

```

/*
*
*          #####          #####          #####          #####          ###          ###
*          ##          ##          ##          ##          ##          ##          ##          ##
*          ##          ##          ##          ##          ##          ##          ##          ##
*          ##          ##          ##          ##          ##          ##          ##          ##
*          ##          ##          ##          ##          ##          ##          ##          ##
*          #####          #####          ##          #####          ##          ##
*
*
*          OOFEM : Object Oriented Finite Element Code
*
*          Copyright (C) 1993 - 2019   Borek Patzak
*
*
*          Czech Technical University, Faculty of Civil Engineering,
*          Department of Structural Mechanics, 166 29 Prague, Czech Republic
*
*          This library is free software; you can redistribute it and/or
*          modify it under the terms of the GNU Lesser General Public
*          License as published by the Free Software Foundation; either
*          version 2.1 of the License, or (at your option) any later version.
*
*          This program is distributed in the hope that it will be useful,
*          but WITHOUT ANY WARRANTY; without even the implied warranty of
*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
*          Lesser General Public License for more details.
*
*          You should have received a copy of the GNU Lesser General Public
*          License along with this library; if not, write to the Free Software
*          Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
*/

#ifdef dofiditemh
#define dofiditemh

#include "enumitem.h"

#include <string>

namespace oofem {
#define DofIDItem_DEF \
ENUM_ITEM_WITH_VALUE(_UNDEF, 0) /* Error value */ \
ENUM_ITEM_WITH_VALUE(D_u, 1) /* u-displacement (in direction of x-axis) */ \
ENUM_ITEM_WITH_VALUE(D_v, 2) /* v-displacement (in direction of y-axis) */ \
ENUM_ITEM_WITH_VALUE(D_w, 3) /* w-displacement (in direction of z-axis) */ \
ENUM_ITEM_WITH_VALUE(R_u, 4) /* Rotation around x-axis (right hand rule assumed) */ \
ENUM_ITEM_WITH_VALUE(R_v, 5) /* Rotation around y-axis */ \
ENUM_ITEM_WITH_VALUE(R_w, 6) /* Rotation around z-axis */ \
\
ENUM_ITEM_WITH_VALUE(V_u, 7) /* u-velocity (in direction of x-axis) */ \
ENUM_ITEM_WITH_VALUE(V_v, 8) /* v-velocity (in direction of y-axis) */ \
ENUM_ITEM_WITH_VALUE(V_w, 9) /* w-velocity (in direction of z-axis) */ \
\
ENUM_ITEM_WITH_VALUE(T_f, 10) /* Temperature field */ \
ENUM_ITEM_WITH_VALUE(P_f, 11) /* Pressure field */ \

```

```

ENUM_ITEM_WITH_VALUE(G_0, 12) /* DOF for gradient formulation no. 0 */ \
ENUM_ITEM_WITH_VALUE(G_1, 13) /* DOF for gradient formulation no. 1 */ \
ENUM_ITEM_WITH_VALUE(C_1, 14) /* Mass concentration of the first constituent */ \
ENUM_ITEM_WITH_VALUE(W_u, 15) /* u-component of change in director field (in direction of x-axis) */ \
ENUM_ITEM_WITH_VALUE(W_v, 16) /* v-component of change in director field (in direction of y-axis) */ \
ENUM_ITEM_WITH_VALUE(W_w, 17) /* w-component of change in director field (in direction of z-axis) */ \
ENUM_ITEM_WITH_VALUE(Gamma, 18) /* inhomogenous thickness strain in direction of the directorfield m */ \
ENUM_ITEM_WITH_VALUE(D_u_edge_const, 19) /* Constant part of boundary u-displacement used by Trefftz element*/ \
ENUM_ITEM_WITH_VALUE(D_u_edge_lin, 20) /* Linear part of boundary u-displacement used by Trefftz element*/ \
ENUM_ITEM_WITH_VALUE(D_v_edge_const, 21) /* Constant part of boundary v-displacement used by Trefftz element*/ \
ENUM_ITEM_WITH_VALUE(D_v_edge_lin, 22) /* Linear part of boundary v-displacement used by Trefftz element*/ \
ENUM_ITEM_WITH_VALUE(Warp_PsiTheta, 23) /* Relative twist times deplanation function, used by Trwarp element*/ \
ENUM_ITEM_WITH_VALUE(Warp_Theta, 24) /* Relative twist, used by Trwarp element*/ \
\
ENUM_ITEM_WITH_VALUE(LMP_u, 25) /* Lagrange multiplier in x-direction*/ \
ENUM_ITEM_WITH_VALUE(LMP_v, 26) /* Lagrange multiplier in y-direction*/ \
ENUM_ITEM_WITH_VALUE(LMP_w, 27) /* Lagrange multiplier in z-direction*/ \
\
ENUM_ITEM_WITH_VALUE(Trac_u, 28) /* Independent traction field in x-direction*/ \
ENUM_ITEM_WITH_VALUE(Trac_v, 29) /* Independent traction field in y-direction*/ \
ENUM_ITEM_WITH_VALUE(Trac_w, 30) /* Independent traction field in z-direction*/ \
\
ENUM_ITEM_WITH_VALUE(E_xx, 31) /* Macroscopic strain component xx*/ \
ENUM_ITEM_WITH_VALUE(E_yy, 32) /* Macroscopic strain component yy*/ \
ENUM_ITEM_WITH_VALUE(E_zz, 33) /* Macroscopic strain component zz*/ \
ENUM_ITEM_WITH_VALUE(E_yz, 34) /* Macroscopic strain component yz*/ \
ENUM_ITEM_WITH_VALUE(E_zy, 35) /* Macroscopic strain component zy*/ \
ENUM_ITEM_WITH_VALUE(E_xz, 36) /* Macroscopic strain component xz*/ \
    ENUM_ITEM_WITH_VALUE(E_zx, 37) /* Macroscopic strain component zx*/ \
ENUM_ITEM_WITH_VALUE(E_xy, 38) /* Macroscopic strain component xy*/ \
    ENUM_ITEM_WITH_VALUE(E_yx, 39) /* Macroscopic strain component yx*/ \
\
ENUM_ITEM_WITH_VALUE(G_yz, 40) /* Macroscopic shear strain component xy (E_yz+E_zy)*/ \
    ENUM_ITEM_WITH_VALUE(G_xz, 41) /* Macroscopic shear strain component xz (E_xz+E_zx)*/ \
    ENUM_ITEM_WITH_VALUE(G_xy, 42) /* Macroscopic shear strain component xz (E_xz+E_zx)*/ \
\
    ENUM_ITEM_WITH_VALUE(K_xx, 43) /* Macroscopic curvature component xx*/ \
    ENUM_ITEM_WITH_VALUE(K_yy, 44) /* Macroscopic curvature component yy*/ \
ENUM_ITEM_WITH_VALUE(K_zz, 45) /* Macroscopic curvature component zz*/ \
    ENUM_ITEM_WITH_VALUE(K_yz, 46) /* Macroscopic curvature component yz*/ \
    ENUM_ITEM_WITH_VALUE(K_zy, 47) /* Macroscopic curvature component zy*/ \
    ENUM_ITEM_WITH_VALUE(K_zx, 48) /* Macroscopic curvature component zx*/ \
    ENUM_ITEM_WITH_VALUE(K_xy, 49) /* Macroscopic curvature component xy*/ \
    ENUM_ITEM_WITH_VALUE(K_yx, 50) /* Macroscopic curvature component yx*/ \
\
    ENUM_ITEM_WITH_VALUE(S_u, 51) /* Macroscopic reinforcement slip field in x-direction */ \
    ENUM_ITEM_WITH_VALUE(S_v, 52) /* Macroscopic reinforcement slip field in y-direction */ \
    ENUM_ITEM_WITH_VALUE(S_w, 53) /* Macroscopic reinforcement slip field in z-direction */ \
\
    ENUM_ITEM_WITH_VALUE(VF, 54) /* Volume fraction */ \
\
/**
 * Type representing particular dof type. Values of this type describe the physical meaning of
 * available DOFs.
 * @note{The implementation of Node::computeGNTransformation rely on D_u, D_v and D_w (R_u, R_v, R_w) order.
 * Do not change their order and do not insert any values between these values.}
 */

```

```
enum DofIDItem {
    DofIDItem_DEF
    MaxDofID = 500
};

#undef ENUM_ITEM
#undef ENUM_ITEM_WITH_VALUE
#undef enumitem_h

std :: string __DofIDItemToString(DofIDItem _value);
} // end namespace oofem
#endif // dofiditem_h
```