Subject: Re: again... use if deleted function :?
Posted by idkfa46 on Sat, 27 Jan 2018 13:25:46 GMT
View Forum Message <> Reply to Message

Hi Klugier,
thank you for continue supporting me.

here below more details:

TabellaCarichi.cpp (rows 192 and 195):

```
void TabellaCarichi::AcceptedGrid()
{
 SLU.Clear();
 SLE.Clear();
 qtf_loads.Clear();

 if(!slu)
 {
  TabellaCarichiQtf();

  // Combinatore di carichi
  Combiner combiner;
  combiner.LoadGrid(Soll_CaratGrid);
  // Slu
  combiner.CombineSLU();
  SLU = combiner.GetSluVector();          <<<< ROW 192
  // Sle
  combiner.CombineSLE();
  SLE = combiner.GetSleVector();          <<<< ROW 195
 }
 else
 {
  for(int r=0; r< Soll_SluGrid.GetCount(); r++)
   SLU.Add(Soll_SluGrid.Get(r, TAG_DurataCARICO)) << "" << "" << Soll_SluGrid.Get(r,
TAG_SOLLECITAZIONI);

  // Scrivi tabella dei carichi SLU
  TabellaCarichiQtf();
 }
}
```

combiner.cpp:

```
void Combiner::LoadGrid(GridCtrl &grid)
{
```

```
 G1.Clear(); G2.Clear(); Q.Clear();

 for(int i=0; i < grid.GetCount(); i++)
 {
  String cdc    = grid.Get(i, TAG_CondCARICO);
  String durata   = grid.Get(i, TAG_DurataCARICO);
  String cat    = grid.Get(i, TAG_CatCARICO);
  double sollecitaz  = grid.Get(i, TAG_SOLLECITAZIONI);
  int key    = DurConvToInt(durata);

  if(cat == TAG_G1)
  {
   G1.Add(key, _G(cdc, durata, cat, sollecitaz));
  }
  else if(cat == TAG_G2)
  {
   G2.Add(key, _G(cdc, durata, cat, sollecitaz));
  }
  else
  {
   Impostazioni &s = globalImpostazioni();
   double psi0j = s.GetPsi0j(cat);
   double psi1j = s.GetPsi1j(cat);
   double psi2j = s.GetPsi2j(cat);

   Q.Add (key, _Q(cdc, durata, cat, sollecitaz, psi0j, psi1j, psi2j));
  }
 }
 ::SortDurata(G1);          <<<< call to template<class T> (combiner.h)
 ::SortDurata(G2);          <<<< call to template<class T> (combiner.h)
 ::SortDurata(Q);          <<<< call to template<class T> (combiner.h)

}


combiner.h:


struct _G : public Moveable<_G> {
    String cdc;
    String durata;
    String categoria;
    double sollecitazione;

    _G(String s1, String s2, String s3, double d) : cdc(s1), durata(s2), categoria(s3),
sollecitazione(d) {}
};
```

```
struct _Q : public Moveable<_Q> {
    String cdc;
    String durata;
    String categoria;
    double sollecitazione;
    double Psi0j;
    double Psi1j;
    double Psi2j;

    _Q(String s1, String s2, String s3, double d1, double d2, double d3, double d4): cdc(s1),
durata(s2), categoria(s3), sollecitazione(d1), Psi0j(d2), Psi1j(d3), Psi2j(d4) {}
};

struct _Output : public Moveable<_Output> {
    String durata;
    String txt;
    double tot;

    _Output(String s1, String s2, double d): durata(s1), txt(s2), tot(d) {}
};


class Combiner {
 // variabili - forse da eliminare
 String g1_txt, g2_txt, q_txt;
 double g1_tot, g2_tot, q_tot;

 // vettore dei carichi
 VectorMap<int, _G> G1;    // Carichi permanenti
 VectorMap<int, _G> G2;    // Carichi permanenti non strut
 VectorMap<int, _Q> Q;    // Carichi variabili

 // matrici carichi variabili
 VectorMap<int, Vector<double> > loads;
 VectorMap<int, Vector<double> > coefs;
 Vector<int> pos;

 // vettore dei risultati
 VectorMap<String, Vector<Value> > SLU;
 VectorMap<String, Vector<Value> > SLE;

 // calcolo combinazioni
 bool Combine(int i, int sl);  // i=durata - sl-> 0=SLU, 1=SLErare, 2= SLEfreq, 3=SLEqperm
 void calcola_slu(int i);   // i= durata del carico
 void calcola_sle_rara();
 void calcola_sle_freq();
 void calcola_sle_qperm();
```

```
 //
 int DurConvToInt(String s);
 String DurConvToStrin(int i);
 void PlusCdC(String &s){if(!s.IsEmpty()) s << "+";}
 String StringaCdC(double gammaG1, double gammaG2, double gammaQ);
 void clean(){ g1_txt.Clear(); g2_txt.Clear(); q_txt.Clear(); g1_tot = 0; g2_tot = 0; q_tot = 0; }

public:
 // carica grid - tradizionale o multicolonna
 void LoadGrid(GridCtrl &grid);
 void LoadGrid(GridCtrl &grid, Id col);

 // attiva combinazione
 void CombineSLU();   // Stato limite ultimo
 void CombineSLE();   // Stato limite di esercizio

 // vettore dei risultati
 VectorMap<String, Vector<Value> > &GetSluVector() { return SLU; }        <<<< pick(SLU) ???
 VectorMap<String, Vector<Value> > &GetSleVector() { return SLE; }        <<<< pick(SLE) ???

 double GetSluLoads(String durata);
 String GetSluTxt(String durata);
 double GetSleLoads(String combinazione);
 String GetSleTxt(String combinazione);

 // Tabelle Qtf
 String qtf_SLU();
 String qtf_SLE();
 String qtf_SLErare();
 String qtf_SLEfreq();
 String qtf_SLEqperm();

 typedef Combiner CLASSNAME;
 Combiner(){}

 Combiner(Combiner&&) = default;
 Combiner& operator=(Combiner&&) = default;
};

template<class T> void SortDurata(VectorMap<int, T> &Load)
{
 //Riordina VectorMap in ordine crescente - durata di carico
 Vector<int> keys = Load.PickKeys();
 Vector<T> values = Load.GetValues();
 IndexSort(keys, values);
 Load.Clear();
 Load = VectorMap<int, T>(keys, values);
}
```

```
template<class T> void CheckDurata(bool &check, VectorMap<int, T> &X, int n, int i)
{
 if(X.GetKey(n) == i)
  check = true;
}
```

Compiler error:

```
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\TabellaCarichi.cpp (192): error: use of deleted function 'Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >& Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >::operator=(const Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >&)'
 (): SLU = combiner.GetSluVector( ) ;
C:\upp/uppsrc/Core/Map.h (193): note: 'Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >& Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >::operator=(const Upp::VectorMap<Upp::String, Upp::Vec
tor<Upp::Value> >&)' is implicitly deleted because the default definition would be ill-formed:
 (): class  V ectorMap  : public MoveableAndDeepCopyOption<VectorMap<K, T>>,
C:\upp/uppsrc/Core/Map.h (193): error: use of deleted function 'Upp::AMap<Upp::String, Upp::Vector<Upp::Value>, Upp::Vector<Upp::Vector<Upp::Value> > >& Upp::AMap<Upp::String, Upp::Vector<Upp::Value>, U
pp::Vector<Upp::Vector<Upp::Value> > >::operator=(const Upp::AMap<Upp::String, Upp::Vector<Upp::Value>, Upp::Vector<Upp::Vector<Upp::Value> > >&)'
C:\upp/uppsrc/Core/Map.h (33): error: use of deleted function 'Upp::Index<Upp::String>& Upp::Index<Upp::String>::operator=(const Upp::Index<Upp::String>&)'
C:\upp/uppsrc/Core/Map.h (33): error: use of deleted function 'constexpr Upp::Vector<Upp::Vector<Upp::Value> >& Upp::Vector<Upp::Vector<Upp::Value> >::operator=(const Upp::Vector<Upp::Vector<Upp::Value>
 >&)'
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\TabellaCarichi.cpp (195): error: use of deleted function 'Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >& Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >::operator=(const Upp::VectorMap<Upp::String, Upp::Vector<Upp::Value> >&)'
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\Combiner.h (113): error: use of deleted function 'constexpr Upp::Vector<_G>::Vector(const Upp::Vector<_G>&)'
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\Combiner.h (116): error: cannot bind rvalue reference of type 'Upp::Vector<int>&&' to lvalue of type 'Upp::Vector<int>'
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\Combiner.h (113): error: use of deleted function 'constexpr Upp::Vector<_Q>::Vector(const Upp::Vector<_Q>&)'
C:\Users\Matteo\Dropbox\2_Sviluppo++\Workspace_upp\LibCombiner\Combiner.h (116): error: cannot bind rvalue reference of type 'Upp::Vector<int>&&' to lvalue of type 'Upp::Vector<int>'
```

Best regards,
Matteo