OrdinarySurfaceCoefficients

```
# _lambda: std::vector< std::vector< pair< RowMatrix<GLdouble>, RowMatrix<GLdouble> > > >

+ OrdinarySurfaceCoefficients(u_dimension: GLint, v_dimension: GLint, sigma: const std::vector<GLint>&)
+ <<const>> operator()(k: GLint, zeta: GLint, type: variable::Type, index: GLint): GLdouble
+ operator()(k: GLint, zeta: GLint, type: variable::Type, index: GLint): GLdouble&
+ <<const>> sigma(k: GLint): GLint
+ <<const>> dimension(type: variable::Type): GLint
+ <<const>> clone(): OrdinarySurfaceCoefficients*
```

BSurface3: public TensorProductSurface3

```
# _S: ECSpace[2]
# _T: SP<RealSquareMatrix>::Default[2]

+ BSurface3(u_space: const ECSpace&, v_space: const ECSpace&)
+ setInterval(type: variable::Type, alpha: GLdouble, beta: GLdouble,
               check_for_ill_conditioned_matrices: bool = false,
               expected_correct_significant_digits: GLint = 5): GLboolean
+ <<const>> blendingFunctionValues(type: variable::Type, parameter_value: GLdouble,
                                    values: RowMatrix<GLdouble>&): GLboolean
+ <<const>> calculateAllPartialDerivatives(maximum_order_of_derivatives: GLint,
                                           u: GLdouble, v: GLdouble, pd: PartialDerivatives&): GLboolean
+ <<const>> calculateDirectionalDerivatives(direction: variable::Type,
                                             maximum_order_of__derivatives: GLint,
                                             u: GLdouble, v: GLdouble v,
                                             d: DirectionalDerivatives&): GLboolean
+ <<const>> operator()(type: variable::Type, i: GLint, j: GLint, parameter_value: GLdouble): GLdouble
+ <<const>> performOrderElevation(type: variable::Type, a: GLdouble, b: GLdouble, order: GLint,
                                  check_for_ill_conditioned_matrices: bool = false,
                                  expected_correct_significant_digits: GLint = 5): BSurface3*
+ <<const>> performOrderElevation(type: variable::Type, zero: const CharacteristicPolynomial::Zero&,
                                  check_for_ill_conditioned_matrices: bool = false,
                                  expected_correct_significant_digits: GLint = 5): BSurface3*
+ <<const>> performSubdivision(type: variable::Type, gamma: GLdouble,
                               check_for_ill_conditioned_matrices: bool = false,
                               expected_correct_significant_digits: GLint = 5):
                               RowMatrix<SP<BSurface3>::Default>*
+ updateControlPointsForExactDescription(lambda: const OrdinarySurfaceCoefficients&): GLboolean
+ <<const>> clone(): BSurface3*
```