UNIVERSITÄT LEIPZIG

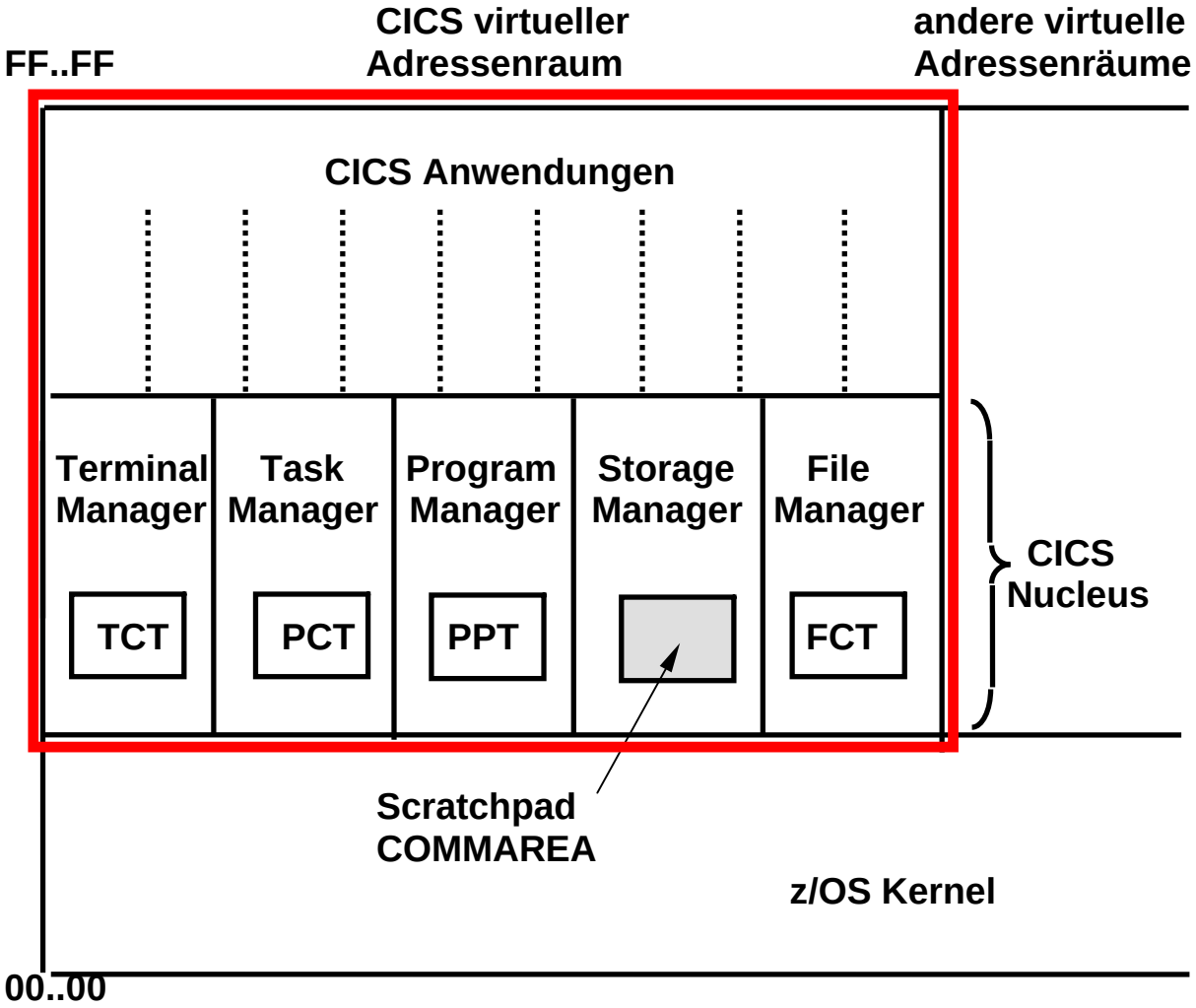
Enterprise Computing
Einführung in das Betriebssystem z/OS

Prof. Dr. Martin Bogdan
Prof. Dr.-Ing. Wilhelm G. Spruth

WS2012/2013

CICS Communication Teil 2

Basic Mapping Support



CICS läuft als lang laufender Stapelverarbeitungsjob in einem einzigen virtuellen Adressenraum.

Die CICS Nucleus Komponenten (Terminal Manager, Task Manager, Program Manager, Storage Manager and File Manager) nutzen den gleichen virtuellen Adressenraum wie alle Anwendungen. Jede Nucleus Komponente hat eine zugeordnete Tabelle: TCT, PCT, PPT, FCT. Über den Scratchpad werden Sessions eingerichtet: Der State einer Transaktion ist für die Folgetransaktion verfügbar. COMMAREA ist die Bezeichnung des I/O Puffers.

CICS Komponenten

Die Datenübertragung vom /zum CICS Terminal erfolgt normalerweise über einen Input/Output Puffer mit dem Namen COMMAREA.

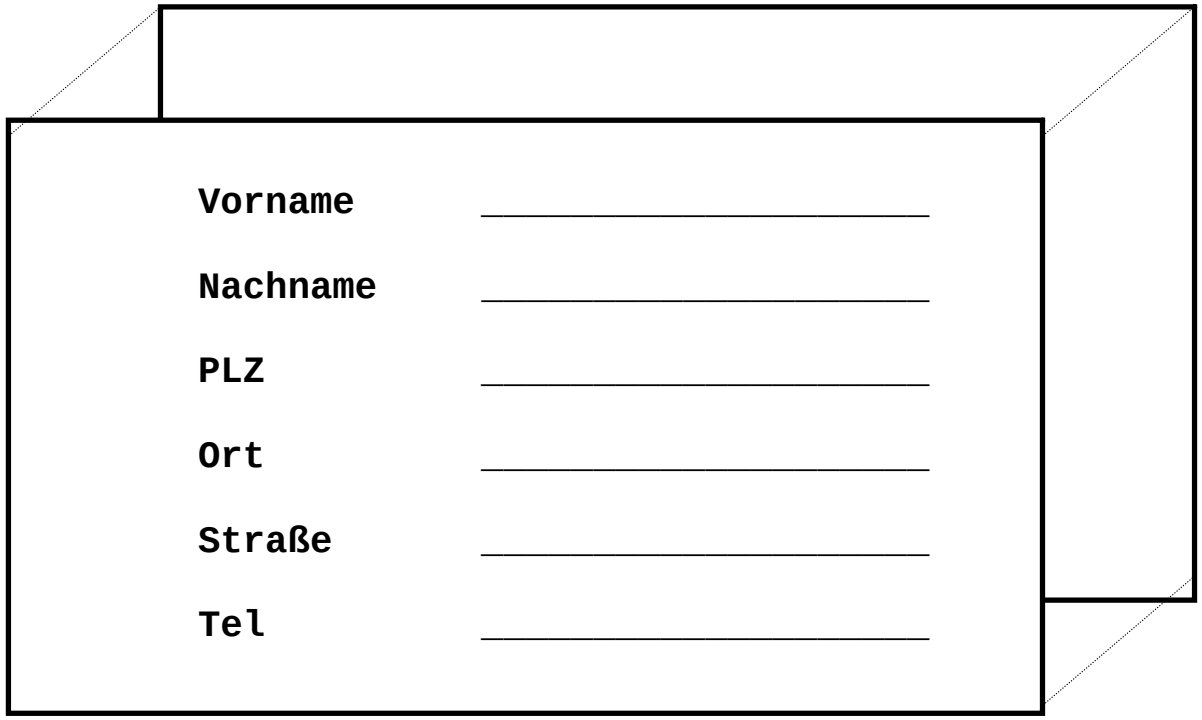
COMMAREA ist Teil des Scratchpad Bereiches, der vom CICS Storage Manager unterhalten wird.

```
struct adresse { char vorname  [20];
                 char nachname [20];
                 char plz      [20];
                 char ort      [20];
                 char strasse  [20];
                 char tel      [20];
                };

main()
{
...
EXEC CICS RECEIVE MAP(„adresse“) MAPSET(„wgsset“);
...
EXEC CICS SEND MAP(„adresse“) MAPSET(„wgsset“);
...
}
```

Die Daten, welche ein Anwendungsprogramm in den COMMAREA I/O Puffer zwecks Ausgabe an den Terminal stellt, werden normalerweise in der Form einer Struktur dargestellt, die oft auch als „Unit Record“ bezeichnet wird.

Wiedergegeben ist ein Beispiel in der Programmiersprache C/C++ .

Presentation Space Bildschirmpuffer, 24 x 80 Bytes



Bildschirm, 24 Zeilen, je 80 Zeichen monospaced

Struktur eines CICS Programms

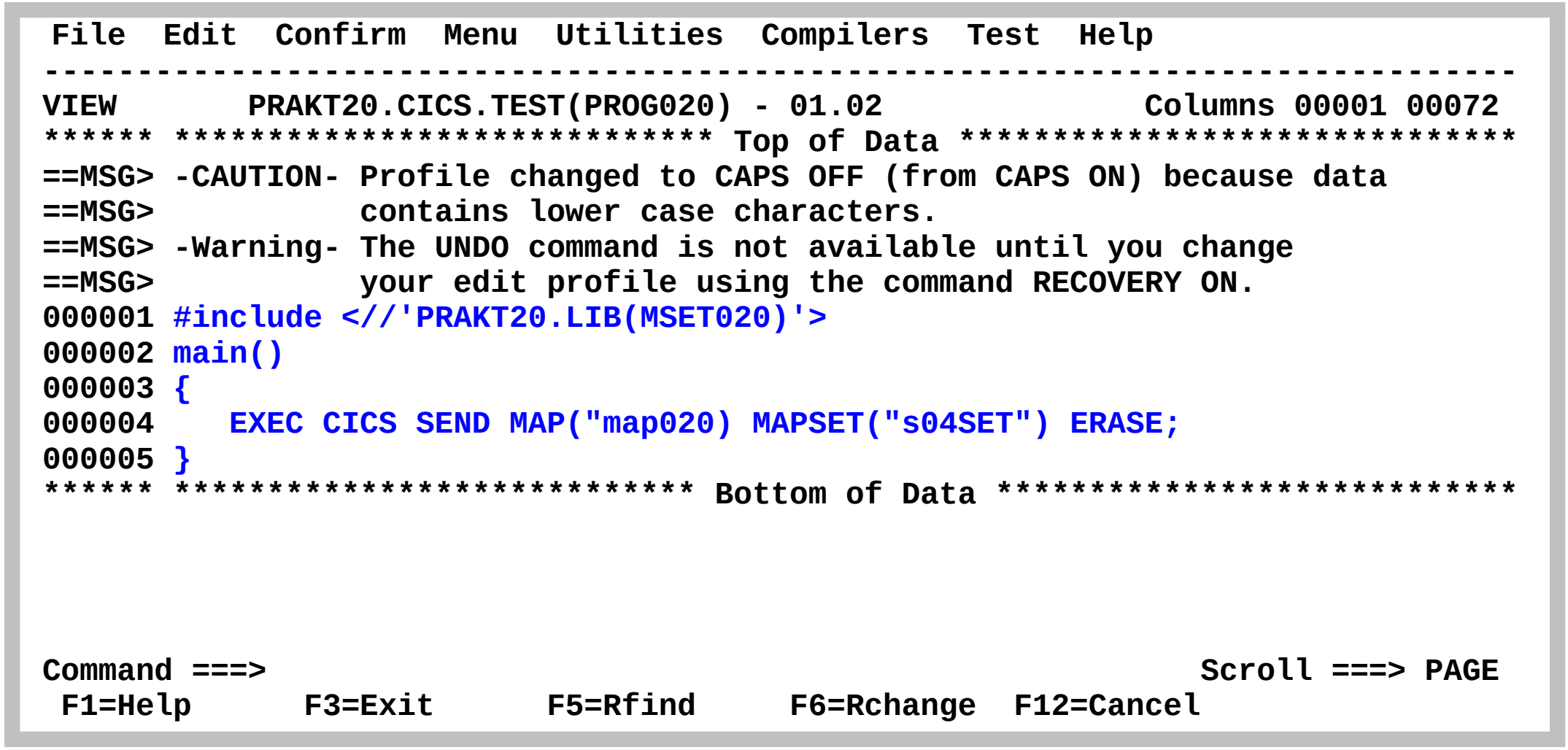
Der Presentation Space Bildschirmpuffer hat eine Strultur von 24 Worten zu je 80 Byte. Jede Wortadresse und Byte Adresse innerhalb eines Wortes entspricht genau der Zeilen- und Spaltenadresse. auf der das Byte auf dem Bildschirm wiedergegeben wird.

```
#include <//'PRAKT20.LIB(MSET020)'>

main()
{
EXEC CICS SEND MAP("map020") MAPSET("s04set") ERASE;
}
```

Struktur der „Hello World“ Transaktion

Im Folgenden schauen wir uns ein einfaches CICS Hello World Programm in C/C++ an, einschließlich seiner BMS Präsentationslogik.

Dargestellt ist der CICS Hello World Programm Code. Es wird eine Map mit dem Namen map020 gesendet, die ein Teil des Mapsets s04set ist.



Dies ist das gleiche Programm in der ISPF Editor Darstellung

```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
 ------------------------------------------------------------------------------
 EDIT       SPRUTH.CICS.TEST04(MAP04) - 01.02               Columns 00001 00072
 ****** ***************************** Top of Data ******************************
 ==MSG> -CAUTION- Profile changed to CAPS ON (from CAPS OFF) because the
 ==MSG>           data does not contain any lower case characters.
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 000001 //PREPARE  JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID
 000002 //ASSEM    EXEC DFHMAPS,MAPNAME='S04SET',RMODE=24
 000003 //SYSUT1   DD *
 000004 S04SET  DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
 000005 *       MENU MAP.
 000006 map020  DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
 000007         DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34,                X
 000008                INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
 000009         DFHMDF POS=(12,27),ATTRB=(ASKIP,NORM),LENGTH=26,               X
 000010                INITIAL='MAY THE FORCE BE WITH YOU!'
 000011         DFHMSD TYPE=FINAL
 000012         END
 000013 /*
 000014 //
 Command ===>                                                 Scroll ===> PAGE
  F1=Help      F3=Exit      F5=Rfind     F6=Rchange  F12=Cancel
```

…und hier ist der ISPF Editor Screen mit der Beschreibung der MAP in der BMS Sprache. Der Mapset unseres Hello World Programms besteht aus einer einzigen Map, die mit Hilfe von EXEC CICS SEND MAP gesendet wird.

```
S04SET  DFHMSD TYPE=MAP,MODE=INOUT,LANG=C,STORAGE=AUTO,TIOAPFX=YES
*       MENU MAP.
map020  DFHMDI SIZE=(24,80),CTRL=(PRINT,FREEKB)
        DFHMDF POS=(9,23),ATTRB=(ASKIP,NORM),LENGTH=34,
               INITIAL='WELCOME TO THE MAGIC WORLD OF CICS'
        DFHMDF POS=(12,27),ATTRB=(ASKIP,NORM),LENGTH=26,
               INITIAL='MAY THE FORCE BE WITH YOU!'
        DFHMSD TYPE=FINAL
        END
```

Beispiel für ein einfaches BMS Programm

Ein BMS Programm verwendet ausschließlich drei Arten von Befehlen, nämlich

DFHMSD	**Data Facility Hierarchical Map Set Definition.** **Mit Hilfe dieses Befehls wird ein Mapset definiert.**
DFHMDI	**Data Facility Hierarchical Map Definition Information.** **Mit Hilfe dieses Befehls wird eine Map definiert.**
DFHMDF	**Data Facility Hierarchical Map Data Field.** **Mit Hilfe dieses Befehls wird ein Feld innerhalb einer Map definiert.**

Der Mapset hat den Namen S04SET, die einzige Map dieses Mapsets hat den Namen map020. Jede Map innerhalb des Mapsets muss durch einen Namen gekennzeichnet sein.

Der Parameter TIOAPFX=YES in dem DFHMSD Befehl fügt ein 12-Byte Feld an den Anfang jeder Map ein.

… und hier verraten wir Ihnen ein Geheimnis: Es gibt überhaupt keine BMS Sprache. Was Sie hier sehen ist in Wirklichkeit ein Assembler Programm, welches ausschließlich aus Assembler Makros besteht, nämlich den drei Makros DFHMSD. DFHMDI und DFHMDF. IBM hat sich die Entwicklung einer eigenen BMS Sprache erspart, und stattdessen Assembler Macros verwendet. EXEC DFHMAPS in der JCL File bewirkt ein Assembler Compile, Link and Go.

Die beiden DFHMDF Befehle in der JCL File definieren 2 Felder in der Map, die mit Hilfe des EXEC CICS SEND MAP Befehls an den Terminal gesendet werden. Diese beiden Felder werden mit statischer Information initiiert. Für das erste der beiden Felder ist dies:

WELCOME TO THE MAGIC WORLD OF CICS

Angegeben wird, in welcher Zeile (Zeile 9) und Spalte (Spalte 13) dieses Feld in den Presentation Space Puffer geladen werden soll.

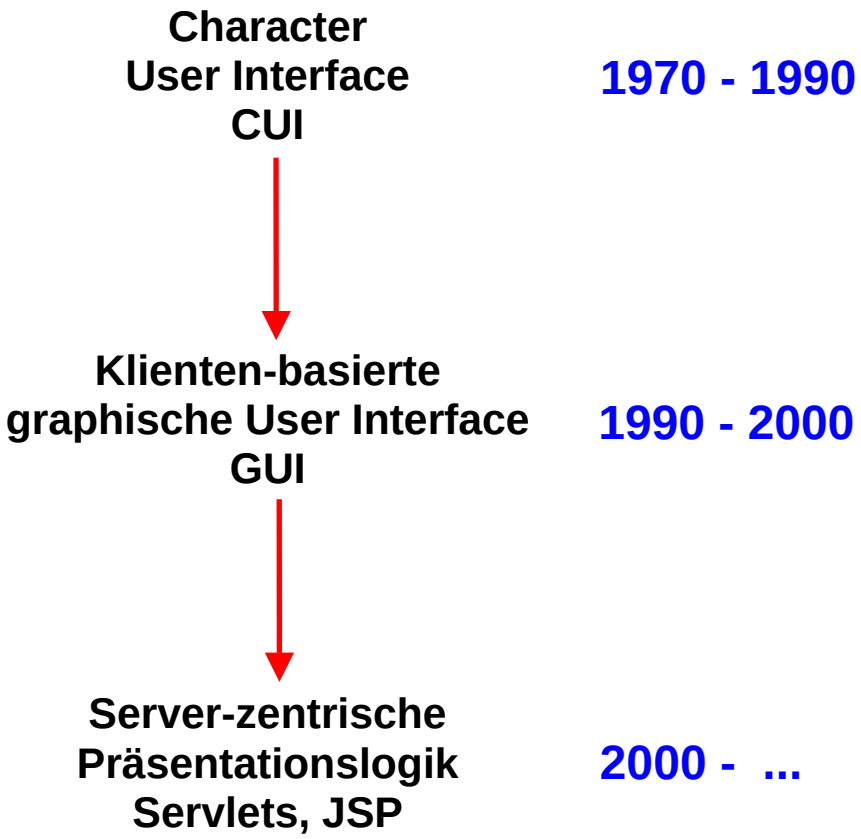
Es ist grundsätzlich möglich, CICS Maps mit Hilfe von Sprachen wie Cobol, C/C++ oder PL/1 zu erstellen. Dies geschah häufig in der Anfangszeit von CICS, wird heute aber praktisch so gut wie nie mehr gemacht.

```
WELCOME TO THE MAGIC WORLD OF CICS


MAY THE FORCE BE WITH YOU!




DFHAC2001 02/04/01 11:31:55 A06C001 Transaction '' is not recognized. Check
 that the transaction name is correct. CEDA DISPLAY GROUP(SPRUTH4)
```

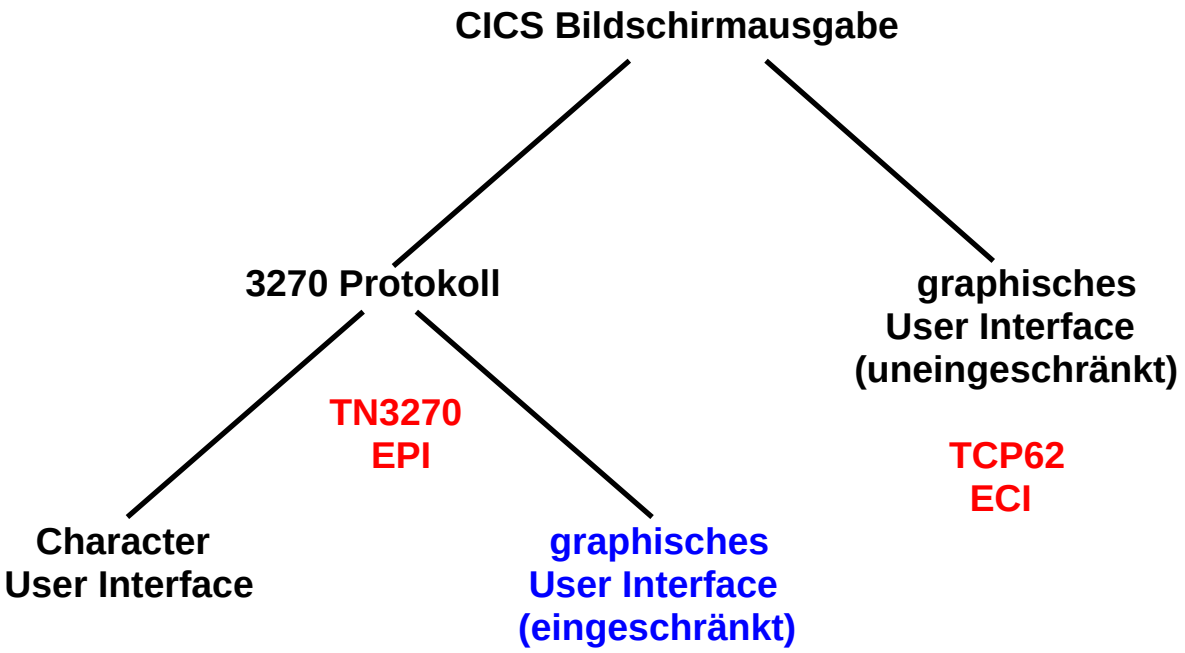
… und dies ist das tolle Ergebnis unseres CICS Hello World Programms.

Im Anhang befindet sich ein erweitertes Mapset-Beispiel mit 2 Maps.

Entwicklung der Präsentationslogik



Die Entwicklung der CICS Präsentationslogik erfolgte zeitlich in drei Schritten. Ursprünglich existierte für CICS nur die BMS orientierte Character User Interface.



CUI Character User Interface
GUI Graphical User Interface

Mit der Verfügbarkeit von PCs begann man, diese an Stelle der bisherigen Terminals einzusetzen.

Dies ermöglichte auf dem Klientenrechner Programme, die den Inhalt des Presentation Puffer für die Erstellung einer (eingeschränkten) graphischen Ausgabe benutzte.

Hierfür wurde die EPI (External Programming Interface) Schnittstelle zur Verfügung gestellt.

Diese Art der Verarbeitung wird auch heute noch vor allem für ältere CICS Anwendungen eingesetzt und als Screen Scraping bezeichnet.

Alternativen der Bildschirmausgabe

Screen Scraping

Als „Screen Scraping“ wird eine GUI für den 3270 Datenstrom bezeichnet.

Das Klienten Anwendungsprogramm (z.B. in Java geschrieben) empfängt den 3270 Datenstrom über die EPI (External Programming Interface) Schnittstelle und erzeugt eine graphische Darstellung der Datenausgabe. Der Vorteil ist, dass diese Art der Bildschirmdarstellung transparent für die Server-seitigen CICS Anwendungs-programme ist.

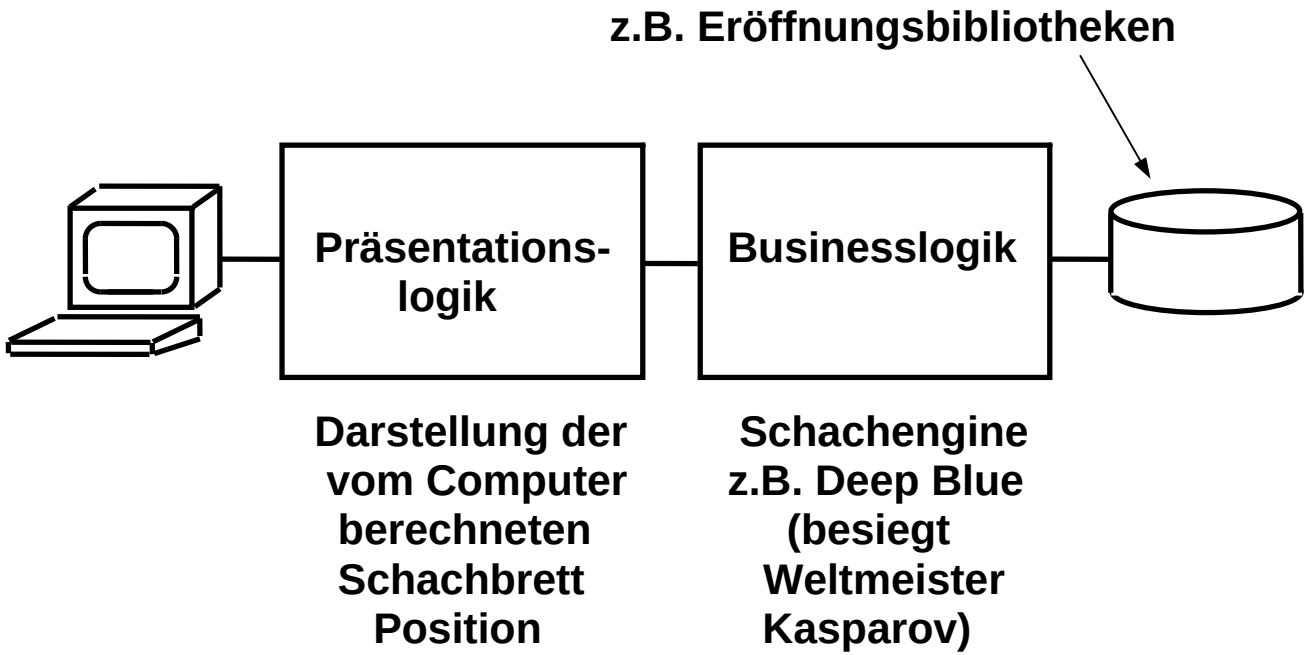
Erreicht wird eine gefälligere Darstellung, aber es fehlen manche Funktionen. Nicht möglich ist zum Beispiel eine Scroll Funktion mittels eines Scroll Balken, weil im 3270 Protokoll hierfür die Vorraussetzungen fehlen.

Ein weiterer Vorteil ist eine einfache, schnelle und kostengünstige Umstellung auf eine graphische Darstellung. Es existieren zahlreiche Software Produkte von unterschiedlichen Herstellern, welche die Umstellung erleichtern. Ein Beispiel ist „Host-on-Demand“ von IBM; es existieren viele ISV (Independent Software Vendor) Alternativen (z.B. von der Firma Attachmate).

Probleme:

- **2000 Benutzer erfordern die Wartung für 2000 Emulatoren auf den Arbeitsplatzrechnern**
- **Schwierigkeiten mit der Wartung der Maps. Maps können nicht mehr bequem geändert werden, z.B. "move field 1 byte“.**

Eine Server-seitige Lösung umgeht das Administrationsproblem. Z.B. „Host Access Transformation Services“ (HATS) von IBM läuft auf einem Server (z.B. einer zLinux Maschine) und konvertiert eine 3270 Nachricht in Browser-fähiges HTML Format.

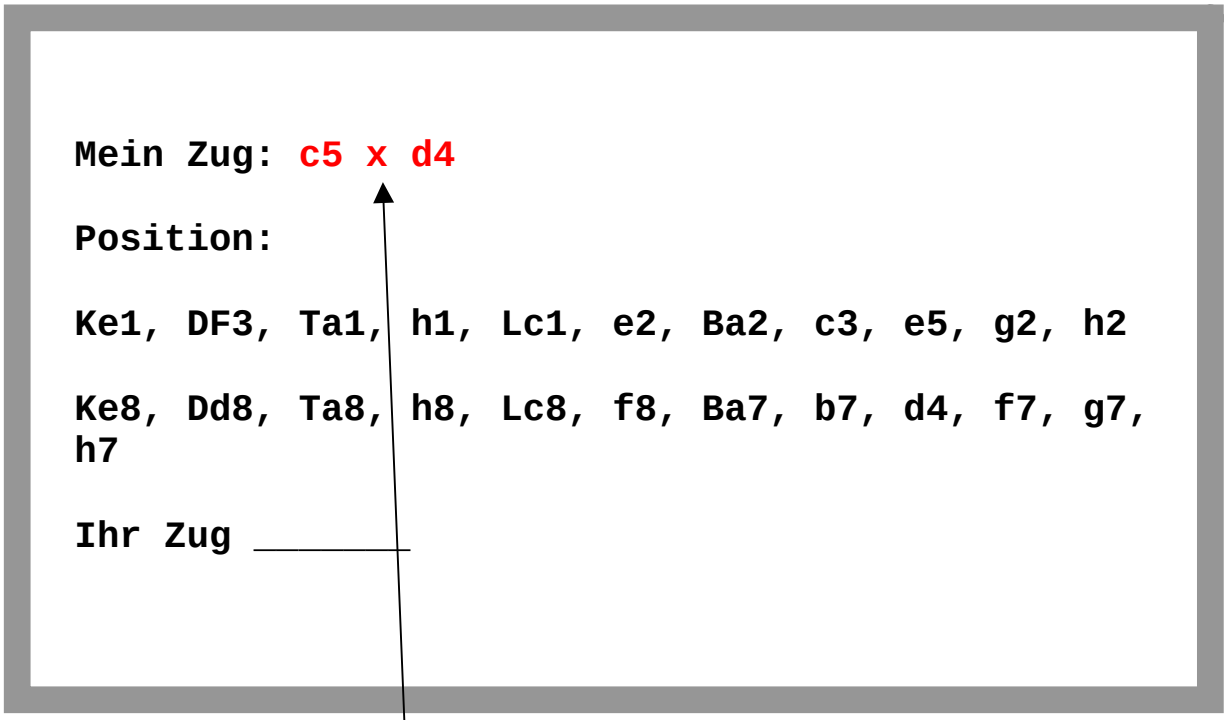


Beispiel Computer-Schachprogramm

Gliederung in Präsentationslogik und Businesslogik

Als Beispiel schreiben wir ein Computer Schachprogramm. (Wir übersehen, dass man ein Mainframe vermutlich nicht zum Schachspielen benutzen würde, und wenn doch, das Schachprogramm wahrscheinlich nicht unter CICS laufen würde).

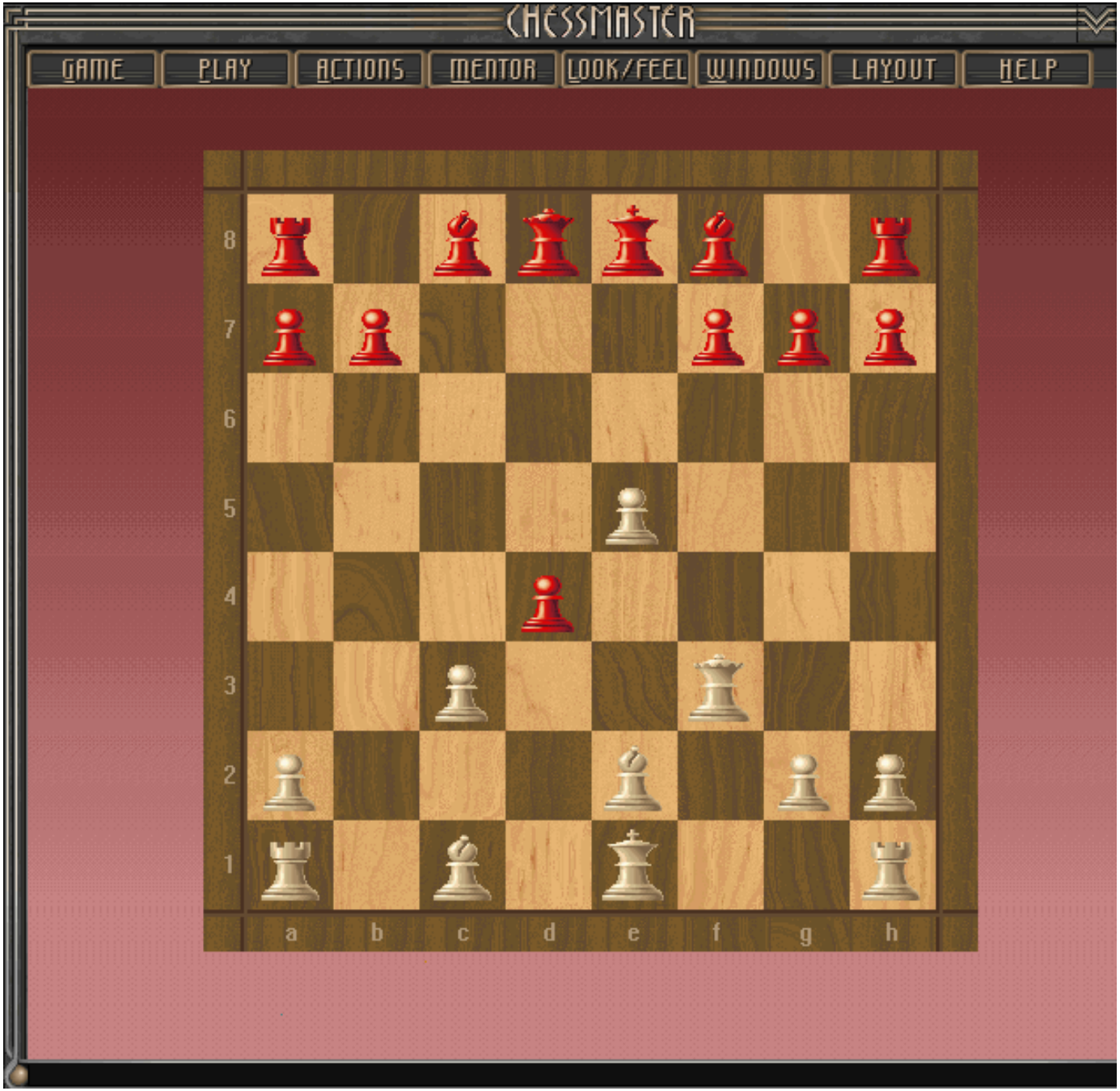
Die Business Logik läuft als CICS Anwendung. Die Präsentationslogik wird einmal mit Hilfe von BMS erstellt und als 3270/CUI dargestellt, und ein zweites Mal mit Hilfe von Screen Scraping graphisch dargestellt.



Minimale Präsentationslogik

Dies ist die Darstellung mit Hilfe der BMS 3270/CUI Präsentationslogik. Der Inhalt des Schachbretts ist mit Hilfe der „International Chess Notation“ dargestellt. Wiedergegeben sind die Positionen der einzelnen Figuren auf dem Schachbrett. Der Computer spielt mit den schwarzen Steinen, und eine seiner Figuren auf dem Feld c5 hat gerade eine weiße Figur auf dem Feld d4 geschlagen.

Der dargestellte Bildschirminhalt ist in dieser Form im Presentation Space Puffer des Terminals abgelegt, und wird unverändert auf dem Bildschirm wiedergegeben. Mitglieder eines Schachclubs sind mit dieser Notation sehr vertraut.

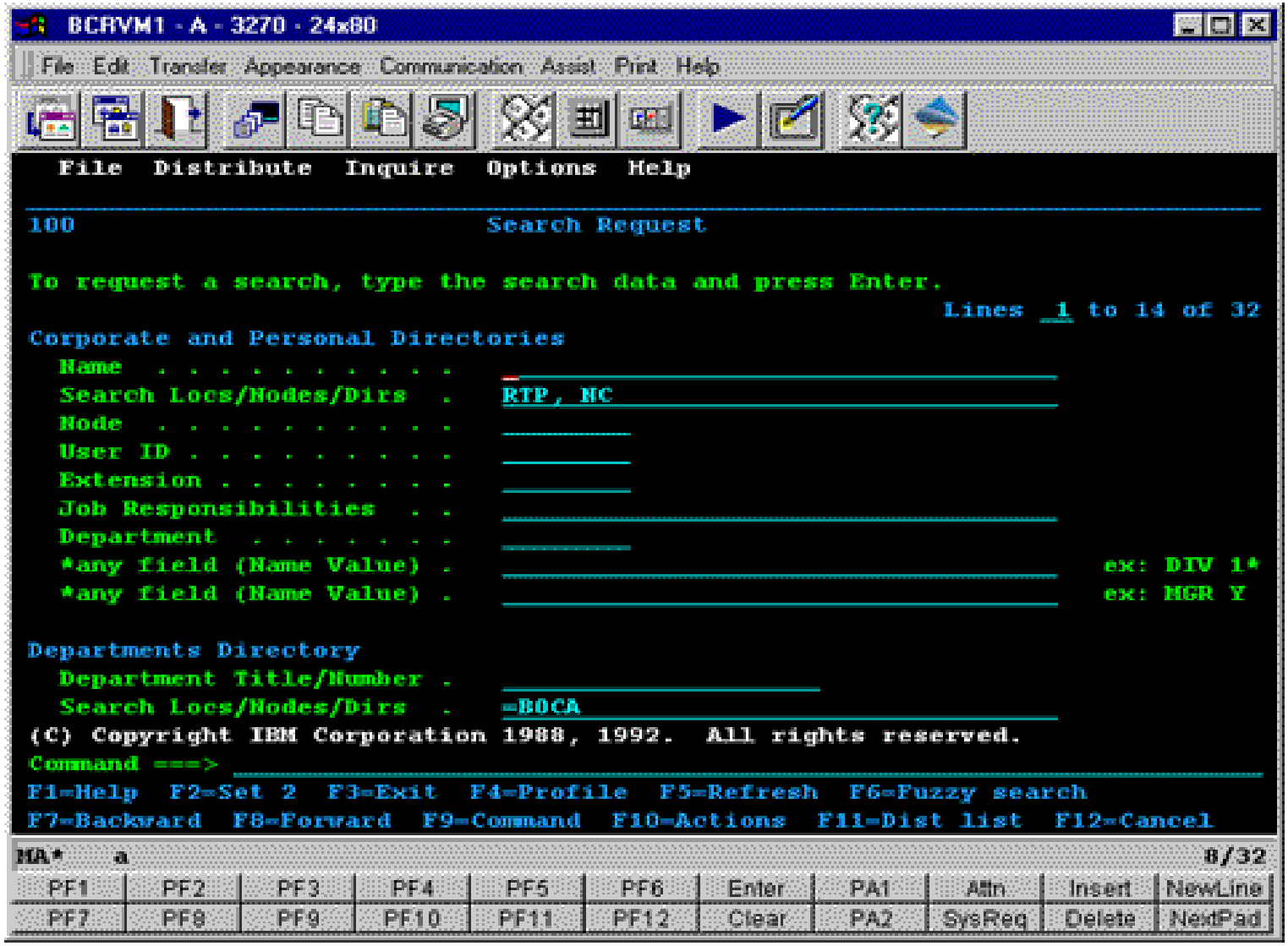


Graphische Präsentations-Logik

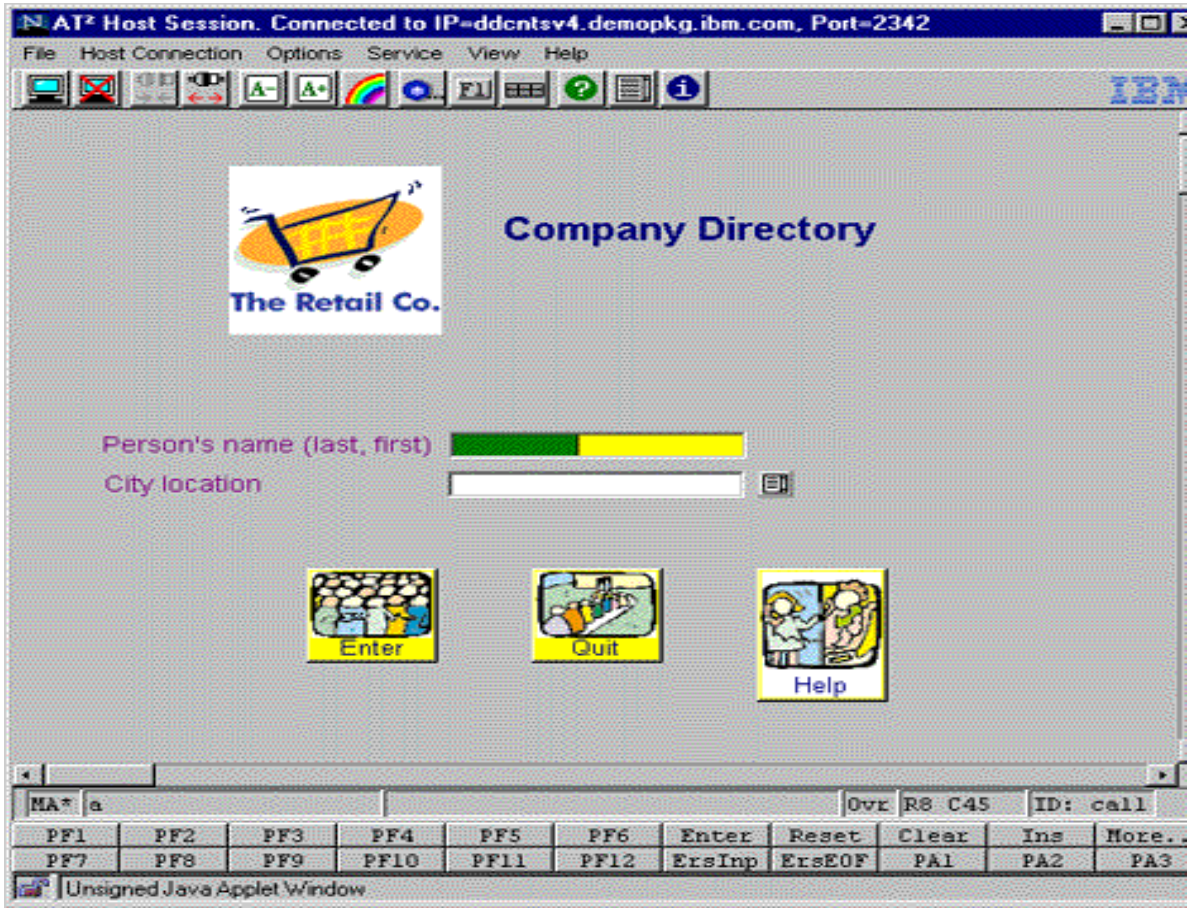
Alternativ greift ein Java Programm mit Hilfe der EPI Schnittstelle auf den Inhalt des Presentation Space Buffers zu und bereitet die Darstellung wie gezeigt graphisch auf.

Wichtig ist, dass die gezeigten beiden Bildschirm-Darstellungen inhaltlich identisch sind.

Es sind zusätzliche Funktionen möglich. Z. B. kann die zuletzt gezogene Figur blinkend dargestellt werden, Figuren können mit der Maus bewegt werden, usw.



Unser Schachbeispiel ist sicher sehr einprägsam, aber sicher auch sehr realitätsfern. Ein mehr realistisches Beispiel würde z.B. den oben gezeigten Screen in ...



... etwas Ähnliches wie den hier gezeigten Screen umsetzen.

Zu beachten ist, dass auch Aktionen mit der Maus möglich sind, die über die EPI Schnittstelle in 3270 Aktionen umgesetzt werde können, oder aber lokal (z.B. eine Help Funktion) abgearbeitet werden.

Nachteilig ist beim Screen Scraping Ansatz, dass ein Teil der Präsentationslogik auf dem Klientenrechner ausgeführt wird. Dies verursacht zusätzliche Administrationskosten.

Unintelligente 3270 Terminals haben den Vorteil, dass nicht sehr viel schief gehen kann. Der Terminal funktioniert entweder, oder er funktioniert nicht. Im Problemfall kommt der Techniker mit einem Ersatzterminal, prüft mit einem einfachen Testgerät ob der Kabelanschluss ok ist (ja/nein), wenn ja tauscht den Terminal aus und nimmt den alten Terminal mit in die Werkstatt. Für diese Tätigkeit ist keine Spezialausbildung erforderlich. PCs mit zusätzlicher Software als Terminal Ersatz erfordern schnell im Problemfall einen Spezialisten.

Zur Abhilfe setzt man heute als Terminals gerne PCs mit nur rudimentären Funktionen ein, z. B. ohne Plattenspeicher und ohne USB Anschluss, auf denen z.B. nur ein Browser läuft. Derartige PCs werden als „Thin Clients“ bezeichnet. Die 3270 Emulations- und Screen Scraping Software wird auf einen Server verlagert, der eine ganze Reihe von derartig abgerüsteten Arbeitsplatzrechnern bedient. Auf dem Server läuft Software, welche 3270 Daten in eine Browser Darstellung umsetzt.

Es gibt zahlreiche Software Produkte, die eine derartige Umsetzung durchführen. Ein Beispiel ist „Host Access Transformation Services“ (HATS) von der IBM. Wir werden später in dem Modul „Java Connection Architecture“ ein Beispiel zeigen.

Die Firma IGEL Technology GmbH in 28199 Bremen ist ein führender europäischer Hersteller von Thin Client PC Hardware, siehe https://www.igel.com/de/. Besonders in der öffentlichen Verwaltung werden Thin Client PCs gerne eingesetzt.

Viele Hersteller haben das 3270 Protokoll für ihre Client/Server Produkte eingesetzt, aber es existieren

Probleme des Screen Scraping Ansatzes

Das 3270 Protokoll lässt einige Funktionen nicht zu, z.B.:

- **Scroll Bar**
- **Mehrere Fenster**

Als Lösung bieten sich zwei Ansätze an:

- **Erweiterung des 3270 Protokolls um die fehlenden Funktionen. Dies wurde z.B. von der Firma SAP für ihre System R/3 SAPGUI implementiert.**

- **Java und http für die Präsentationslogik. Wir werden dies später in dem Modul „Java Connection Architecture“ diskutieren.**

Anhang

Zur Vertiefung zeigen wir ein einfaches CICS Programm, welches zwei Maps in seinem Mapset verwendet.

Die erste Map (Eingabe Map) stellt einen Eingabe Screen dar, in den der Benutzer zwei Zahlen in zwei dafür vorgesehene Felder eingeben kann.

Die zweite Map stellt einen Ergebnis Screen dar. Hier wird das Ergebnis der Addition der beiden Zahlen wiedergegeben.

Addition von zwei positiven Zahlen

Summand 1: **(positiv und max. 6 Stellen)**

Summand 2: **(positiv und max. 3 Stellen)**

Dies ist die Eingabe Map

Addition von zwei positiven Zahlen

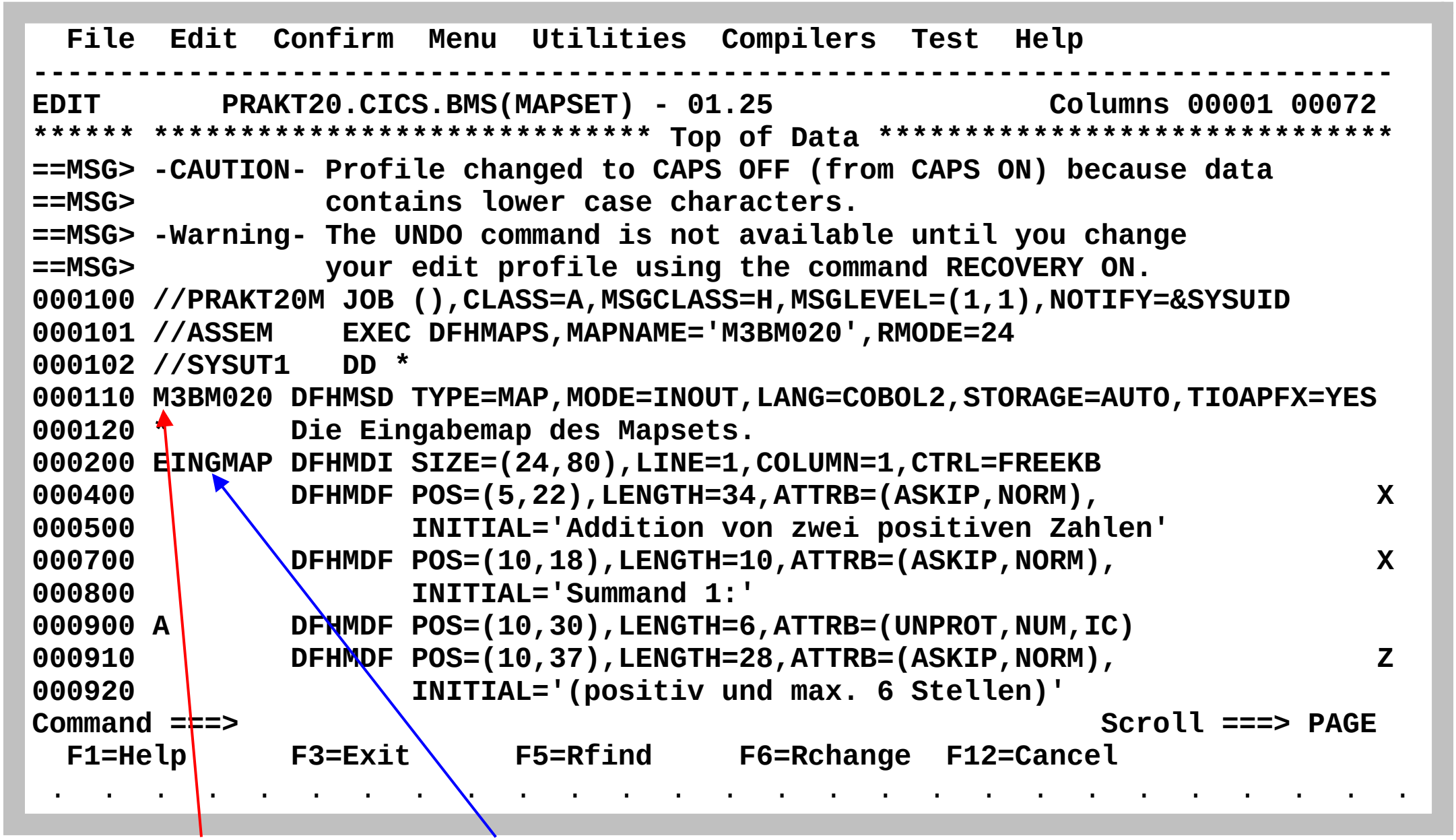
Summand 1: **333** **(positiv und max. 6 Stellen)**

Summand 2: **444** **(positiv und max. 3 Stellen)**

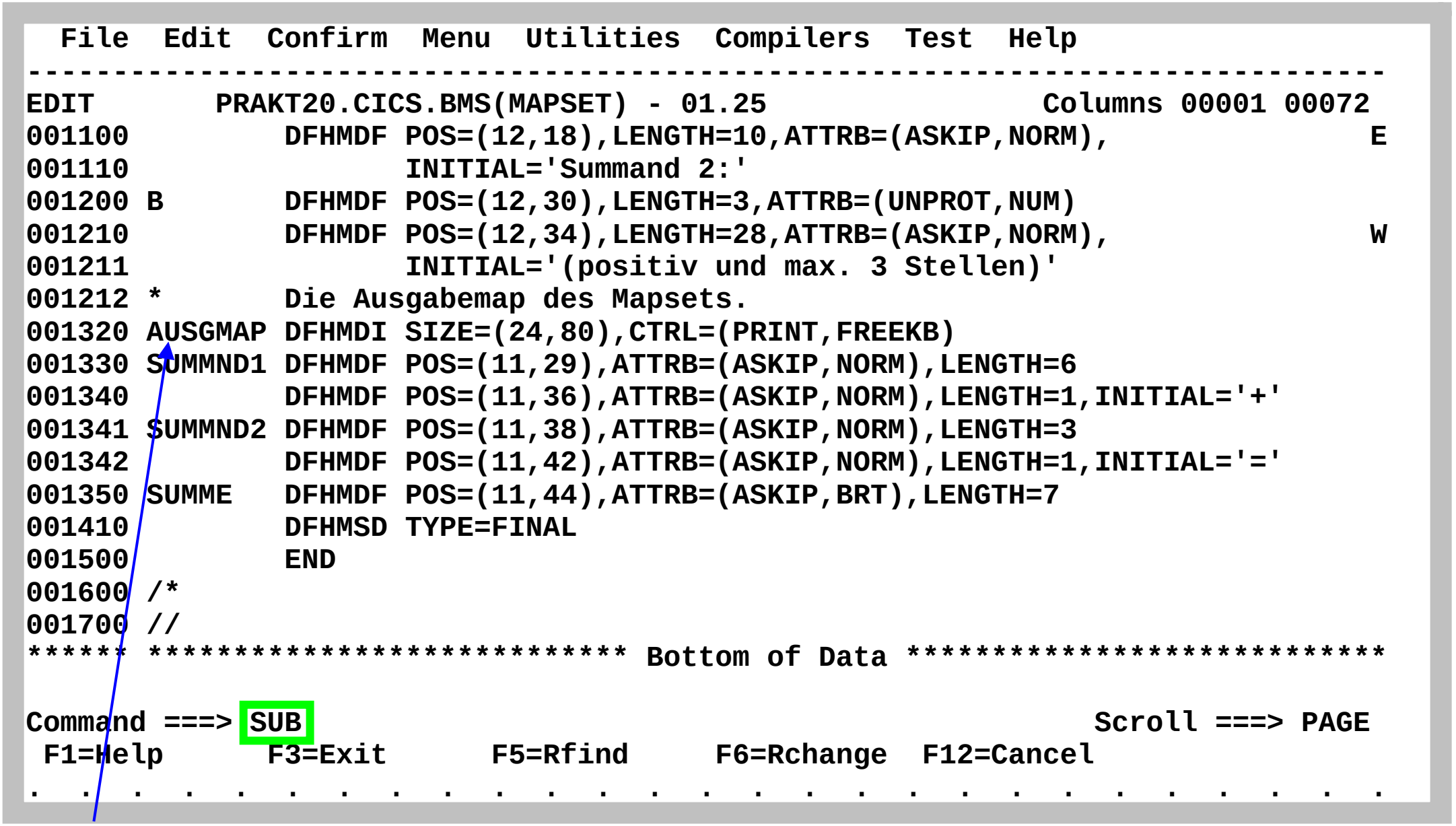
Dies ist die Eingabe Map, nachdem der Benutzer 2 Zahlen in die dafür vorgesehen (nicht markierten Felder) eingegeben hat.

333 + 444 = 777

… und dies ist die CICS Ausgabe. Die nächsten beiden Abbildungen zeigen das dafür benutzte BMS Programm, welches aus zwei ISPF Panels besteht..



Dies ist die erste Hälfte des BMS Programms



Map # 2 und dies die zweite Hälfte, in der die Map für die Ergebnisausgabe definiert wird.

Der Mapset mit dem Namen M3BM020 besteht aus genau zwei Maps. Eine Map (EINGMAP) definiert das Layout des ersten Screens, in den die beiden Summanden eingegeben werden sollen. Die zweite Map (AUSGMAP) definiert den Screen, der die Ausgabe enthält. Das JCL-Script beschreibt ein BMS-Programm, welches einen Mapset mit dem Namen "M3BM020" und den beiden Maps mit den Namen EINGMAP und AUSGMAP.

Die Ausführung des JCL Scripts (SUB oder SUBMIT) bewirkt die Erzeugung der entsprechenden Maps.