

```
/*
*      Eratosthenes.c                      1.003                      23.04.2011
*
*      Primzahl-Berechnung nach Eratosthenes in einem iterierten
*      Verfahren.
*
*      Literatur:
*      [0397Bibel]
*      "Die Bibel, oder die ganze Heilige Schrift Alten und Neuen Testaments
*      nach der deutschen Uebersetzung Dr. Martin Luthers.", Revision durch
*      Dr. Johann Philipp Fresenius, (1751); Druck und Verlag von Heinrich
*      Ludwig Brönner, Frankfurt am Main, 40. Auflage, (1841); der Text gilt
*      als vollständig abgeschlossen seit dem Konzil von Karthago, (397)
*
*      [1995OliS]
*      Olivastro D., Schmidt M.: Das chinesische Dreieck (Ancient Puzzles),
*      Lizenzausgabe mit freundlicher Genehmigung des Droemer Knaur
*      Verlages, München für Zweitausendeins, Frankfurt am Main, (1995)
*
*      Bearbeitung:
*      02.04.2011 - 23.04.2011 Norbert Südland, D-73431 Aalen
*/

/*
*
*      Zugang:
*      2 ist nur durch 2 und 1 teilbar, also eine Primzahl.
*      Eine Zahl ist durch 2 teilbar, wenn sie gerade ist.
*      Als mögliche weitere Primzahlen kommen somit nur ungerade Zahlen,
*      die größer als 2 sind, in Frage.
*
*      Es wird also ein Feld aus ungeraden und geraden Zahlen aufgebaut,
*      wobei alle geraden Zahlen außer 2 auf Null gesetzt werden.
*      Dann wird die 2 abgespeichert.
*      Es folgt die 3, wobei alle ungeradzahligigen Vielfache von 3 auf Null
*      gesetzt werden.
*      So wird 5, 7, 11 usw. als Primzahl ermittelt.
*      Ist das Quadrat der gefundenen Primzahl größer als das Feld, so sind
*      alle noch übrigen Zahlen Primzahlen.
*      Das Feld wird gelöscht, indem nun alle ungeraden Zahlen eingetragen
*      und alle geraden gelöscht werden. Mit Versatz wird nun wieder nach
*      den auftretenden Primzahlen gesucht.
*
*      Dieses Verfahren ist das "Sieb des Eratosthenes" (vgl. [1995OliS],
*      Seite 28-30) und auch für beliebig große Primzahlen erweiterbar.
*
*      Nach 5.Mose 19,15 ist ein richtiges Ergebnis auf zwei oder drei
*      unabhängigen Rechenwegen erhältlich. Nach 3.Mose 19,14 muss auch
*      der Rechenweg offengelegt werden.
*
*      Außer dem Ergebnis wird auch die benötigte Rechenzeit ausgegeben.
*/
```

```

#include          <SOBS.Kpf>
#include          <StdEA.Kpf>
#include          <StdBib.Kpf>

/*****
Nichts          Hauptteil( Ganzzahl      ParameterAnzahl ,
                        Buchstabe      *Parameter[ ] )
*****/
{
    Buchstabe          ErgebnisDateiName[ ] = "Eratosthenes.csv" ,
                        PrimzahlDateiName[ ] = "Eratosthenes.dmp" ,
                        *AusgabeFormat[ ] = {
                            "%10lu.Ergebnis = %10lu: %10lu Schleifen\n" ,
                            "%10lu.Ergebnis = %10lu: %10lu Schleife\n"
                        } ;

    NatuerlicheZahl      pb , z ;
    GrosseNatuerlicheZahl Feld[ 512 ] , fz , n , n_max , p , s , pz ;
    Datei                *ErgebnisDatei , *PrimzahlDatei ;
    ZeitStruktur          Zeit ;

    /* Vorbereitung: */
    /*-----*/
    Wenn( ParameterAnzahl > 1 )
        n_max = TextZuNatuerlicherZahl( Parameter[ 1 ] , NullZeiger , 10 ) ;
    Sonst
        n_max = 0 ;
    Wenn( !n_max )
        BildAusgabe( "Syntax:\nEratosthenes [AnzahlPrimzahlen]\n\n" ) ;
    pb = PlatzBedarf( GrosseNatuerlicheZahl ) ;
    SetzenWannStrgC( 1 ) ;

    /* Primzahlen erzeugen und ausgeben: */
    /*-----*/
    ErgebnisDatei = DateiOeffnen( ErgebnisDateiName , "w+t" ) ;
    Wenn( !ErgebnisDatei )
        BildAusgabe( "Fehler beim Öffnen der Datei \"%s\".\n\7" ,
                        ErgebnisDateiName ) ;
    PrimzahlDatei = DateiOeffnen( PrimzahlDateiName , "w+b" ) ;
    Wenn( !PrimzahlDatei )
        BildAusgabe( "Fehler beim Öffnen der Datei \"%s\".\n\7" ,
                        PrimzahlDateiName ) ;

    ZeitAbfragen( &Zeit ) ;
    DateiAusgabe(
        ErgebnisDatei ,
        "%02u:%02u:%02u:%02u\n" ,
        Zeit.Zeit_Stunden ,
        Zeit.Zeit_Minuten ,
        Zeit.Zeit_Sekunden ,
        Zeit.Zeit_HundertstelSekunden
    ) ;
}

```

```

/* Erste Primzahl abspeichern: */
/*-----*/
n = 1 ;
p = 2 ;
s = 1 ;
DateiPositionSetzen( PrimzahlDatei , 0 , SetzeAbEnde ) ;
InDateiSchreiben( &p , pb , 1 , PrimzahlDatei ) ;
DateiAusgabe( ErgebnisDatei , "%lu,%lu,%lu\n" , n , p , s ) ;
BildAusgabe( AusgabeFormat[ s == 1 ? 1 : 0 ] , n , p , s ) ;
ZaehlSchleife( z = 0 ; z < 512 ; z += 2 )
    Feld[ z ] = 0 ;
ZaehlSchleife( fz = 0 , n = 1 ; n < n_max ; fz++ ) {
    ZaehlSchleife( z = 1 ; z < 512 ; z += 2 )
        Feld[ z ] = z + fz * 512 ;
    ZaehlSchleife( s = 1 ; s < n ; s++ ) {
        DateiPositionSetzen( PrimzahlDatei , s * pb , SetzeAbAnfang ) ;
        VonDateiLesen( &p , pb , 1 , PrimzahlDatei ) ;
        Wenn( p * p > ( fz + 1 ) * 512 )
            Ausstieg ;
        pz = ( fz * 512 ) / p ;
        ZaehlSchleife( ; pz * p <= fz * 512 ||
            ( pz * p / 2 ) * 2 == pz * p ; pz++ ) ;
        ZaehlSchleife( ; pz * p < ( fz + 1 ) * 512 ; pz += 2 )
            Feld[ (NaturlicheZahl)( pz * p - fz * 512 ) ] = 0 ;
    }
    ZaehlSchleife( z = 1 ; z < 512 ; z += 2 )
        Wenn( Feld[ z ] > 2 ) {
            n++ ;
            p = Feld[ z ] ;
            DateiPositionSetzen( PrimzahlDatei , 0 , SetzeAbEnde ) ;
            InDateiSchreiben( &p , pb , 1 , PrimzahlDatei ) ;
            DateiAusgabe( ErgebnisDatei , "%lu,%lu,%lu\n" , n , p , s ) ;
            BildAusgabe( AusgabeFormat[ s == 1 ? 1 : 0 ] , n , p , s ) ;
            pz = p ;
            ZaehlSchleife( ; pz * p <= fz * 512 ||
                ( pz * p / 2 ) * 2 == pz * p ; pz++ ) ;
            ZaehlSchleife( ; pz * p < ( fz + 1 ) * 512 ; pz += 2 )
                Feld[ (NaturlicheZahl)( pz * p - fz * 512 ) ] = 0 ;
            Wenn( n >= n_max )
                z = 512 ;
        }
    }
}
ZeitAbfragen( &Zeit ) ;
DateiAusgabe(
    ErgebnisDatei ,
    "%02u:%02u:%02u:%02u\n" ,
    Zeit.Zeit_Stunden ,
    Zeit.Zeit_Minuten ,
    Zeit.Zeit_Sekunden ,
    Zeit.Zeit_HundertstelSekunden
) ;
AlleDateienSchliessen( ) ;
}
/*__ Ende von Hauptteil __*/

/*
* Ende von "Eratosthenes.c"
*/

```