumstrategy.MolodenskiiTransformationAbridged.transform (   
    GeographicCoordinateInterface geographicCoordinate )
```

abstract method for the transformation of a geographic coordinate

## Parameter

<i>geographicCoordinate</i>	- an object of type GeographicCoordinateInterface which is used for input/output for the transformation algorithm
-----------------------------	---

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

## 8.22.4 Dokumentation der Datenelemente

### 8.22.4.1 myInstance

```
MolodenskiiTransformationAbridged datumstrategy.MolodenskiiTransformationAbridged.myInstance =
null [static], [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[MolodenskiiTransformationAbridged.java](#)

## 8.23 test.datumstrategy.MolodenskiiTransformationAbridgedTest Klassenreferenz

Test of the algorithmic realization of the abridged molodensky transformation.

### Öffentliche Methoden

- void [testMolodenskiiTransformationAbridged](#) ()

### Private Attribute

- final double [RHO](#) = 180 / Math.PI
- final double [EPSILON\\_meter](#) = 0.0001
- final double [EPSILON\\_rad](#) = 0.000000001

### 8.23.1 Ausführliche Beschreibung

Test of the algorithmic realization of the abridged molodensky transformation.

#### Autor

Svea Krikau

#### Bemerkungen

created 7. December 2019

#### Version

0.1 Quelle: <http://www.mygeodesy.id.au/documents/Molodensky%20V2.pdf>

## 8.23.2 Dokumentation der Elementfunktionen

### 8.23.2.1 testMolodenskiiTransformationAbridged()

```
void test.datumstrategy.MolodenskiiTransformationAbridgedTest.testMolodenskiiTransformation↔  
Abridged ( )
```

Test of the abridged Molodenskii [Transformation](#). [Transformation](#) from the Australian Geodetic Datum 1966 (AGD66) to the World Geodetic System 1984 (WGS84) Source: <http://www.mygeodesy.id.au/documents/↔Molodensky%20V2.pdf>

## 8.23.3 Dokumentation der Datenelemente

### 8.23.3.1 EPSILON\_meter

```
final double test.datumstrategy.MolodenskiiTransformationAbridgedTest.EPSILON_meter = 0.0001  
[private]
```

### 8.23.3.2 EPSILON\_rad

```
final double test.datumstrategy.MolodenskiiTransformationAbridgedTest.EPSILON_rad = 0.000000001  
[private]
```

### 8.23.3.3 RHO

```
final double test.datumstrategy.MolodenskiiTransformationAbridgedTest.RHO = 180 / Math.PI  
[private]
```

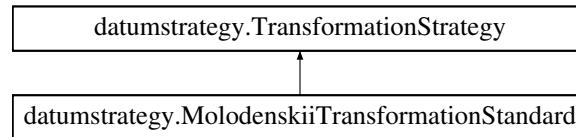
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [test/datumstrategy/MolodenskiiTransformationAbridgedTest.java](#)

## 8.24 datumstrategy.MolodenskiiTransformationStandard Klassenreferenz

Algorithmic realization of the molodensky transformation.

Klassendiagramm für datumstrategy.MolodenskiiTransformationStandard:



### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*abstract method for the transformation of a geographic coordinate*

### Öffentliche, statische Methoden

- static [MolodenskiiTransformationStandard getInstance](#) ()

### Private Methoden

- [MolodenskiiTransformationStandard](#) ()

### Statische, private Attribute

- static [MolodenskiiTransformationStandard myInstance](#) = null

### 8.24.1 Ausführliche Beschreibung

Algorithmic realization of the molodensky transformation.

#### Autor

unknown

#### Bemerkungen

last refactored 11.12.2017 by Eva Majer

- adaption for ControlParms Singleton
- Translation of some comments
- removed declaration of dl hd db which appeared 3 times

changed the class name (CamelCase) on 20.12.17 by Markus Hillemann

#### Version

0.1

## 8.24.2 Beschreibung der Konstruktoren und Destruktoren

### 8.24.2.1 MolodenskiiTransformationStandard()

```
datumstrategy.MolodenskiiTransformationStandard.MolodenskiiTransformationStandard ( ) [private]
```

## 8.24.3 Dokumentation der Elementfunktionen

### 8.24.3.1 getInstance()

```
static MolodenskiiTransformationStandard datumstrategy.MolodenskiiTransformationStandard.get←  
Instance ( ) [static]
```

### 8.24.3.2 transform()

```
void datumstrategy.MolodenskiiTransformationStandard.transform (   
    GeographicCoordinateInterface geographicCoordinate )
```

abstract method for the transformation of a geographic coordinate

#### Parameter

<i>geographicCoordinate</i>	- an object of type GeographicCoordinateInterface which is used for input/output for the transformation algorithm
-----------------------------	---

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

## 8.24.4 Dokumentation der Datenelemente

### 8.24.4.1 myInstance

```
MolodenskiiTransformationStandard datumstrategy.MolodenskiiTransformationStandard.myInstance =  
null [static], [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [datumstrategy/MolodenskiiTransformationStandard.java](#)



## 8.25 test.datumstrategy.MolodenskiiTransformationStandardTest Klassenreferenz

Tests the Molodenskii [Transformation](#) Standard as a Datum Strategy. Input values and expected values were calculated by hand.

### Öffentliche Methoden

- void [testTransform](#) ()

### Private Attribute

- final double [RHO](#) = 180/Math.PI
- final double [EPSILON\\_IN\\_METER](#) = 0.001
- final double [EPSILON\\_IN\\_RAD](#) = 0.00000001

### 8.25.1 Ausführliche Beschreibung

Tests the Molodenskii [Transformation](#) Standard as a Datum Strategy. Input values and expected values were calculated by hand.

#### Autor

Steven Landgraf

#### Datum

December 2019

#### Version

0.1

### 8.25.2 Dokumentation der Elementfunktionen

#### 8.25.2.1 testTransform()

```
void test.datumstrategy.MolodenskiiTransformationStandardTest.testTransform ( )
```

### 8.25.3 Dokumentation der Datenelemente

### 8.25.3.1 EPSILON\_IN\_METER

```
final double test.datumstrategy.MolodenskiiTransformationStandardTest.EPSILON_IN_METER = 0.001  
[private]
```

### 8.25.3.2 EPSILON\_IN\_RAD

```
final double test.datumstrategy.MolodenskiiTransformationStandardTest.EPSILON_IN_RAD = 0.↵  
00000001 [private]
```

### 8.25.3.3 RHO

```
final double test.datumstrategy.MolodenskiiTransformationStandardTest.RHO = 180/Math.PI [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/datumstrategy/[MolodenskiiTransformationStandardTest.java](#)

## 8.26 params.MolodenskyParm Klassenreferenz

Molodensky Parameters.

### Öffentliche Methoden

- [MolodenskyParm](#) ()  
*Transformation parameter from WGS84 to the geodesic datumstrategy. European Datum 1950 (ED50) for West Germany.*
- double [getTransx](#) ()
- void [setTransx](#) (double [transx](#))
- double [getTransy](#) ()
- void [setTransy](#) (double [transy](#))
- double [getTransz](#) ()
- void [setTransz](#) (double [transz](#))
- double [getDa](#) ()
- void [setDa](#) (double [da](#))
- double [getDf](#) ()
- void [setDf](#) (double [df](#))

### Private Attribute

- double [transx](#)
- double [da](#)

## 8.26.1 Ausführliche Beschreibung

Molodensky Parameters.

### Autor

unknown

### Bemerkungen

add header on 2018-02-02 by Yunhao Huang

## 8.26.2 Beschreibung der Konstruktoren und Destruktoren

### 8.26.2.1 MolodenskyParm()

```
params.MolodenskyParm.MolodenskyParm ( )
```

[Transformation](#) parameter from WGS84 to the geodesic datumstrategy. European Datum 1950 (ED50) for West Germany.

### Autor

Nima Mazroob

## 8.26.3 Dokumentation der Elementfunktionen

### 8.26.3.1 getDa()

```
double params.MolodenskyParm.getDa ( )
```

### 8.26.3.2 getDf()

```
double params.MolodenskyParm.getDf ( )
```

### 8.26.3.3 getTransx()

```
double params.MolodenskyParm.getTransx ( )
```

**8.26.3.4 getTransy()**

```
double params.MolodenskyParm.getTransy ( )
```

**8.26.3.5 getTransz()**

```
double params.MolodenskyParm.getTransz ( )
```

**8.26.3.6 setDa()**

```
void params.MolodenskyParm.setDa (
    double da )
```

**8.26.3.7 setDf()**

```
void params.MolodenskyParm.setDf (
    double df )
```

**8.26.3.8 setTransx()**

```
void params.MolodenskyParm.setTransx (
    double transx )
```

**8.26.3.9 setTransy()**

```
void params.MolodenskyParm.setTransy (
    double transy )
```

**8.26.3.10 setTransz()**

```
void params.MolodenskyParm.setTransz (
    double transz )
```

## 8.26.4 Dokumentation der Datenelemente

### 8.26.4.1 da

```
double params.MolodenskyParm.da [private]
```

### 8.26.4.2 transx

```
double params.MolodenskyParm.transx [private]
```

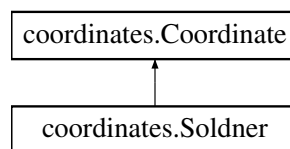
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [params/MolodenskyParm.java](#)

## 8.27 coordinates.Soldner Klassenreferenz

Class [Soldner](#) extends class [Coordinate](#).

Klassendiagramm für coordinates.Soldner:



## Öffentliche Methoden

- [Soldner](#) (double [l0](#), double [b0](#), double [abszisse](#), double [ordinate](#), double [height](#), int [ellipsoidal](#))
- [Soldner](#) (double [l0](#), double [b0](#))
- [Soldner](#) ()
- void [print](#) ()  
*prints [Soldner](#) coordinates with reference longitude, -latitude, abscissa, ordinate and height*
- [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ell)  
*getAsGeographic returns a [GeographicCoordinate](#)*
- void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ell, [GeographicCoordinateInterface](#) geocoord)  
*fromGeographic calculates [Soldner](#) coordinate from geographic coordinate*
- double [getL0](#) ()  
*getter method that returns the reference latitude of the particular [Soldner](#) coordinate*
- void [setL0](#) (double [l0](#))  
*setter method that accepts the reference latitude of the particular [Soldner](#) coordinate*
- double [getB0](#) ()  
*getter method that returns the reference longitude of the particular [Soldner](#) coordinate*

- void `setB0` (double `b0`)  
setter method that accepts the reference longitude of the particular *Soldner* coordinate
- double `getAbszisse` ()  
getter method that returns the abscissa of the particular *Soldner* coordinate
- void `setAbszisse` (double `abszisse`)  
setter method that accepts the abscissa of the particular *Soldner* coordinate
- double `getOrdinate` ()  
getter method that returns the ordinate of the particular *Soldner* coordinate
- void `setOrdinate` (double `ordinate`)  
setter method that accepts the ordinate of the particular *Soldner* coordinate
- double `getHeight` ()  
getter method that returns the height of the particular *Soldner* coordinate
- void `setHeight` (double `height`)  
setter method that accepts the height of the particular *Soldner* coordinate
- int `getEllipsoidal` ()  
getter method that returns the ellipsoidal of the particular *Soldner* coordinate
- void `setEllipsoidal` (int `ellipsoidal`)  
setter method that accepts the height above ellipsoid of the particular *Soldner* coordinate

## Private Attribute

- double `l0`
- double `b0`
- double `abszisse`
- double `ordinate`
- double `height`
- int `ellipsoidal`

### 8.27.1 Ausführliche Beschreibung

Class *Soldner* extends class *Coordinate*.

#### Autor

unknown

#### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

#### Version

0.1

#### Parameter

<i>l0</i>	- reference longitude as double
<i>b0</i>	- reference latitude as double
<i>abszisse</i>	- x-value as double
<i>ordinate</i>	- y-value as double
<i>height</i>	- height as double
<i>ellipsoidal</i>	- height above ellipsoid as integer

## 8.27.2 Beschreibung der Konstruktoren und Destruktoren

### 8.27.2.1 Soldner() [1/3]

```
coordinates.Soldner.Soldner (
    double l0,
    double b0,
    double abszisse,
    double ordinate,
    double height,
    int ellipsoidal )
```

### 8.27.2.2 Soldner() [2/3]

```
coordinates.Soldner.Soldner (
    double l0,
    double b0 )
```

### 8.27.2.3 Soldner() [3/3]

```
coordinates.Soldner.Soldner ( )
```

## 8.27.3 Dokumentation der Elementfunktionen

### 8.27.3.1 getAbszisse()

```
double coordinates.Soldner.getAbszisse ( )
```

getter method that returns the abscissa of the particular [Soldner](#) coordinate

#### Rückgabe

abszisse - returns the abscissa of the particular [Soldner](#) coordinate as double

### 8.27.3.2 getAsGeographicInterface()

```
GeographicCoordinateInterface coordinates.Soldner.getAsGeographicInterface (
    EllipsoidParms ell )
```

getAsGeographic returns a [GeographicCoordinate](#)

**Parameter**

<i>ell</i>	- the ellipsoidal parameters as an instance of <code>EllipsoidParms</code>
------------	--

**Rückgabe**

geo - output: returns `GeographicCoordinate`

Erneute Implementation von `coordinates.Coordinate`.

**8.27.3.3 getAsTargetCoordinate()**

```
void coordinates.Soldner.getAsTargetCoordinate (
    EllipsoidParms ell,
    GeographicCoordinateInterface geocoord )
```

fromGeographic calculates `Soldner` coordinate from geographic coordinate

**Parameter**

<i>ell</i>	- the ellipsoidal parameters as an instance of <code>EllipsoidParms</code>
<i>geocoord</i>	- a geographic coordinate as an instance of <code>GeographicCoordinate</code>

Erneute Implementation von `coordinates.Coordinate`.

**8.27.3.4 getB0()**

```
double coordinates.Soldner.getB0 ( )
```

getter method that returns the reference longitude of the particular `Soldner` coordinate

**Rückgabe**

b0 - returns the reference longitude of the particular `Soldner` coordinate as double

**8.27.3.5 getEllipsoidal()**

```
int coordinates.Soldner.getEllipsoidal ( )
```

getter method that returns the ellipsoidal of the particular `Soldner` coordinate

**Rückgabe**

ellipsoidal - returns the ellipsoidal of the particular `Soldner` coordinate as integer



#### 8.27.3.6 getHeight()

```
double coordinates.Soldner.getHeight ( )
```

getter method that returns the height of the particular [Soldner](#) coordinate

##### Rückgabe

height - returns the height of the particular [Soldner](#) coordinate as double

#### 8.27.3.7 getL0()

```
double coordinates.Soldner.getL0 ( )
```

getter method that returns the reference latitude of the particular [Soldner](#) coordinate

##### Rückgabe

l0 - returns the reference latitude of the particular [Soldner](#) coordinate as double

#### 8.27.3.8 getOrdinate()

```
double coordinates.Soldner.getOrdinate ( )
```

getter method that returns the ordinate of the particular [Soldner](#) coordinate

##### Rückgabe

ordinate - returns the ordinate of the particular [Soldner](#) coordinate as double

#### 8.27.3.9 print()

```
void coordinates.Soldner.print ( )
```

prints [Soldner](#) coordinates with reference longitude, -latitude, abscissa, ordinate and height

Erneute Implementation von [coordinates.Coordinate](#).

#### 8.27.3.10 setAbszisse()

```
void coordinates.Soldner.setAbszisse (
    double abszisse )
```

setter method that accepts the abscissa of the particular [Soldner](#) coordinate

## Parameter

<i>abszisse</i>	- sets a double representing the abscissa of the particular <a href="#">Soldner</a> coordinate
-----------------	--

**8.27.3.11 setB0()**

```
void coordinates.Soldner.setB0 (  
    double b0 )
```

setter method that accepts the reference longitude of the particular [Soldner](#) coordinate

## Parameter

<i>b0</i>	- sets a double representing the reference longitude of the particular <a href="#">Soldner</a> coordinate
-----------	---

**8.27.3.12 setEllipsoidal()**

```
void coordinates.Soldner.setEllipsoidal (  
    int ellipsoidal )
```

setter method that accepts the height above ellipsoid of the particular [Soldner](#) coordinate

## Parameter

<i>ellipsoidal</i>	- sets a integer representing the height above ellipsoid of the particular <a href="#">Soldner</a> coordinate
--------------------	---

**8.27.3.13 setHeight()**

```
void coordinates.Soldner.setHeight (  
    double height )
```

setter method that accepts the height of the particular [Soldner](#) coordinate

## Parameter

<i>height</i>	- sets a double representing the height of the particular <a href="#">Soldner</a> coordinate
---------------	--

#### 8.27.3.14 setL0()

```
void coordinates.Soldner.setL0 (
    double l0 )
```

setter method that accepts the reference latitude of the particular [Soldner](#) coordinate

##### Parameter

<i>l0</i>	- sets a double representing the reference latitude of the particular <a href="#">Soldner</a> coordinate
-----------	--

#### 8.27.3.15 setOrdinate()

```
void coordinates.Soldner.setOrdinate (
    double ordinate )
```

setter method that accepts the ordinate of the particular [Soldner](#) coordinate

##### Parameter

<i>ordinate</i>	- sets a double representing the ordinate of the particular <a href="#">Soldner</a> coordinate
-----------------	--

### 8.27.4 Dokumentation der Datenelemente

#### 8.27.4.1 abszisse

```
double coordinates.Soldner.abszisse [private]
```

x-value

#### 8.27.4.2 b0

```
double coordinates.Soldner.b0 [private]
```

reference latitude

#### 8.27.4.3 ellipsoidal

```
int coordinates.Soldner.ellipsoidal [private]
```

Height above ellipsoid: 0 => no height (not defined), 1 => ellipsoidal, 2 => geoid

#### 8.27.4.4 height

```
double coordinates.Soldner.height [private]
```

Height

#### 8.27.4.5 l0

```
double coordinates.Soldner.l0 [private]
```

reference longitude

#### 8.27.4.6 ordinate

```
double coordinates.Soldner.ordinate [private]
```

y-value

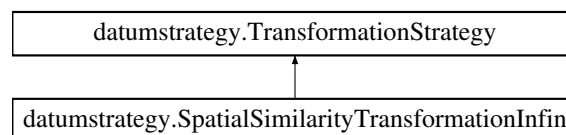
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- coordinates/[Soldner.java](#)

## 8.28 datumstrategy.SpatialSimilarityTransformationInfin Klassenreferenz

Algorithmic realization of the 3D similarity transformation also known as 7-parameter transformation.

Klassendiagramm für datumstrategy.SpatialSimilarityTransformationInfin:



### Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*abstract method for the transformation of a geographic coordinate*

### 8.28.1 Ausführliche Beschreibung

Algorithmic realization of the 3D similarity transformation also known as 7-parameter transformation.

The method takes all needed parameters from the class ControlParms that extends the class EllipsoidParms. Needed parameter for the 7-Param-Transformation: 3 translation-, 1 scale- and 3 rotation -parameters. The start coordinates are taken from the class GeographicCoordinate and have to be converted into cartesian coordinates to realize the transformation. The 7-Param-Transformation is done with cartesian coordinates. After the [Transformation](#), the new cartesian coordinates are converted into ellipsoidal coordinates by an iterative process.

source of conversions: Geodaetische Flaechenkoordinaten, Uebung2, kart2geo.pdf, 20.11.2014 source of 7-Param-Transformation: <https://de.wikipedia.org/wiki/Helmert-Transformation>

#### Autor

Andreas Eppler

#### Bemerkungen

adaptions for ControlParms Singleton on 11.12.2017 by Eva Majer  
use the methods of class XYZCoordinate to convert between ellipsoidal and cartesian coordinates on 19.12.17 by Markus Hillemann  
changed the class name (CamelCase) on 20.12.17 by Markus Hillemann  
implemented the calculation of the 7-Param-Transformation with matrices instead of singular values on 11.12.2019 by Steven Landgraf

#### Version

0.1

The method takes all needed parameters from the class ControlParms that extends the class EllipsoidParms. Needed parameter for the 7-Param-Transformation: 3 translation-, 1 scale- and 3 rotation -parameters. The start coordinates are taken from the class GeographicCoordinate and have to be converted into cartesian coordinates to realize the transformation. The 7-Param-Transformation is done with cartesian coordinates. After the [Transformation](#), the new cartesian coordinates are converted into ellipsoidal coordinates by an iterative process.

source of conversions: Geodaetische Flaechenkoordinaten, Uebung2, kart2geo.pdf, 20.11.2014 source of 7-Param-Transformation: <https://de.wikipedia.org/wiki/Helmert-Transformation>

#### Autor

Andreas Eppler

#### Bemerkungen

adaptions for ControlParms Singleton on 11.12.2017 by Eva Majer  
use the methods of class XYZCoordinate to convert between ellipsoidal and cartesian coordinates on 19.12.17 by Markus Hillemann  
changed the class name (CamelCase) on 20.12.17 by Markus Hillemann

•

#### Version

0.1

## 8.28.2 Dokumentation der Elementfunktionen

### 8.28.2.1 transform()

```
void datumstrategy.SpatialSimilarityTransformationInfin.transform (
    GeographicCoordinateInterface geographicCoordinate )
```

abstract method for the transformation of a geographic coordinate

#### Parameter

<i>geographicCoordinate</i>	- an object of type <code>GeographicCoordinateInterface</code> which is used for input/output for the transformation algorithm
-----------------------------	--

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [datumstrategy/SpatialSimilarityTransformationInfin.java](#)

## 8.29 test.datumstrategy.SpatialSimilarityTransformationInfinTest

### Klassenreferenz

Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.

#### Öffentliche Methoden

- void [testTransform](#) ()

#### Private Attribute

- final double [RHO](#) = 180/Math.PI
- final double [EPSILON\\_IN\\_METER](#) = 0.001
- final double [EPSILON\\_IN\\_RAD](#) = 0.00000001

### 8.29.1 Ausführliche Beschreibung

Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.

#### Autor

Steven Landgraf

#### Datum

December 2019

#### Version

0.1

## 8.29.2 Dokumentation der Elementfunktionen

### 8.29.2.1 testTransform()

```
void test.datumstrategy.SpatialSimilarityTransformationInfinTest.testTransform ( )
```

## 8.29.3 Dokumentation der Datenelemente

### 8.29.3.1 EPSILON\_IN\_METER

```
final double test.datumstrategy.SpatialSimilarityTransformationInfinTest.EPSILON_IN_METER =  
0.001 [private]
```

### 8.29.3.2 EPSILON\_IN\_RAD

```
final double test.datumstrategy.SpatialSimilarityTransformationInfinTest.EPSILON_IN_RAD = 0.↵  
00000001 [private]
```

### 8.29.3.3 RHO

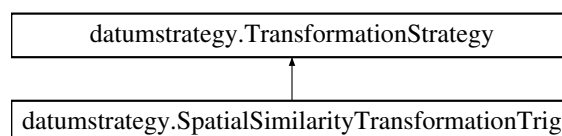
```
final double test.datumstrategy.SpatialSimilarityTransformationInfinTest.RHO = 180/Math.PI  
[private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/datumstrategy/[SpatialSimilarityTransformationInfinTest.java](#)

## 8.30 datumstrategy.SpatialSimilarityTransformationTrig Klassenreferenz

Klassendiagramm für datumstrategy.SpatialSimilarityTransformationTrig:



## Öffentliche Methoden

- void [transform](#) ([GeographicCoordinateInterface](#) geo)  
*abstract method for the transformation of a geographic coordinate*

### 8.30.1 Dokumentation der Elementfunktionen

#### 8.30.1.1 transform()

```
void datumstrategy.SpatialSimilarityTransformationTrig.transform (
    GeographicCoordinateInterface geographicCoordinate )
```

abstract method for the transformation of a geographic coordinate

##### Parameter

<i>geographicCoordinate</i>	- an object of type <a href="#">GeographicCoordinateInterface</a> which is used for input/output for the transformation algorithm
-----------------------------	---

Erneute Implementation von [datumstrategy.TransformationStrategy](#).

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- datumstrategy/[SpatialSimilarityTransformationTrig.java](#)

## 8.31 test.datumstrategy.SpatialSimilarityTransformationTrigTest Klassenreferenz

Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.

## Öffentliche Methoden

- void [testTransform](#) ()

## Private Attribute

- final double [RHO](#) = 180/Math.PI
- final double [EPSILON\\_IN\\_METER](#) = 0.001
- final double [EPSILON\\_IN\\_RAD](#) = 0.00000001



### 8.31.1 Ausführliche Beschreibung

Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.

#### Autor

Wentao Lu

#### Datum

Jan 2020

#### Version

0.1

### 8.31.2 Dokumentation der Elementfunktionen

#### 8.31.2.1 testTransform()

```
void test.datumstrategy.SpatialSimilarityTransformationTrigTest.testTransform ( )
```

### 8.31.3 Dokumentation der Datenelemente

#### 8.31.3.1 EPSILON\_IN\_METER

```
final double test.datumstrategy.SpatialSimilarityTransformationTrigTest.EPSILON_IN_METER =  
0.001 [private]
```

#### 8.31.3.2 EPSILON\_IN\_RAD

```
final double test.datumstrategy.SpatialSimilarityTransformationTrigTest.EPSILON_IN_RAD = 0.↔  
00000001 [private]
```

### 8.31.3.3 RHO

```
final double test.datumstrategy.SpatialSimilarityTransformationTrigTest.RHO = 180/Math.PI  
[private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- test/datumstrategy/[SpatialSimilarityTransformationTrigTest.java](#)

## 8.32 params.SpatialSimilarityTransformParm Klassenreferenz

Spatial similarity transform parameters.

### Öffentliche Methoden

- [SpatialSimilarityTransformParm](#) ()
- double [getTransx](#) ()
- void [setTransx](#) (double [transx](#))
- double [getTransy](#) ()
- void [setTransy](#) (double [transy](#))
- double [getTransz](#) ()
- void [setTransz](#) (double [transz](#))
- double [getAlpha](#) ()
- void [setAlpha](#) (double [alpha](#))
- double [getBeta](#) ()
- void [setBeta](#) (double [beta](#))
- double [getGamma](#) ()
- void [setGamma](#) (double [gamma](#))
- double [getScale](#) ()
- void [setScale](#) (double [scale](#))
- int [getFormula](#) ()
- void [setFormula](#) (int [formula](#))

### Private Attribute

- double [transx](#)
- double [alpha](#)
- double [scale](#)
- int [formula](#)

### 8.32.1 Ausführliche Beschreibung

Spatial similarity transform parameters.

#### Autor

unknown

#### Bemerkungen

add header on 2018-02-02 by Yunhao Huang

## 8.32.2 Beschreibung der Konstruktoren und Destruktoren

### 8.32.2.1 SpatialSimilarityTransformParm()

```
params.SpatialSimilarityTransformParm.SpatialSimilarityTransformParm ( )
```

## 8.32.3 Dokumentation der Elementfunktionen

### 8.32.3.1 getAlpha()

```
double params.SpatialSimilarityTransformParm.getAlpha ( )
```

### 8.32.3.2 getBeta()

```
double params.SpatialSimilarityTransformParm.getBeta ( )
```

### 8.32.3.3 getFormula()

```
int params.SpatialSimilarityTransformParm.getFormula ( )
```

### 8.32.3.4 getGamma()

```
double params.SpatialSimilarityTransformParm.getGamma ( )
```

### 8.32.3.5 getScale()

```
double params.SpatialSimilarityTransformParm.getScale ( )
```

**8.32.3.6 getTransx()**

```
double params.SpatialSimilarityTransformParm.getTransx ( )
```

**8.32.3.7 getTransy()**

```
double params.SpatialSimilarityTransformParm.getTransy ( )
```

**8.32.3.8 getTransz()**

```
double params.SpatialSimilarityTransformParm.getTransz ( )
```

**8.32.3.9 setAlpha()**

```
void params.SpatialSimilarityTransformParm.setAlpha (
    double alpha )
```

**8.32.3.10 setBeta()**

```
void params.SpatialSimilarityTransformParm.setBeta (
    double beta )
```

**8.32.3.11 setFormula()**

```
void params.SpatialSimilarityTransformParm.setFormula (
    int formula )
```

**8.32.3.12 setGamma()**

```
void params.SpatialSimilarityTransformParm.setGamma (
    double gamma )
```

**8.32.3.13 setScale()**

```
void params.SpatialSimilarityTransformParm.setScale (
    double scale )
```

**8.32.3.14 setTransx()**

```
void params.SpatialSimilarityTransformParm.setTransx (
    double transx )
```

**8.32.3.15 setTransy()**

```
void params.SpatialSimilarityTransformParm.setTransy (
    double transy )
```

**8.32.3.16 setTransz()**

```
void params.SpatialSimilarityTransformParm.setTransz (
    double transz )
```

**8.32.4 Dokumentation der Datenelemente****8.32.4.1 alpha**

```
double params.SpatialSimilarityTransformParm.alpha [private]
```

**8.32.4.2 formula**

```
int params.SpatialSimilarityTransformParm.formula [private]
```

**8.32.4.3 scale**

```
double params.SpatialSimilarityTransformParm.scale [private]
```

#### 8.32.4.4 transx

```
double params.SpatialSimilarityTransformParm.transx [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [params/SpatialSimilarityTransformParm.java](#)

### 8.33 Transformation Klassenreferenz

Main Transformatin class.

#### Öffentliche, statische Methoden

- static void [main](#) (String[] args)

#### 8.33.1 Ausführliche Beschreibung

Main Transformatin class.

##### Autor

Norbert Rösch

##### Version

0.1

##### Datum

January 2020

##### Warnung

For educational purposes only

##### Copyright

No licence restrictions yet

#### 8.33.2 Dokumentation der Elementfunktionen

### 8.33.2.1 main()

```
static void Transformation.main (
    String[] args ) [static]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [Transformation.java](#)

## 8.34 datumstrategy.TransformationStrategy Klassenreferenz

abstract class that encapsulates transformation algorithms for different coordinates

the concrete transformation algorithms that are implemented by the child classes of [TransformationStrategy](#) implement the abstract method [transform\(GeographicCoordinateInterface\)](#) that performs the transformation with geographic coordinates as input/output

Klassendiagramm für datumstrategy.TransformationStrategy:



### Öffentliche Methoden

- abstract void [transform](#) ([GeographicCoordinateInterface](#) geographicCoordinate)  
*abstract method for the transformation of a geographic coordinate*

### 8.34.1 Ausführliche Beschreibung

abstract class that encapsulates transformation algorithms for different coordinates

the concrete transformation algorithms that are implemented by the child classes of [TransformationStrategy](#) implement the abstract method [transform\(GeographicCoordinateInterface\)](#) that performs the transformation with geographic coordinates as input/output

#### Bemerkungen

last refactored 26.11.2016 by Patrick Huebner

- changed from interface to abstract class
- renamed to [TransformationStrategy](#)
- renamed method datum to transform(Coordinate, Coordinate) and made it protected
- added abstract method [transform\(GeographicCoordinateInterface\)](#)

last refactored 11.12.2017 by Eva Majer

last refactored 21.01.2019 by Norbert Rösch (now only object of GeographicCoordinateInterface are allowed to communicate with the different strategies)

- adaption for ControlParms Singleton
- Translation of some comments

## 8.34.2 Dokumentation der Elementfunktionen

### 8.34.2.1 transform()

```
abstract void datumstrategy.TransformationStrategy.transform (
    GeographicCoordinateInterface geographicCoordinate ) [abstract]
```

abstract method for the transformation of a geographic coordinate

#### Parameter

<i>geographicCoordinate</i>	- an object of type <code>GeographicCoordinateInterface</code> which is used for input/output for the transformation algorithm
-----------------------------	--

Erneute Implementation in `datumstrategy.MockStrategy`, `datumstrategy.CardansTransformation`, `datumstrategy.EulersTransformation`, `datumstrategy.MolodenskiiTransformationStandard`, `datumstrategy.MolodenskiiTransformationAbridged`, `datumstrategy.SpatialSimilarityTransformation` und `datumstrategy.SpatialSimilarityTransformationTrig`.

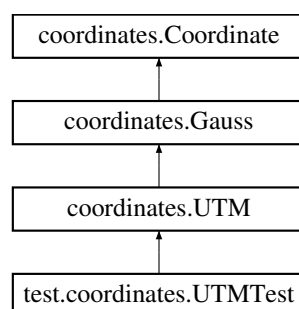
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- `datumstrategy/TransformationStrategy.java`

## 8.35 coordinates.UTM Klassenreferenz

Class `UTM` extends class `Gauss`.

Klassendiagramm für `coordinates.UTM`:



### Öffentliche Methoden

- `UTM ()`
- void `init ()`
- void `print ()`  
prints *UTM* coordinates
- `GeographicCoordinateInterface` `getAsGeographicInterface (EllipsoidParms ell)`  
*getAsGeographic* returns a *GeographicCoordinate* object



- void `getAsTargetCoordinate` (`EllipsoidParms` ell, `GeographicCoordinateInterface` geocoord)  
*fromGeographic calculates UTM coordinate from geographic coordinate*
- double `getEast` ()
- void `setEast` (double east)
- double `getNorth` ()
- void `setNorth` (double north)
- int `getZone` ()
- void `setZone` (int zone)
- boolean `isNorthhem` ()
- void `setNorthhem` (boolean northhem)

## Private Attribute

- double east
- double north
- int zone
- boolean northhem

## Weitere Geerbte Elemente

### 8.35.1 Ausführliche Beschreibung

Class `UTM` extends class `Gauss`.

#### Autor

unknown

#### Bemerkungen

updated header on 2017-11-29 by Markus Mueller

#### Version

0.1

#### Parameter

<i>east</i>	- easting as double
<i>north</i>	- Northing as double
<i>zone</i>	- UTM-Zone as integer
<i>northhem</i>	- hemisphere (0: North or 1:South) as boolean

### 8.35.2 Beschreibung der Konstruktoren und Destruktoren

### 8.35.2.1 UTM()

```
coordinates.UTM.UTM ( )
```

## 8.35.3 Dokumentation der Elementfunktionen

### 8.35.3.1 getAsGeographicInterface()

```
GeographicCoordinateInterface coordinates.UTM.getAsGeographicInterface (
    EllipsoidParms ell )
```

getAsGeographic returns a [GeographicCoordinate](#) object

#### Parameter

<i>ell</i>	- ellipsoidal parameters
------------	--------------------------

#### Rückgabe

geo - output: [GeographicCoordinate](#)

Erneute Implementation von [coordinates.Gauss](#).

### 8.35.3.2 getAsTargetCoordinate()

```
void coordinates.UTM.getAsTargetCoordinate (
    EllipsoidParms ell,
    GeographicCoordinateInterface geocoord )
```

fromGeographic calculates [UTM](#) coordinate from geographic coordinate

#### Parameter

<i>ell</i>	- the ellipsoidal parameters as an instance of <a href="#">EllipsoidParms</a>
<i>geocoord</i>	- a geographic coordinate as an instance of <a href="#">GeographicCoordinate</a>

Erneute Implementation von [coordinates.Coordinate](#).

### 8.35.3.3 getEast()

```
double coordinates.UTM.getEast ( )
```

**8.35.3.4 getNorth()**

```
double coordinates.UTM.getNorth ( )
```

**8.35.3.5 getZone()**

```
int coordinates.UTM.getZone ( )
```

**8.35.3.6 init()**

```
void coordinates.UTM.init ( )
```

**8.35.3.7 isNorthhem()**

```
boolean coordinates.UTM.isNorthhem ( )
```

**8.35.3.8 print()**

```
void coordinates.UTM.print ( )
```

prints [UTM](#) coordinates

Erneute Implementation von [coordinates.Gauss](#).

**8.35.3.9 setEast()**

```
void coordinates.UTM.setEast (
    double east )
```

**8.35.3.10 setNorth()**

```
void coordinates.UTM.setNorth (
    double north )
```

#### 8.35.3.11 setNorthhem()

```
void coordinates.UTM.setNorthhem (
    boolean northhem )
```

#### 8.35.3.12 setZone()

```
void coordinates.UTM.setZone (
    int zone )
```

### 8.35.4 Dokumentation der Datenelemente

#### 8.35.4.1 east

```
double coordinates.UTM.east [private]
```

Easting

#### 8.35.4.2 north

```
double coordinates.UTM.north [private]
```

Northing

#### 8.35.4.3 northhem

```
boolean coordinates.UTM.northhem [private]
```

Hemisphere: 0 => North, 1 => South

#### 8.35.4.4 zone

```
int coordinates.UTM.zone [private]
```

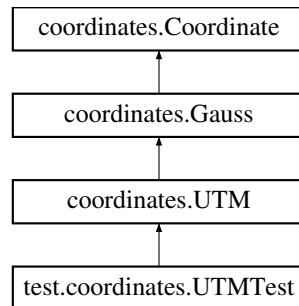
[coordinates.UTM](#)-Zone

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/UTM.java](#)

## 8.36 test.coordinates.UTMTest Klassenreferenz

Klassendiagramm für test.coordinates.UTMTest:



### Öffentliche Methoden

- void [testGetAsGeographicInterface](#) ()
- void [testGetAsUTM](#) ()

### Weitere Geerbte Elemente

#### 8.36.1 Dokumentation der Elementfunktionen

##### 8.36.1.1 testGetAsGeographicInterface()

```
void test.coordinates.UTMTest.testGetAsGeographicInterface ( )
```

##### 8.36.1.2 testGetAsUTM()

```
void test.coordinates.UTMTest.testGetAsUTM ( )
```

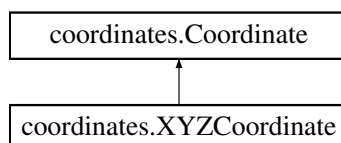
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [test/coordinates/UTMTest.java](#)

## 8.37 coordinates.XYZCoordinate Klassenreferenz

Class [XYZCoordinate](#).

Klassendiagramm für coordinates.XYZCoordinate:



## Öffentliche Methoden

- [XYZCoordinate](#) ()  
*main constructor to set the coordinates*
- void [init](#) ()
- [XYZCoordinate](#) (double xValue, double yValue, double zValue)  
*Constructor for generating a geocentricCoordinate.*
- [XYZCoordinate](#) (double xValue, double yValue)
- [GeographicCoordinateInterface](#) [getAsGeographicInterface](#) ([EllipsoidParms](#) ell)  
*Function for transforming geocentric coordinates to geographic coordinates.*
- void [getAsTargetCoordinate](#) ([EllipsoidParms](#) ell, [GeographicCoordinateInterface](#) geocoord)  
*Function for transforming geographic coordinates to geocentric coordinates.*
- void [print](#) ()  
*abstract methnd that allows the concrete [Coordinate](#) to be output on screen*
- double [getX](#) ()  
*getter method that returns the x-coordinate of the particular [XYZCoordinate](#)*
- void [setX](#) (double x)  
*setter method that accepts x-coordinate for the particular [XYZCoordinate](#)*
- double [getY](#) ()  
*getter method that returns the y-coordinate of the particular [XYZCoordinate](#)*
- void [setY](#) (double y)  
*setter method that accepts y-coordinate for the particular [XYZCoordinate](#)*
- double [getZ](#) ()  
*getter method that returns the z-coordinate of the particular [XYZCoordinate](#)*
- void [setZ](#) (double z)  
*setter method that accepts z-coordinate for the particular [XYZCoordinate](#)*

## Private Attribute

- double [x](#)

### 8.37.1 Ausführliche Beschreibung

Class [XYZCoordinate](#).

#### Autor

David Li

#### Bemerkungen

2016-12-04 removed unused comments and variables by David Li

2016-12-04 added new method [calculateRotationMatrix\(SpatialSimilarityTransformParm\)](#) by David Li

2018-01-29 added constructor with [ControlParms](#) by Markus Mueller

#### Version

0.1

## Parameter

<i>x</i>	- x-coordinate as double
<i>y</i>	- y-coordinate as double
<i>z</i>	- z-coordinate as double

## 8.37.2 Beschreibung der Konstruktoren und Destruktoren

### 8.37.2.1 XYZCoordinate() [1/3]

```
coordinates.XYZCoordinate.XYZCoordinate ( )
```

main constructor to set the coordinates

### 8.37.2.2 XYZCoordinate() [2/3]

```
coordinates.XYZCoordinate.XYZCoordinate (
    double xValue,
    double yValue,
    double zValue )
```

Constructor for generating a geocentricCoordinate.

## Bemerkungen

parameters changed to x-, y- and zAxis (were too inexpressive x1, x2 and x3) by unknown

2018-01-25 renamed parameters to xValue, yValue, zValue (were xAxis, yAxis, zAxis before) by Markus MÄ¼ller

## Parameter

<i>xValue</i>	- x component of XYZ coordinate
<i>yValue</i>	- y component of XYZ coordinate
<i>zValue</i>	- z component of XYZ coordinate

### 8.37.2.3 XYZCoordinate() [3/3]

```
coordinates.XYZCoordinate.XYZCoordinate (
    double xValue,
    double yValue )
```

### 8.37.3 Dokumentation der Elementfunktionen

#### 8.37.3.1 `getAsGeographicInterface()`

```
GeographicCoordinateInterface coordinates.XYZCoordinate.getAsGeographicInterface (
    EllipsoidParms ell )
```

Function for transforming geocentric coordinates to geographic coordinates.

Implemented by Michael Loos on 16.06.1998, based on a FORTRAN-Subroutine from 12.08.1993 by Hansjoerg Kutterer

=> Needs the function 'Constant' from [LatParm.java](#)!

##### Autor

Michael Loos

##### Bemerkungen

renamed the variable bralt to altitude by David Li 2016-12-04

renamed the variable eps to maxLonLatDistance by David Li 2016-12-04

formal changes in summer 2000 by Daniel Mueller

##### Parameter

<i>ell</i>	- an object of type EllipsoidParms
------------	------------------------------------

##### Rückgabe

geo - returns geographic coordinate consisting of the double values latitude, longitude and ellipsoidal height

Erneute Implementation von [coordinates.Coordinate](#).

#### 8.37.3.2 `getAsTargetCoordinate()`

```
void coordinates.XYZCoordinate.getAsTargetCoordinate (
    EllipsoidParms ell,
    GeographicCoordinateInterface geocoord )
```

Function for transforming geographic coordinates to geocentric coordinates.

##### Autor

Michael Loos, 1998-06-06 as of a FORTRAN-Subroutine from Hansjoerg Kutterer, 1993-08-12



**Bemerkungen**

formal changes in Summer 2000 by Daniel Mueller  
Java-Version in February 2003 by Norbert Roesch

Algorithm as of G. Schmitt, M. Illner and R. Jaeger, "Transformationsprobleme", pp. 125-142 in Mitteilungen des D↔VW Landesvereins BaWue, Sonderheft "GPS und Integration von GPS in bestehende geodaetische Netze", 1991.

**Parameter**

<i>ell</i>	- an object of type <code>EllipsoidParms</code> that represents ellipsoidal parameters
<i>geocoord</i>	- an object of type <code>GeographicCoordinate</code> that represents the geographic coordinate to transform

Erneute Implementation von `coordinates.Coordinate`.

**8.37.3.3 getX()**

```
double coordinates.XYZCoordinate.getX ( )
```

getter method that returns the x-coordinate of the particular `XYZCoordinate`

**Rückgabe**

x - returns the x-coordinate value of the particular `Coordinate` point as double

**8.37.3.4 getY()**

```
double coordinates.XYZCoordinate.getY ( )
```

getter method that returns the y-coordinate of the particular `XYZCoordinate`

**Rückgabe**

y - returns the y-coordinate value of the particular `Coordinate` point as double

**8.37.3.5 getZ()**

```
double coordinates.XYZCoordinate.getZ ( )
```

getter method that returns the z-coordinate of the particular `XYZCoordinate`

**Rückgabe**

z - returns the z-coordinate value of the particular `Coordinate` point as double

### 8.37.3.6 init()

```
void coordinates.XYZCoordinate.init ( )
```

### 8.37.3.7 print()

```
void coordinates.XYZCoordinate.print ( )
```

abstract methnd that allows the concrete [Coordinate](#) to be output on screen

Erneute Implementation von [coordinates.Coordinate](#).

### 8.37.3.8 setX()

```
void coordinates.XYZCoordinate.setX (
    double x )
```

setter method that accepts x-coordinate for the particular [XYZCoordinate](#)

#### Parameter

<i>x</i>	- sets a double representing the x-coordinate value for the particular <a href="#">XYZCoordinate</a>
----------	--

### 8.37.3.9 setY()

```
void coordinates.XYZCoordinate.setY (
    double y )
```

setter method that accepts y-coordinate for the particular [XYZCoordinate](#)

#### Parameter

<i>y</i>	- sets a double representing the y-coordinate value for the particular <a href="#">XYZCoordinate</a>
----------	--

### 8.37.3.10 setZ()

```
void coordinates.XYZCoordinate.setZ (
    double z )
```

setter method that accepts z-coordinate for the particular [XYZCoordinate](#)

## Parameter

z	- sets a double representing the z-coordinate value for the particular <a href="#">XYZCoordinate</a>
---	--

## 8.37.4 Dokumentation der Datenelemente

### 8.37.4.1 x

```
double coordinates.XYZCoordinate.x [private]
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [coordinates/XYZCoordinate.java](#)

## 8.38 test.coordinates.XYZCoordinateTest Klassenreferenz

### Öffentliche Methoden

- void [testGetAsGeographicInterface](#) ()
- void [testGetAsTargetCoordinate](#) ()

### 8.38.1 Dokumentation der Elementfunktionen

#### 8.38.1.1 testGetAsGeographicInterface()

```
void test.coordinates.XYZCoordinateTest.testGetAsGeographicInterface ( )
```

#### Bemerkungen

got reference data from [www.sapos-bw.de/trafoErg\\_B\\_X.php](http://www.sapos-bw.de/trafoErg_B_X.php)

#### 8.38.1.2 testGetAsTargetCoordinate()

```
void test.coordinates.XYZCoordinateTest.testGetAsTargetCoordinate ( )
```

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- [test/coordinates/XYZCoordinateTest.java](#)



## Kapitel 9

# Datei-Dokumentation

### 9.1 C:/Users/YangXiaofei/Desktop/Workplace0115/Transformation1/bin/↵ README.md-Dateireferenz

### 9.2 coordinates/Coordinate.java-Dateireferenz

#### Klassen

- class [coordinates.Coordinate](#)

*Abstract class that represents a coordinate. Enables any concrete [Coordinate](#) class be used as input for TransformationStrategy which relies on geographic coordinates as input/output*

#### Pakete

- package [coordinates](#)

### 9.3 coordinates/Gauss.java-Dateireferenz

#### Klassen

- class [coordinates.Gauss](#)

*Class [Gauss](#) extends class [Coordinate](#).*

#### Pakete

- package [coordinates](#)

## 9.4 coordinates/GaussKrueger.java-Dateireferenz

### Klassen

- class [coordinates.GaussKrueger](#)  
*[Coordinate](#) point relating to a certain Gauss-Krueger reference meridian.*

### Pakete

- package [coordinates](#)

## 9.5 coordinates/GeographicCoordinate.java-Dateireferenz

### Klassen

- class [coordinates.GeographicCoordinate](#)  
*concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (as input coordinate)*

### Pakete

- package [coordinates](#)

## 9.6 coordinates/GeographicCoordinateInterface.java-Dateireferenz

### Klassen

- class [coordinates.GeographicCoordinateInterface](#)  
*concrete implementation of abstract class [Coordinate](#), represents a geographic coordinate (As interface for strategy)*

### Pakete

- package [coordinates](#)

## 9.7 coordinates/GetAppropriateCoordinate.java-Dateireferenz

### Klassen

- class [coordinates.GetAppropriateCoordinate](#)  
*[Coordinate](#) factory for plane coordinates and geographic coordinates.*

## Pakete

- package [coordinates](#)

## 9.8 coordinates/Soldner.java-Dateireferenz

### Klassen

- class [coordinates.Soldner](#)  
*Class [Soldner](#) extends class [Coordinate](#).*

## Pakete

- package [coordinates](#)

## 9.9 coordinates/UTM.java-Dateireferenz

### Klassen

- class [coordinates.UTM](#)  
*Class [UTM](#) extends class [Gauss](#).*

## Pakete

- package [coordinates](#)

## 9.10 coordinates/XYZCoordinate.java-Dateireferenz

### Klassen

- class [coordinates.XYZCoordinate](#)  
*Class [XYZCoordinate](#).*

## Pakete

- package [coordinates](#)

## 9.11 datumstrategy/CardansTransformation.java-Dateireferenz

### Klassen

- class [datumstrategy.CardansTransformation](#)  
*Algorithmic realization of Translation and the Cardan's Rotation.*

## Pakete

- package [datumstrategy](#)

## 9.12 datumstrategy/EulersTransformation.java-Dateireferenz

### Klassen

- class [datumstrategy.EulersTransformation](#)  
*Algorithmic realization of the Euler's transformation.*

## Pakete

- package [datumstrategy](#)

## 9.13 datumstrategy/GetAppropriateTransformationAlgorithm.java- Dateireferenz

### Klassen

- class [datumstrategy.GetAppropriateTransformationAlgorithm](#)

## Pakete

- package [datumstrategy](#)

## 9.14 datumstrategy/MockStrategy.java-Dateireferenz

### Klassen

- class [datumstrategy.MockStrategy](#)

## Pakete

- package [datumstrategy](#)

## 9.15 datumstrategy/MolodenskiiTransformationAbridged.java- Dateireferenz

### Klassen

- class [datumstrategy.MolodenskiiTransformationAbridged](#)  
*Algorithmic realization of the abridged molodensky transformation.*



## Pakete

- package [datumstrategy](#)

## 9.16 datumstrategy/MolodenskiiTransformationStandard.java- Dateireferenz

## Klassen

- class [datumstrategy.MolodenskiiTransformationStandard](#)  
*Algorithmic realization of the molodensky transformation.*

## Pakete

- package [datumstrategy](#)

## 9.17 datumstrategy/SpatialSimilarityTransformationInfin.java- Dateireferenz

## Klassen

- class [datumstrategy.SpatialSimilarityTransformationInfin](#)  
*Algorithmic realization of the 3D similarity transformation also known as 7-parameter transformation.*

## Pakete

- package [datumstrategy](#)

## 9.18 datumstrategy/SpatialSimilarityTransformationTrig.java- Dateireferenz

## Klassen

- class [datumstrategy.SpatialSimilarityTransformationTrig](#)

## Pakete

- package [datumstrategy](#)

## 9.19 datumstrategy/TransformationStrategy.java-Dateireferenz

contains abstract class TransformationStrategy

## Klassen

- class [datumstrategy.TransformationStrategy](#)

*abstract class that encapsulates transformation algorithms for different coordinates  
the concrete transformation algorithms that are implemented by the child classes of [TransformationStrategy](#) implement the abstract method [transform\(GeographicCoordinateInterface\)](#) that performs the transformation with geographic coordinates as input/output*

## Pakete

- package [datumstrategy](#)

### 9.19.1 Ausführliche Beschreibung

contains abstract class TransformationStrategy

Autor

unknown as of 11.2016

## 9.20 [params/ControlParms.java](#)-Dateireferenz

### Klassen

- class [params.ControlParms](#)

*Class [ControlParms](#) extends class [EllipsoidParms](#).*

### Pakete

- package [params](#)

## 9.21 [params/EllipsoidParms.java](#)-Dateireferenz

### Klassen

- class [params.EllipsoidParms](#)

*Ellipsoid parameters.*

### Pakete

- package [params](#)

## 9.22 params/LatParm.java-Dateireferenz

### Klassen

- class [params.LatParm](#)  
*Latitude Parameters.*

### Pakete

- package [params](#)

## 9.23 params/MolodenskyParm.java-Dateireferenz

### Klassen

- class [params.MolodenskyParm](#)  
*Molodensky Parameters.*

### Pakete

- package [params](#)

## 9.24 params/SpatialSimilarityTransformParm.java-Dateireferenz

### Klassen

- class [params.SpatialSimilarityTransformParm](#)  
*Spatial similarity transform parameters.*

### Pakete

- package [params](#)

## 9.25 test/coordinates/AllTests.java-Dateireferenz

### Klassen

- class [test.coordinates.AllTests](#)

### Pakete

- package [test.coordinates](#)

## 9.26 test/coordinates/CoordinateFactoryTest.java-Dateireferenz

### Klassen

- class [test.coordinates.CoordinateFactoryTest](#)  
*Coordinate factory for plane coordinates and geographic coordinates.*

### Pakete

- package [test.coordinates](#)

## 9.27 test/coordinates/GaussKruegerTest.java-Dateireferenz

### Klassen

- class [test.coordinates.GaussKruegerTest](#)  
*Test class with unit tests to test behavior class GaussKrueger.*

### Pakete

- package [test.coordinates](#)

## 9.28 test/coordinates/GaussTest.java-Dateireferenz

### Klassen

- class [test.coordinates.GaussTest](#)  
*Test class with unit tests to test behavior class Gauss.*

### Pakete

- package [test.coordinates](#)

## 9.29 test/coordinates/GeographicCoordinateTest.java-Dateireferenz

### Klassen

- class [test.coordinates.GeographicCoordinateTest](#)

### Pakete

- package [test.coordinates](#)

## 9.30 test/coordinates/UTMTest.java-Dateireferenz

### Klassen

- class [test.coordinates.UTMTest](#)

### Pakete

- package [test.coordinates](#)

## 9.31 test/coordinates/XYZCoordinateTest.java-Dateireferenz

### Klassen

- class [test.coordinates.XYZCoordinateTest](#)

### Pakete

- package [test.coordinates](#)

## 9.32 test/datumstrategy/GetAppropriateTransformationAlgorithmTest.java-Dateireferenz

### Klassen

- class [test.datumstrategy.GetAppropriateTransformationAlgorithmTest](#)

### Pakete

- package [test.datumstrategy](#)

## 9.33 test/datumstrategy/MockStrategyTest.java-Dateireferenz

### Klassen

- class [test.datumstrategy.MockStrategyTest](#)

*Test class with two integration tests to test the transition of a Gauss-Krueger-coordinate from the third to the fourth strip and vice versa. The datumstrategy "MockStrategy" is used.*

### Pakete

- package [test.datumstrategy](#)

### 9.34 test/datumstrategy/MolodenskiiTransformationAbridgedTest.java-↵ Dateireferenz

#### Klassen

- class [test.datumstrategy.MolodenskiiTransformationAbridgedTest](#)  
*Test of the algorithmic realization of the abridged molodensky transformation.*

#### Pakete

- package [test.datumstrategy](#)

### 9.35 test/datumstrategy/MolodenskiiTransformationStandardTest.java-↵ Dateireferenz

#### Klassen

- class [test.datumstrategy.MolodenskiiTransformationStandardTest](#)  
*Tests the Molodenskii [Transformation](#) Standard as a Datum Strategy. Input values and expected values were calculated by hand.*

#### Pakete

- package [test.datumstrategy](#)

### 9.36 test/datumstrategy/SpatialSimilarityTransformationInfinTest.java-↵ Dateireferenz

#### Klassen

- class [test.datumstrategy.SpatialSimilarityTransformationInfinTest](#)  
*Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.*

#### Pakete

- package [test.datumstrategy](#)

### 9.37 test/datumstrategy/SpatialSimilarityTransformationTrigTest.java-↵ Dateireferenz

#### Klassen

- class [test.datumstrategy.SpatialSimilarityTransformationTrigTest](#)  
*Tests the Spatial Similarity [Transformation](#) as a Datum Strategy. Input values and expected values were calculated by hand.*

## Pakete

- package [test.datumstrategy](#)

## 9.38 test/params/LatParmTest.java-Dateireferenz

## Klassen

- class [test.params.LatParmTest](#)

## Pakete

- package [test.params](#)

## 9.39 Transformation.java-Dateireferenz

## Klassen

- class [Transformation](#)  
*Main Transformatin class.*





# Index

- abszisse
  - coordinates.Gauss, [64](#)
  - coordinates.Soldner, [119](#)
- alpha
  - params.SpatialSimilarityTransformParm, [129](#)
- ausgabetypp
  - params.ControlParms, [35](#)
- b0
  - coordinates.Soldner, [119](#)
- c
  - params.EllipsoidParms, [53](#)
- C:/Users/YangXiaofei/Desktop/Workplace0115/Transformation1/bin/README.md,  
[145](#)
- CalculateAbszisseOrdinate
  - coordinates.Gauss, [60](#)
- calculatelongitudeParms
  - params.EllipsoidParms, [47](#)
- cbr
  - params.LatParm, [99](#)
- centralMeridian
  - coordinates.GaussKrueger, [70](#)
- changedatum
  - params.ControlParms, [35](#)
- Constant
  - params.LatParm, [96](#)
- ControlParms
  - params.ControlParms, [23](#)
- coordinates, [13](#)
- coordinates.Coordinate, [40](#)
  - getAsGeographicInterface, [41](#)
  - getAsTargetCoordinate, [42](#)
  - getPointid, [42](#)
  - pointid, [43](#)
  - print, [42](#)
  - setPointid, [43](#)
- coordinates.Gauss, [58](#)
  - abszisse, [64](#)
  - CalculateAbszisseOrdinate, [60](#)
  - ellipsoidal, [64](#)
  - getAbszisse, [60](#)
  - getAsGeographicInterface, [60](#)
  - getEllipsoidal, [61](#)
  - getHeight, [61](#)
  - getOrdinate, [61](#)
  - getScale, [61](#)
  - height, [64](#)
  - meridianLength, [62](#)
  - ordinate, [64](#)
  - print, [62](#)
  - scale, [64](#)
  - setAbszisse, [62](#)
  - setEllipsoidal, [62](#)
  - setHeight, [63](#)
  - setOrdinate, [63](#)
  - setScale, [63](#)
- coordinates.GaussKrueger, [64](#)
  - centralMeridian, [70](#)
  - GaussKrueger, [66](#)
  - getAsGeographicInterface, [66](#)
  - getAsTargetCoordinate, [66](#)
  - getHoch, [68](#)
  - getllao, [68](#)
  - getRechts, [68](#)
  - hoch, [70](#)
  - print, [68](#)
  - rechts, [70](#)
  - setHoch, [69](#)
  - setllao, [70](#)
  - setRechts, [70](#)
- coordinates.GeographicCoordinate, [74](#)
  - ellipsoidal, [81](#)
  - GeographicCoordinate, [76](#)
  - getAsGeographicInterface, [76](#)
  - getAsTargetCoordinate, [77](#)
  - getEllipsoidal, [77](#)
  - getHeight, [77](#)
  - getInstance, [77](#), [78](#)
  - getLatitude, [78](#)
  - getLongitude, [78](#)
  - getPoint\_number, [79](#)
  - height, [81](#)
  - latitude, [81](#)
  - longitude, [81](#)
  - myInstance, [81](#)
  - point\_number, [81](#)
  - print, [79](#)
  - resetState, [79](#)
  - setEllipsoidal, [79](#)
  - setHeight, [80](#)
  - setLatitude, [80](#)
  - setLongitude, [80](#)
  - setPoint\_number, [80](#)
- coordinates.GeographicCoordinateInterface, [82](#)
  - copy, [85](#)
  - ellipsoidal, [91](#)
  - fupubre, [85](#)
  - GeographicCoordinateInterface, [84](#), [85](#)

- GeographicLongitudeLatitude, 86
- getAsGeographicInterface, 86
- getAsTargetCoordinate, 86
- getEllipsoidal, 87
- getHeight, 87
- getInstance, 87, 88
- getLatitude, 88
- getLongitude, 88
- getPoint\_number, 88
- height, 91
- latitude, 91
- longitude, 91
- molodenskytrafo, 89
- myInstance, 91
- point\_number, 92
- print, 89
- resetState, 89
- setEllipsoidal, 89
- setHeight, 90
- setLatitude, 90
- setLongitude, 90
- setPoint\_number, 91
- coordinates.GetAppropriateCoordinate, 93
  - getCoord, 94
  - getGeographicCoordinateInterface, 94
- coordinates.Soldner, 113
  - abszisse, 119
  - b0, 119
  - ellipsoidal, 119
  - getAbszisse, 115
  - getAsGeographicInterface, 115
  - getAsTargetCoordinate, 116
  - getB0, 116
  - getEllipsoidal, 116
  - getHeight, 116
  - getL0, 117
  - getOrdinate, 117
  - height, 119
  - l0, 120
  - ordinate, 120
  - print, 117
  - setAbszisse, 117
  - setB0, 118
  - setEllipsoidal, 118
  - setHeight, 118
  - setL0, 118
  - setOrdinate, 119
  - Soldner, 115
- coordinates.UTM, 132
  - east, 136
  - getAsGeographicInterface, 134
  - getAsTargetCoordinate, 134
  - getEast, 134
  - getNorth, 134
  - getZone, 135
  - init, 135
  - isNorthhem, 135
  - north, 136
  - northhem, 136
  - print, 135
  - setEast, 135
  - setNorth, 135
  - setNorthhem, 135
  - setZone, 136
  - UTM, 133
  - zone, 136
- coordinates.XYZCoordinate, 137
  - getAsGeographicInterface, 140
  - getAsTargetCoordinate, 140
  - getX, 141
  - getY, 141
  - getZ, 141
  - init, 141
  - print, 142
  - setX, 142
  - setY, 142
  - setZ, 142
  - x, 143
  - XYZCoordinate, 139
- coordinates/Coordinate.java, 145
- coordinates/Gauss.java, 145
- coordinates/GaussKrueger.java, 146
- coordinates/GeographicCoordinate.java, 146
- coordinates/GeographicCoordinateInterface.java, 146
- coordinates/GetAppropriateCoordinate.java, 146
- coordinates/Soldner.java, 147
- coordinates/UTM.java, 147
- coordinates/XYZCoordinate.java, 147
- copy
  - coordinates.GeographicCoordinateInterface, 85
- da
  - params.ControlParms, 35
  - params.MolodenskyParm, 113
- datumstrategy, 13
- datumstrategy.CardansTransformation, 17
  - e1, 18
  - e2, 19
  - e3, 19
  - e4, 19
  - e5, 19
  - e6, 19
  - e7, 19
  - e8, 19
  - e9, 19
  - transform, 18
- datumstrategy.EulersTransformation, 55
  - e1, 57
  - e2, 57
  - e3, 57
  - e4, 57
  - e5, 57
  - e6, 57
  - e7, 57
  - e8, 57
  - e9, 58
  - transform, 56

- datumstrategy.GetAppropriateTransformationAlgorithm, 94
  - getStrategy, 94
- datumstrategy.MockStrategy, 101
  - transform, 101
- datumstrategy.MolodenskiiTransformationAbridged, 103
  - getInstance, 104
  - MolodenskiiTransformationAbridged, 104
  - myInstance, 105
  - transform, 104
- datumstrategy.MolodenskiiTransformationStandard, 107
  - getInstance, 108
  - MolodenskiiTransformationStandard, 108
  - myInstance, 108
  - transform, 108
- datumstrategy.SpatialSimilarityTransformationInfin, 120
  - transform, 122
- datumstrategy.SpatialSimilarityTransformationTrig, 123
  - transform, 124
- datumstrategy.TransformationStrategy, 131
  - transform, 132
- datumstrategy/CardansTransformation.java, 147
- datumstrategy/EulersTransformation.java, 148
- datumstrategy/GetAppropriateTransformationAlgorithm.java, 148
- datumstrategy/MockStrategy.java, 148
- datumstrategy/MolodenskiiTransformationAbridged.java, 148
- datumstrategy/MolodenskiiTransformationStandard.java, 149
- datumstrategy/SpatialSimilarityTransformationInfin.java, 149
- datumstrategy/SpatialSimilarityTransformationTrig.java, 149
- datumstrategy/TransformationStrategy.java, 149
- destinationCoordinateX
  - params.ControlParms, 35
- destinationCoordinateY
  - params.ControlParms, 36
- destinationCoordinateZ
  - params.ControlParms, 36
- df
  - params.ControlParms, 36
- dx
  - params.ControlParms, 36
- dy
  - params.ControlParms, 36
- dz
  - params.ControlParms, 36
- e1
  - datumstrategy.CardansTransformation, 18
  - datumstrategy.EulersTransformation, 57
- e2
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e3
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e4
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e5
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e6
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e7
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e8
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 57
- e9
  - datumstrategy.CardansTransformation, 19
  - datumstrategy.EulersTransformation, 58
- east
  - coordinates.UTM, 136
- ellipsoidal
  - coordinates.Gauss, 64
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 91
  - coordinates.Soldner, 119
- EllipsoidParms
  - params.EllipsoidParms, 47
- EPSILON\_IN\_DEGREE
  - test.coordinates.GaussKruegerTest, 72
- EPSILON\_IN\_METER
  - test.coordinates.GaussKruegerTest, 72
  - test.coordinates.GaussTest, 73
  - test.datumstrategy.MockStrategyTest, 103
  - test.datumstrategy.MolodenskiiTransformationStandardTest, 109
  - test.datumstrategy.SpatialSimilarityTransformationInfinTest, 123
  - test.datumstrategy.SpatialSimilarityTransformationTrigTest, 125
- EPSILON\_IN\_RAD
  - test.datumstrategy.MolodenskiiTransformationStandardTest, 110
  - test.datumstrategy.SpatialSimilarityTransformationInfinTest, 123
  - test.datumstrategy.SpatialSimilarityTransformationTrigTest, 125
- EPSILON\_meter
  - test.datumstrategy.MolodenskiiTransformationAbridgedTest, 106
- EPSILON\_rad
  - test.datumstrategy.MolodenskiiTransformationAbridgedTest, 106
- EPSILONFORDEGREE
  - test.coordinates.GeographicCoordinateTest, 92
- etabr
  - params.LatParm, 99
- fileinput
  - params.ControlParms, 36

- fileoutput
  - params.ControlParms, 36
- firstEccentricity
  - params.EllipsoidParms, 53
- flattening
  - params.EllipsoidParms, 53
- formula
  - params.SpatialSimilarityTransformParm, 129
- fromawert
  - params.ControlParms, 37
- frombwert
  - params.ControlParms, 37
- fromdatum
  - params.ControlParms, 37
- fromprojection
  - params.ControlParms, 37
- fupubre
  - coordinates.GeographicCoordinateInterface, 85
- GaussKrueger
  - coordinates.GaussKrueger, 66
- GeographicCoordinate
  - coordinates.GeographicCoordinate, 76
- GeographicCoordinateInterface
  - coordinates.GeographicCoordinateInterface, 84, 85
- GeographicLongitudeLatitude
  - coordinates.GeographicCoordinateInterface, 86
- getAbszisse
  - coordinates.Gauss, 60
  - coordinates.Soldner, 115
- getAlpha
  - params.SpatialSimilarityTransformParm, 127
- getAsGaussKrueger
  - test.coordinates.GaussKruegerTest, 71
- getAsGeographic
  - test.coordinates.GaussKruegerTest, 72
- getAsGeographicInterface
  - coordinates.Coordinate, 41
  - coordinates.Gauss, 60
  - coordinates.GaussKrueger, 66
  - coordinates.GeographicCoordinate, 76
  - coordinates.GeographicCoordinateInterface, 86
  - coordinates.Soldner, 115
  - coordinates.UTM, 134
  - coordinates.XYZCoordinate, 140
- getAsTargetCoordinate
  - coordinates.Coordinate, 42
  - coordinates.GaussKrueger, 66
  - coordinates.GeographicCoordinate, 77
  - coordinates.GeographicCoordinateInterface, 86
  - coordinates.Soldner, 116
  - coordinates.UTM, 134
  - coordinates.XYZCoordinate, 140
- getAusgabetyt
  - params.ControlParms, 23
- getB0
  - coordinates.Soldner, 116
- getBeta
  - params.SpatialSimilarityTransformParm, 127
- getC
  - params.EllipsoidParms, 47
- getCbr
  - params.LatParm, 96
- getCoord
  - coordinates.GetAppropriateCoordinate, 94
- getDa
  - params.ControlParms, 23
  - params.MolodenskyParm, 111
- getDestinationCoordinateX
  - params.ControlParms, 23
- getDestinationCoordinateY
  - params.ControlParms, 24
- getDestinationCoordinateZ
  - params.ControlParms, 24
- getDf
  - params.ControlParms, 24
  - params.MolodenskyParm, 111
- getDx
  - params.ControlParms, 24
- getDy
  - params.ControlParms, 24
- getDz
  - params.ControlParms, 24
- getEast
  - coordinates.UTM, 134
- getEllipsoidal
  - coordinates.Gauss, 61
  - coordinates.GeographicCoordinate, 77
  - coordinates.GeographicCoordinateInterface, 87
  - coordinates.Soldner, 116
- getEtabr
  - params.LatParm, 96
- getFileinput
  - params.ControlParms, 24
- getFileoutput
  - params.ControlParms, 24
- getFirstEccentricity
  - params.EllipsoidParms, 47
- getFlattening
  - params.EllipsoidParms, 47
- getFormula
  - params.SpatialSimilarityTransformParm, 127
- getFromawert
  - params.ControlParms, 25
- getFrombwert
  - params.ControlParms, 25
- getFromdatum
  - params.ControlParms, 25
- getFromprojection
  - params.ControlParms, 25
- getGamma
  - params.SpatialSimilarityTransformParm, 127
- getGeographicCoordinateInterface
  - coordinates.GetAppropriateCoordinate, 94
- getGK\_refmer
  - params.EllipsoidParms, 48

- getHeight
  - coordinates.Gauss, 61
  - coordinates.GeographicCoordinate, 77
  - coordinates.GeographicCoordinateInterface, 87
  - coordinates.Soldner, 116
- getHoch
  - coordinates.GaussKrueger, 68
- getId
  - params.EllipsoidParms, 48
- getIla0
  - coordinates.GaussKrueger, 68
- getInstance
  - coordinates.GeographicCoordinate, 77, 78
  - coordinates.GeographicCoordinateInterface, 87, 88
  - datumstrategy.MolodenskiiTransformationAbridged, 104
  - datumstrategy.MolodenskiiTransformationStandard, 108
  - params.ControlParms, 25
- getKindofhoehe
  - params.ControlParms, 25
- getKindoftrafo
  - params.ControlParms, 25
- getL0
  - coordinates.Soldner, 117
- getLatitude
  - coordinates.GeographicCoordinate, 78
  - coordinates.GeographicCoordinateInterface, 88
- getLongitude
  - coordinates.GeographicCoordinate, 78
  - coordinates.GeographicCoordinateInterface, 88
- getM
  - params.EllipsoidParms, 48
- getMassstab
  - params.ControlParms, 26
- getN
  - params.EllipsoidParms, 48
- getNorth
  - coordinates.UTM, 134
- getNorthhem
  - params.ControlParms, 26
- getOrdinate
  - coordinates.Gauss, 61
  - coordinates.Soldner, 117
- getPoint\_number
  - coordinates.GeographicCoordinate, 79
  - coordinates.GeographicCoordinateInterface, 88
  - params.ControlParms, 26
- getPointid
  - coordinates.Coordinate, 42
  - params.ControlParms, 26
- getQupkt
  - params.ControlParms, 26
- getQutausch
  - params.ControlParms, 26
- getRadg
  - params.LatParm, 97
- getRadm
  - params.LatParm, 97
- getRadn
  - params.LatParm, 97
- getRechts
  - coordinates.GaussKrueger, 68
- getRx2
  - params.ControlParms, 27
- getRz1
  - params.ControlParms, 27
- getRz3
  - params.ControlParms, 27
- getScale
  - coordinates.Gauss, 61
  - params.SpatialSimilarityTransformParm, 127
- getSecondEccentricity
  - params.EllipsoidParms, 48
- getSemiMajorAxis
  - params.EllipsoidParms, 48
- getSemiMinorAxis
  - params.EllipsoidParms, 48
- getSoldner\_b0
  - params.EllipsoidParms, 48
- getSoldner\_I0
  - params.EllipsoidParms, 49
- getSourceCoordinateX
  - params.ControlParms, 27
- getSourceCoordinateY
  - params.ControlParms, 27
- getSourceCoordinateZ
  - params.ControlParms, 27
- getStrategy
  - datumstrategy.GetAppropriateTransformationAlgorithm, 94
- getTbr
  - params.LatParm, 97
- getTbr2
  - params.LatParm, 97
- getToa
  - params.EllipsoidParms, 49
- getToawert
  - params.ControlParms, 27
- getTob
  - params.EllipsoidParms, 49
- getTobwert
  - params.ControlParms, 27
- getToc
  - params.EllipsoidParms, 49
- getTodatum
  - params.ControlParms, 28
- getToe2
  - params.EllipsoidParms, 49
- getToes2
  - params.EllipsoidParms, 49
- getTof
  - params.EllipsoidParms, 49
- getToid
  - params.EllipsoidParms, 49

- getTopprojection
  - params.ControlParms, 28
- getTransx
  - params.MolodenskyParm, 111
  - params.SpatialSimilarityTransformParm, 127
- getTransy
  - params.MolodenskyParm, 111
  - params.SpatialSimilarityTransformParm, 128
- getTransz
  - params.MolodenskyParm, 112
  - params.SpatialSimilarityTransformParm, 128
- getVbr
  - params.LatParm, 97
- getVbr2
  - params.LatParm, 97
- getWx
  - params.ControlParms, 28
- getWy
  - params.ControlParms, 28
- getWz
  - params.ControlParms, 28
- getX
  - coordinates.XYZCoordinate, 141
- getY
  - coordinates.XYZCoordinate, 141
- getZ
  - coordinates.XYZCoordinate, 141
- getZihoehe
  - params.ControlParms, 28
- getZipkt
  - params.ControlParms, 28
- getZitausch
  - params.ControlParms, 28
- getZone
  - coordinates.UTM, 135
- GK\_refmer
  - params.EllipsoidParms, 53
- height
  - coordinates.Gauss, 64
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 91
  - coordinates.Soldner, 119
- hoch
  - coordinates.GaussKrueger, 70
- id
  - params.EllipsoidParms, 53
- init
  - coordinates.UTM, 135
  - coordinates.XYZCoordinate, 141
- instance
  - params.ControlParms, 37
- isChangedatum
  - params.ControlParms, 29
- isNorthhem
  - coordinates.UTM, 135
- kindofhoehe
  - params.ControlParms, 37
- kindoftrafo
  - params.ControlParms, 37
- l0
  - coordinates.Soldner, 120
- latitude
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 91
- LatParm
  - params.LatParm, 96
- longitude
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 91
- M
  - params.EllipsoidParms, 53
- main
  - Transformation, 130
- massstab
  - params.ControlParms, 37
- meridianLength
  - coordinates.Gauss, 62
- MolodenskiiTransformationAbridged
  - datumstrategy.MolodenskiiTransformationAbridged, 104
- MolodenskiiTransformationStandard
  - datumstrategy.MolodenskiiTransformationStandard, 108
- MolodenskyParm
  - params.MolodenskyParm, 111
- molodenskytrafo
  - coordinates.GeographicCoordinateInterface, 89
- myInstance
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 91
  - datumstrategy.MolodenskiiTransformationAbridged, 105
  - datumstrategy.MolodenskiiTransformationStandard, 108
- N
  - params.EllipsoidParms, 53
- north
  - coordinates.UTM, 136
- northhem
  - coordinates.UTM, 136
  - params.ControlParms, 38
- ordinate
  - coordinates.Gauss, 64
  - coordinates.Soldner, 120
- params, 14
  - params.ControlParms, 20
  - ausgabety, 35
  - changedatum, 35
  - ControlParms, 23
  - da, 35

destinationCoordinateX, 35  
destinationCoordinateY, 36  
destinationCoordinateZ, 36  
df, 36  
dx, 36  
dy, 36  
dz, 36  
fileinput, 36  
fileoutput, 36  
fromawert, 37  
frombwert, 37  
fromdatum, 37  
fromprojection, 37  
getAusgabety, 23  
getDa, 23  
getDestinationCoordinateX, 23  
getDestinationCoordinateY, 24  
getDestinationCoordinateZ, 24  
getDf, 24  
getDx, 24  
getDy, 24  
getDz, 24  
getFileinput, 24  
getFileoutput, 24  
getFromawert, 25  
getFrombwert, 25  
getFromdatum, 25  
getFromprojection, 25  
getInstance, 25  
getKindofhoehe, 25  
getKindoftrafo, 25  
getMassstab, 26  
getNorthhem, 26  
getPoint\_number, 26  
getPointid, 26  
getQupkt, 26  
getQutausch, 26  
getRx2, 27  
getRz1, 27  
getRz3, 27  
getSourceCoordinateX, 27  
getSourceCoordinateY, 27  
getSourceCoordinateZ, 27  
getToawert, 27  
getTobwert, 27  
getTodatum, 28  
getTopprojection, 28  
getWx, 28  
getWy, 28  
getWz, 28  
getZihoehe, 28  
getZipkt, 28  
getZitausch, 28  
instance, 37  
isChangedatum, 29  
kindofhoehe, 37  
kindoftrafo, 37  
massstab, 37  
northhem, 38  
point\_number, 38  
pointid, 38  
print, 29  
qupkt, 38  
qutausch, 38  
Rx2, 38  
Rz1, 38  
Rz3, 38  
setAusgabety, 29  
setChangedatum, 29  
setDa, 29  
setDestinationCoordinateX, 29  
setDestinationCoordinateY, 30  
setDestinationCoordinateZ, 30  
setDf, 30  
setDx, 30  
setDy, 30  
setDz, 30  
setFileinput, 30  
setFileoutput, 31  
setFromawert, 31  
setFrombwert, 31  
setFromdatum, 31  
setFromprojection, 31  
setKindofhoehe, 31  
setKindoftrafo, 31  
setMassstab, 32  
setNorthhem, 32  
setPoint\_number, 32  
setPointid, 32  
setQupkt, 32  
setQutausch, 33  
setRx2, 33  
setRz1, 33  
setRz3, 33  
setSourceCoordinateX, 33  
setSourceCoordinateY, 33  
setSourceCoordinateZ, 33  
setToawert, 34  
setTobwert, 34  
setTodatum, 34  
setTopprojection, 34  
setWx, 34  
setWy, 34  
setWz, 34  
setZihoehe, 35  
setZipkt, 35  
setZitausch, 35  
sourceCoordinateX, 39  
sourceCoordinateY, 39  
sourceCoordinateZ, 39  
toawert, 39  
tobwert, 39  
todatum, 39  
topprojection, 39  
wx, 39  
wy, 40

- wz, [40](#)
- zihoehe, [40](#)
- zipkt, [40](#)
- zitausch, [40](#)
- params.EllipsoidParms, [45](#)
  - c, [53](#)
  - calculatelongitudeParms, [47](#)
  - EllipsoidParms, [47](#)
  - firstEccentricity, [53](#)
  - flattening, [53](#)
  - getC, [47](#)
  - getFirstEccentricity, [47](#)
  - getFlattening, [47](#)
  - getGK\_refmer, [48](#)
  - getId, [48](#)
  - getM, [48](#)
  - getN, [48](#)
  - getSecondEccentricity, [48](#)
  - getSemiMajorAxis, [48](#)
  - getSemiMinorAxis, [48](#)
  - getSoldner\_b0, [48](#)
  - getSoldner\_I0, [49](#)
  - getToa, [49](#)
  - getTob, [49](#)
  - getToc, [49](#)
  - getToe2, [49](#)
  - getToes2, [49](#)
  - getTof, [49](#)
  - getToid, [49](#)
  - GK\_refmer, [53](#)
  - id, [53](#)
  - M, [53](#)
  - N, [53](#)
  - print, [50](#)
  - secondEccentricity, [53](#)
  - semiMajorAxis, [54](#)
  - semiMinorAxis, [54](#)
  - setC, [50](#)
  - setFirstEccentricity, [50](#)
  - setFlattening, [50](#)
  - setGK\_refmer, [50](#)
  - setId, [50](#)
  - setM, [50](#)
  - setN, [51](#)
  - setSecondEccentricity, [51](#)
  - setSemiMajorAxis, [51](#)
  - setSemiMinorAxis, [51](#)
  - setSoldner\_b0, [51](#)
  - setSoldner\_I0, [51](#)
  - setToa, [51](#)
  - setTob, [52](#)
  - setToc, [52](#)
  - setToe2, [52](#)
  - setToes2, [52](#)
  - setTof, [52](#)
  - setToid, [52](#)
  - Soldner\_b0, [54](#)
  - Soldner\_I0, [54](#)
- toa, [54](#)
- tob, [54](#)
- toc, [54](#)
- toe2, [54](#)
- toes2, [55](#)
- tof, [55](#)
- toid, [55](#)
- params.LatParm, [95](#)
  - cbr, [99](#)
  - Constant, [96](#)
  - etabr, [99](#)
  - getCbr, [96](#)
  - getEtabr, [96](#)
  - getRadg, [97](#)
  - getRadm, [97](#)
  - getRadn, [97](#)
  - getTbr, [97](#)
  - getTbr2, [97](#)
  - getVbr, [97](#)
  - getVbr2, [97](#)
  - LatParm, [96](#)
  - radg, [99](#)
  - radm, [99](#)
  - radn, [99](#)
  - setCbr, [97](#)
  - setEtabr, [98](#)
  - setRadg, [98](#)
  - setRadm, [98](#)
  - setRadn, [98](#)
  - setTbr, [98](#)
  - setTbr2, [98](#)
  - setVbr, [98](#)
  - setVbr2, [99](#)
  - tbr, [99](#)
  - vbr, [100](#)
- params.MolodenskyParm, [110](#)
  - da, [113](#)
  - getDa, [111](#)
  - getDf, [111](#)
  - getTransx, [111](#)
  - getTransy, [111](#)
  - getTransz, [112](#)
  - MolodenskyParm, [111](#)
  - setDa, [112](#)
  - setDf, [112](#)
  - setTransx, [112](#)
  - setTransy, [112](#)
  - setTransz, [112](#)
  - transx, [113](#)
- params.SpatialSimilarityTransformParm, [126](#)
  - alpha, [129](#)
  - formula, [129](#)
  - getAlpha, [127](#)
  - getBeta, [127](#)
  - getFormula, [127](#)
  - getGamma, [127](#)
  - getScale, [127](#)
  - getTransx, [127](#)



- getTransy, 128
- getTransz, 128
- scale, 129
- setAlpha, 128
- setBeta, 128
- setFormula, 128
- setGamma, 128
- setScale, 128
- setTransx, 129
- setTransy, 129
- setTransz, 129
- SpatialSimilarityTransformParm, 127
- transx, 129
- params/ControlParms.java, 150
- params/EllipsoidParms.java, 150
- params/LatParm.java, 151
- params/MolodenskyParm.java, 151
- params/SpatialSimilarityTransformParm.java, 151
- point\_number
  - coordinates.GeographicCoordinate, 81
  - coordinates.GeographicCoordinateInterface, 92
  - params.ControlParms, 38
- pointid
  - coordinates.Coordinate, 43
  - params.ControlParms, 38
- print
  - coordinates.Coordinate, 42
  - coordinates.Gauss, 62
  - coordinates.GaussKrueger, 68
  - coordinates.GeographicCoordinate, 79
  - coordinates.GeographicCoordinateInterface, 89
  - coordinates.Soldner, 117
  - coordinates.UTM, 135
  - coordinates.XYZCoordinate, 142
  - params.ControlParms, 29
  - params.EllipsoidParms, 50
- qupkt
  - params.ControlParms, 38
- qutausch
  - params.ControlParms, 38
- radg
  - params.LatParm, 99
- radm
  - params.LatParm, 99
- radn
  - params.LatParm, 99
- rechts
  - coordinates.GaussKrueger, 70
- resetState
  - coordinates.GeographicCoordinate, 79
  - coordinates.GeographicCoordinateInterface, 89
- RHO
  - test.coordinates.GaussKruegerTest, 72
  - test.coordinates.GaussTest, 73
  - test.datumstrategy.MolodenskiiTransformationAbridgedTest, 106
- test.datumstrategy.MolodenskiiTransformationStandardTest, 110
- test.datumstrategy.SpatialSimilarityTransformationInfinTest, 123
- test.datumstrategy.SpatialSimilarityTransformationTrigTest, 125
- Rx2
  - params.ControlParms, 38
- Rz1
  - params.ControlParms, 38
- Rz3
  - params.ControlParms, 38
- scale
  - coordinates.Gauss, 64
  - params.SpatialSimilarityTransformParm, 129
- secondEccentricity
  - params.EllipsoidParms, 53
- semiMajorAxis
  - params.EllipsoidParms, 54
- semiMinorAxis
  - params.EllipsoidParms, 54
- setAbszisse
  - coordinates.Gauss, 62
  - coordinates.Soldner, 117
- setAlpha
  - params.SpatialSimilarityTransformParm, 128
- setAusgabotyp
  - params.ControlParms, 29
- setB0
  - coordinates.Soldner, 118
- setBeta
  - params.SpatialSimilarityTransformParm, 128
- setC
  - params.EllipsoidParms, 50
- setCbr
  - params.LatParm, 97
- setChangedatum
  - params.ControlParms, 29
- setDa
  - params.ControlParms, 29
  - params.MolodenskyParm, 112
- setDestinationCoordinateX
  - params.ControlParms, 29
- setDestinationCoordinateY
  - params.ControlParms, 30
- setDestinationCoordinateZ
  - params.ControlParms, 30
- setDf
  - params.ControlParms, 30
  - params.MolodenskyParm, 112
- setDx
  - params.ControlParms, 30
- setDy
  - params.ControlParms, 30
- setDz
  - params.ControlParms, 30
- setEast
  - coordinates.UTM, 135

- setEllipsoidal
  - coordinates.Gauss, [62](#)
  - coordinates.GeographicCoordinate, [79](#)
  - coordinates.GeographicCoordinateInterface, [89](#)
  - coordinates.Soldner, [118](#)
- setEtabr
  - params.LatParm, [98](#)
- setFileinput
  - params.ControlParms, [30](#)
- setFileoutput
  - params.ControlParms, [31](#)
- setFirstEccentricity
  - params.EllipsoidParms, [50](#)
- setFlattening
  - params.EllipsoidParms, [50](#)
- setFormula
  - params.SpatialSimilarityTransformParm, [128](#)
- setFromawert
  - params.ControlParms, [31](#)
- setFrombwert
  - params.ControlParms, [31](#)
- setFromdatum
  - params.ControlParms, [31](#)
- setFromprojection
  - params.ControlParms, [31](#)
- setGamma
  - params.SpatialSimilarityTransformParm, [128](#)
- setGK\_refmer
  - params.EllipsoidParms, [50](#)
- setHeight
  - coordinates.Gauss, [63](#)
  - coordinates.GeographicCoordinate, [80](#)
  - coordinates.GeographicCoordinateInterface, [90](#)
  - coordinates.Soldner, [118](#)
- setHoch
  - coordinates.GaussKrueger, [69](#)
- setId
  - params.EllipsoidParms, [50](#)
- setIlao
  - coordinates.GaussKrueger, [70](#)
- setKindofhoehe
  - params.ControlParms, [31](#)
- setKindoftrafo
  - params.ControlParms, [31](#)
- setLO
  - coordinates.Soldner, [118](#)
- setLatitude
  - coordinates.GeographicCoordinate, [80](#)
  - coordinates.GeographicCoordinateInterface, [90](#)
- setLongitude
  - coordinates.GeographicCoordinate, [80](#)
  - coordinates.GeographicCoordinateInterface, [90](#)
- setM
  - params.EllipsoidParms, [50](#)
- setMassstab
  - params.ControlParms, [32](#)
- setN
  - params.EllipsoidParms, [51](#)
- setNorth
  - coordinates.UTM, [135](#)
- setNorthhem
  - coordinates.UTM, [135](#)
  - params.ControlParms, [32](#)
- setOrdinate
  - coordinates.Gauss, [63](#)
  - coordinates.Soldner, [119](#)
- setPoint\_number
  - coordinates.GeographicCoordinate, [80](#)
  - coordinates.GeographicCoordinateInterface, [91](#)
  - params.ControlParms, [32](#)
- setPointid
  - coordinates.Coordinate, [43](#)
  - params.ControlParms, [32](#)
- setQupkt
  - params.ControlParms, [32](#)
- setQutausch
  - params.ControlParms, [33](#)
- setRadg
  - params.LatParm, [98](#)
- setRadm
  - params.LatParm, [98](#)
- setRadn
  - params.LatParm, [98](#)
- setRechts
  - coordinates.GaussKrueger, [70](#)
- setRx2
  - params.ControlParms, [33](#)
- setRz1
  - params.ControlParms, [33](#)
- setRz3
  - params.ControlParms, [33](#)
- setScale
  - coordinates.Gauss, [63](#)
  - params.SpatialSimilarityTransformParm, [128](#)
- setSecondEccentricity
  - params.EllipsoidParms, [51](#)
- setSemiMajorAxis
  - params.EllipsoidParms, [51](#)
- setSemiMinorAxis
  - params.EllipsoidParms, [51](#)
- setSoldner\_b0
  - params.EllipsoidParms, [51](#)
- setSoldner\_I0
  - params.EllipsoidParms, [51](#)
- setSourceCoordinateX
  - params.ControlParms, [33](#)
- setSourceCoordinateY
  - params.ControlParms, [33](#)
- setSourceCoordinateZ
  - params.ControlParms, [33](#)
- setTbr
  - params.LatParm, [98](#)
- setTbr2
  - params.LatParm, [98](#)
- setToa
  - params.EllipsoidParms, [51](#)

- setToawert
  - params.ControlParms, 34
- setTob
  - params.EllipsoidParms, 52
- setTobwert
  - params.ControlParms, 34
- setToc
  - params.EllipsoidParms, 52
- setTodatum
  - params.ControlParms, 34
- setToe2
  - params.EllipsoidParms, 52
- setToes2
  - params.EllipsoidParms, 52
- setTof
  - params.EllipsoidParms, 52
- setToid
  - params.EllipsoidParms, 52
- setToprojection
  - params.ControlParms, 34
- setTransx
  - params.MolodenskyParm, 112
  - params.SpatialSimilarityTransformParm, 129
- setTransy
  - params.MolodenskyParm, 112
  - params.SpatialSimilarityTransformParm, 129
- setTransz
  - params.MolodenskyParm, 112
  - params.SpatialSimilarityTransformParm, 129
- setVbr
  - params.LatParm, 98
- setVbr2
  - params.LatParm, 99
- setWx
  - params.ControlParms, 34
- setWy
  - params.ControlParms, 34
- setWz
  - params.ControlParms, 34
- setX
  - coordinates.XYZCoordinate, 142
- setY
  - coordinates.XYZCoordinate, 142
- setZ
  - coordinates.XYZCoordinate, 142
- setZihoehe
  - params.ControlParms, 35
- setZipkt
  - params.ControlParms, 35
- setZitausch
  - params.ControlParms, 35
- setZone
  - coordinates.UTM, 136
- Soldner
  - coordinates.Soldner, 115
- Soldner\_b0
  - params.EllipsoidParms, 54
- Soldner\_I0
  - params.EllipsoidParms, 54
- sourceCoordinateX
  - params.ControlParms, 39
- sourceCoordinateY
  - params.ControlParms, 39
- sourceCoordinateZ
  - params.ControlParms, 39
- SpatialSimilarityTransformParm
  - params.SpatialSimilarityTransformParm, 127
- tbr
  - params.LatParm, 99
- test, 14
  - test.params.LatParmTest, 100
- test.coordinates, 14
  - test.coordinates.AllTests, 17
  - test.coordinates.CoordinateFactoryTest, 43
    - testGetCoordGeographic, 44
    - testGetCoordGK, 44
    - testGetCoordSoldner, 44
    - testGetCoordUTM, 44
    - testGetXYZ, 44
  - test.coordinates.GaussKruegerTest, 71
    - EPSILON\_IN\_DEGREE, 72
    - EPSILON\_IN\_METER, 72
    - getAsGaussKrueger, 71
    - getAsGeographic, 72
    - RHO, 72
  - test.coordinates.GaussTest, 72
    - EPSILON\_IN\_METER, 73
    - RHO, 73
    - testMeridianLength, 73
  - test.coordinates.GeographicCoordinateTest, 92
    - EPSILONFORDEGREE, 92
    - testFromGeographicInterface, 92
    - testGetAsGeographicInterface, 92
  - test.coordinates.UTMTest, 137
    - testGetAsGeographicInterface, 137
    - testGetAsUTM, 137
  - test.coordinates.XYZCoordinateTest, 143
    - testGetAsGeographicInterface, 143
    - testGetAsTargetCoordinate, 143
- test.datumstrategy, 15
  - test.datumstrategy.GetAppropriateTransformationAlgorithmTest, 95
    - testSpatialSimilarityTransformationInfin, 95
  - test.datumstrategy.MockStrategyTest, 101
    - EPSILON\_IN\_METER, 103
    - testTransformGaussKruegerStrip3ToStrip4, 102
    - testTransformGaussKruegerStrip4ToStrip3, 102
  - test.datumstrategy.MolodenskiiTransformationAbridgedTest, 105
    - EPSILON\_meter, 106
    - EPSILON\_rad, 106
    - RHO, 106
    - testMolodenskiiTransformationAbridged, 106
  - test.datumstrategy.MolodenskiiTransformationStandardTest, 109
    - EPSILON\_IN\_METER, 109

- EPSILON\_IN\_RAD, [110](#)
- RHO, [110](#)
- testTransform, [109](#)
- test.datumstrategy.SpatialSimilarityTransformationInfinTest, [122](#)
- EPSILON\_IN\_METER, [123](#)
- EPSILON\_IN\_RAD, [123](#)
- RHO, [123](#)
- testTransform, [123](#)
- test.datumstrategy.SpatialSimilarityTransformationTrigTest, [124](#)
- EPSILON\_IN\_METER, [125](#)
- EPSILON\_IN\_RAD, [125](#)
- RHO, [125](#)
- testTransform, [125](#)
- test.params, [15](#)
- test.params.LatParmTest, [100](#)
- test, [100](#)
- test/coordinates/AllTests.java, [151](#)
- test/coordinates/CoordinateFactoryTest.java, [152](#)
- test/coordinates/GaussKruegerTest.java, [152](#)
- test/coordinates/GaussTest.java, [152](#)
- test/coordinates/GeographicCoordinateTest.java, [152](#)
- test/coordinates/UTMTest.java, [153](#)
- test/coordinates/XYZCoordinateTest.java, [153](#)
- test/datumstrategy/GetAppropriateTransformationAlgorithmTest.java, [153](#)
- test/datumstrategy/MockStrategyTest.java, [153](#)
- test/datumstrategy/MolodenskiiTransformationAbridgedTest.java, [154](#)
- test/datumstrategy/MolodenskiiTransformationStandardTest.java, [154](#)
- test/datumstrategy/SpatialSimilarityTransformationInfinTest.java, [154](#)
- test/datumstrategy/SpatialSimilarityTransformationTrigTest.java, [154](#)
- test/params/LatParmTest.java, [155](#)
- testFromGeographicInterface
  - test.coordinates.GeographicCoordinateTest, [92](#)
- testGetAsGeographicInterface
  - test.coordinates.GeographicCoordinateTest, [92](#)
  - test.coordinates.UTMTest, [137](#)
  - test.coordinates.XYZCoordinateTest, [143](#)
- testGetAsTargetCoordinate
  - test.coordinates.XYZCoordinateTest, [143](#)
- testGetAsUTM
  - test.coordinates.UTMTest, [137](#)
- testGetCoordGeographic
  - test.coordinates.CoordinateFactoryTest, [44](#)
- testGetCoordGK
  - test.coordinates.CoordinateFactoryTest, [44](#)
- testGetCoordSoldner
  - test.coordinates.CoordinateFactoryTest, [44](#)
- testGetCoordUTM
  - test.coordinates.CoordinateFactoryTest, [44](#)
- testGetXYZ
  - test.coordinates.CoordinateFactoryTest, [44](#)
- testMeridianLength
  - test.coordinates.GaussTest, [73](#)
- testMolodenskiiTransformationAbridged
  - test.datumstrategy.MolodenskiiTransformationAbridgedTest, [106](#)
- testSpatialSimilarityTransformationInfin
  - test.datumstrategy.GetAppropriateTransformationAlgorithmTest, [95](#)
- testTransform
  - test.datumstrategy.MolodenskiiTransformationStandardTest, [109](#)
  - test.datumstrategy.SpatialSimilarityTransformationInfinTest, [123](#)
  - test.datumstrategy.SpatialSimilarityTransformationTrigTest, [125](#)
- testTransformGaussKruegerStrip3ToStrip4
  - test.datumstrategy.MockStrategyTest, [102](#)
- testTransformGaussKruegerStrip4ToStrip3
  - test.datumstrategy.MockStrategyTest, [102](#)
- toa
  - params.EllipsoidParms, [54](#)
- toawert
  - params.ControlParms, [39](#)
- tob
  - params.EllipsoidParms, [54](#)
- tobwert
  - params.ControlParms, [39](#)
- toc
  - params.EllipsoidParms, [54](#)
- todatum
  - params.ControlParms, [39](#)
- toe2a,
  - params.EllipsoidParms, [54](#)
- toe2,
  - params.EllipsoidParms, [55](#)
- tojava,
  - params.EllipsoidParms, [55](#)
- toid
  - params.EllipsoidParms, [55](#)
- toprojection
  - params.ControlParms, [39](#)
- transform
  - datumstrategy.CardansTransformation, [18](#)
  - datumstrategy.EulersTransformation, [56](#)
  - datumstrategy.MockStrategy, [101](#)
  - datumstrategy.MolodenskiiTransformationAbridged, [104](#)
  - datumstrategy.MolodenskiiTransformationStandard, [108](#)
  - datumstrategy.SpatialSimilarityTransformationInfin, [122](#)
  - datumstrategy.SpatialSimilarityTransformationTrig, [124](#)
  - datumstrategy.TransformationStrategy, [132](#)
- Transformation, [130](#)
- main, [130](#)
- Transformation.java, [155](#)
- transx
  - params.MolodenskyParm, [113](#)

params.SpatialSimilarityTransformParm, [129](#)

## UTM

coordinates.UTM, [133](#)

## vbr

params.LatParm, [100](#)

## wx

params.ControlParms, [39](#)

## wy

params.ControlParms, [40](#)

## wz

params.ControlParms, [40](#)

## x

coordinates.XYZCoordinate, [143](#)

## XYZCoordinate

coordinates.XYZCoordinate, [139](#)

## zihoehe

params.ControlParms, [40](#)

## zipkt

params.ControlParms, [40](#)

## zitausch

params.ControlParms, [40](#)

## zone

coordinates.UTM, [136](#)