| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [TimeCalculators/include](http://docs.google.com/index.html) - ShotTimeCalculator.h (source / [functions](http://docs.google.com/ShotTimeCalculator.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 6 | 6 | 100.0 % | | Date: | 2012-03-23 |  | Functions: | 0 | 0 | - | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 2 | 4 | 50.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef NFT\_SHOTTIMECALCULATOR\_H  2 : : #define NFT\_SHOTTIMECALCULATOR\_H  3 : :   4 : :   5 : : #include <stdint.h>  6 : : #include "ShortTimeInfo.h"  7 : : #include "Position.h"  8 : : //#include ""  9 : :   10 : : namespace nft { class SimplePattern; }   11 : : namespace nft { class DoseTable; }   12 : : namespace nft { class KBRTable; }   13 : : namespace nft { class DOCTable; }   14 : :   15 : : namespace nft {  16 : :   17 : :   18 : : class ShotTimeCalculator {  19 : : public:  20 : : enum ProxMode {  21 : : eProxOff = 0,  22 : : eProxOn,  23 : : eProxBdr  24 : :   25 : : };  26 : :   27 : : private:  28 : : //Implementation function that returns the Dose value for this pattern  29 : : uint16\_t  30 : : calculateDoseValue(SimplePattern & pattern, const PatternPosition &inPosition);   31 : :   32 : : inline double  33 : : translatePositionToDoseMap(int32\_t patternPos,   34 : : int64\_t subFieldPos,   35 : : double doseMapPos,   36 : : double meshSize);  37 : :   38 : : public:  39 : : void  40 : : precalculate(ProxMode proxMode,   41 : : double baseDose,   42 : : double currDensity,   43 : : double stdEta) throw();  44 : :   45 : : void  46 : : calculateShotTime( SimplePattern & pattern,   47 : : const PatternPosition &inPosition,   48 : : const int16\_t dxpValue,   49 : : const int16\_t sflNo,  50 : : uint32\_t inAI) throw(DPL::Error \*);  51 : :   52 : : ShotTimeCalculator( ProxMode inMode,  53 : : const DoseTable \*inDoseTable,  54 : : const KBRTable \*inKBRTable,  55 : : const DOCTable \*inDOCTable);  56 : :   57 : : uint32\_t   58 : : calculateShotTimeRepetition(const SimplePattern \*pattern,  59 : : uint32\_t repetition,   60 : : uint32\_t inSFLNo) throw(DPL::Error \*);  61 : : ~ShotTimeCalculator();  62 : :   63 : : ShotTimeCalculator(const ShotTimeCalculator & source);  64 : :   65 : : ShotTimeCalculator &  66 : : operator =(const ShotTimeCalculator & source);  67 : :   68 : : static inline double  69 : : getBlkUnit();  70 : :   71 : : static void  72 : 1094 : setMIN\_TP(uint16\_t minShotTime) {  73 : 1094 : MIN\_TP = minShotTime;  74 : : }  75 : :   76 : : static void  77 : 1094 : setADJ\_TP(uint16\_t normShotTime) {  78 : 1094 : ADJ\_TP = normShotTime;  79 : : }  80 : :   81 : : /\* inline void  82 : : setIsRatioEnabled(int16\_t sDoseRatioSkip) {  83 : : mIsRatioEnabled = (sDoseRatioSkip) ? 0 : 1 ;  84 : : };  85 : : \*/  86 : : private:  87 : : static const uint32\_t C\_SHOT\_TIME;  88 : : static const double BLK\_UNIT;  89 : : static uint16\_t MIN\_TP;  90 : : static uint16\_t ADJ\_TP;  91 : :   92 : : // static uint16\_t mIsRatioEnabled;  93 : :   94 : : double mPreCalcTime;  95 : : ProxMode mProxMode;   96 : : const DoseTable \*mDoseTable;  97 : : const KBRTable \*mKBRTable;  98 : : const DOCTable \*mDOCTable;  99 : : };  100 : :   101 : :   102 : : inline double  103 : : ShotTimeCalculator::translatePositionToDoseMap(int32\_t patternPos,   104 : : int64\_t subFieldPos,   105 : : double doseMapPos,   106 [ + - ][ + - ]: 2505 : double meshSize) {  107 : : return ( patternPos + subFieldPos - doseMapPos ) / meshSize;  108 : : }  109 : :   110 : : inline double  111 : 1094 : ShotTimeCalculator::getBlkUnit() {   112 : : return BLK\_UNIT;  113 : : }  114 : :   115 : :   116 : : } // namespace nft  117 : : #endif |

|  |
| --- |
| Generated by: [LCOV version 1.9](http://ltp.sourceforge.net/coverage/lcov.php) |