```
#ifndef MATERIAL_MODEL_1D_BAR
#define MATERIAL_MODEL_1D_BAR

#include "/src/Definitions.h"

namespace MaterialModels {

class MaterialModel1DBar {

public:

  typedef Matrix<double, 1, 1> DisplacementGradient;
  typedef Matrix<double, 1, 1> Strain;
  typedef Matrix<double, 1, 1> Stress;
  typedef Matrix<double, 1, 1> TangentMatrix;

  MaterialModel1DBar(const double youngsModulus):
    _youngsModulus(youngsModulus) {
  };


  double
  computeEnergy(const DisplacementGradient & displacementGradient) const {

    //ignoreUnusedVariable(displacementGradient);

    double energy = 0.0;

    //TODO: Evaluate the energy using the displacementGradient given

    energy = _youngsModulus * displacementGradient(0)*displacementGradient(0)/2;

    return energy;
  };

  Stress
  computeStress(const DisplacementGradient & displacementGradient) const {

    //ignoreUnusedVariable(displacementGradient);

    Stress stress = Stress::Zero();

    //TODO: Evaluate the stress using the displacementGradient given

    stress (0) = _youngsModulus * displacementGradient(0);

    return stress;
  };

  TangentMatrix
  computeTangentMatrix(const DisplacementGradient & displacementGradient) const {

    ignoreUnusedVariable(displacementGradient);

    TangentMatrix tangentMatrix = TangentMatrix::Zero();

    //TODO: Evaluate the tangent matrix using the displacementGradient given
    tangentMatrix(0) = _youngsModulus;

    return tangentMatrix;
  };

private:
  double _youngsModulus;
};


} // namespace MaterialModels
#endif //MATERIAL_MODEL_1D_BAR
```