

Break

| |
|----------------------|
| COLLABORATORS |
|----------------------|

| | | | |
|---------------|-------------------------|---------------|------------------|
| | <i>TITLE :</i> Break | | |
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | | July 22, 2024 | |

| |
|-------------------------|
| REVISION HISTORY |
|-------------------------|

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
| | | | |

Contents

| | | |
|----------|----------------------------------|----------|
| 1 | Break | 1 |
| 1.1 | Break | 1 |
| 1.2 | TMP:Modula-2/Break.def | 1 |

Chapter 1

Break

1.1 Break

Prozeduren

| | | |
|-----------------|--------------|------------------|
| ExitBreak | GetBreak | InstallException |
| RemoveException | SetBreakMode | TestBreak |

Konstanten

| | | |
|-----------|--------|-----------|
| BreakText | CBreak | FullBreak |
| NoBreak | | |

1.2 TMP:Modula-2/Break.def

```
DEFINITION MODULE Break;
(* 9.6.90/bp *)

FROM SYSTEM IMPORT LONGSET;
FROM DosD IMPORT ctrlC, ctrlD, ctrlE, ctrlF;

CONST
  BreakText = '\n*** BREAK\n';
  NoBreak   = LONGSET{};
  CBreak    = LONGSET{ctrlC};
  FullBreak = LONGSET{ctrlC..ctrlF};

PROCEDURE SetBreakMode(when:LONGSET);
(*
 * when legt fest, auf welche Signale geachtet wird.
 * CtrlC ist bei Programmstart als default gesetzt.
 *)

PROCEDURE ExitBreak;
(* Schreibt '*** BREAK' und terminiert das Programm. *)
```

```
PROCEDURE TestBreak();
(*
 * Sieht nach, ob ein BreakKey da ist und ruft dann
 * ExitBreak() auf.
 *)

PROCEDURE GetBreak():LONGSET;
(* Liefert die erlaubten und gedrückten Signale zurück *)

PROCEDURE InstallException;
(*
 * Installiert einen Exception-Handler,
 * der bei Ctrl-C das Programm beendet.
 *)

PROCEDURE RemoveException;
(* Unterbindet die Exception bei Ctrl-C wieder *)

END Break.def
```
