Bandbreite der Programmiersprachen

Carsten Gips (FH Bielefeld)

99 Bottles of Beer

99 bottles of beer on the wall, 99 bottles of beer. Take one down and pass it around, 98 bottles of beer on the wall.

98 bottles of beer on the wall, 98 bottles of beer. Take one down and pass it around, 97 bottles of beer on the wall.

[. . .]

2 bottles of beer on the wall, 2 bottles of beer. Take one down and pass it around, 1 bottle of beer on the wall.

1 bottle of beer on the wall, 1 bottle of beer. Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer. Go to the store and buy some more, 99 bottles of beer on the wall.

Imperativ, Hardwarenah: C

```c
#define MAXBEER (99)
void chug(int beers);
main() {
    register beers;
    for(beers = MAXBEER; beers; chug(beers--))  puts("");
    puts("\nTime to buy more beer!\n");
}
void chug(register beers) {
    char howmany[8], *s;
    s = beers != 1 ? "s" : "";
    printf("%d bottle%s of beer on the wall,\n", beers, s);
    printf("%d bottle%s of beeeeer . . . ,\n", beers, s);
    printf("Take one down, pass it around,\n");
    if(--beers) sprintf(howmany, "%d", beers); else strcpy(howmany, "No more");
    s = beers != 1 ? "s" : "";
    printf("%s bottle%s of beer on the wall.\n", howmany, s);
}
```

Imperativ, Objektorientiert: Java

```java
class bottles {
    public static void main(String args[]) {
        String s = "s";
        for (int beers=99; beers>-1;) {
            System.out.print(beers + " bottle" + s + " of beer on the wall, ");
            System.out.println(beers + " bottle" + s + " of beer, ");
            if (beers==0) {
                System.out.print("Go to the store, buy some more, ");
                System.out.println("99 bottles of beer on the wall.\n");
                System.exit(0);
            } else
                System.out.print("Take one down, pass it around, ");
            s = (--beers == 1)?"":"s";
            System.out.println(beers + " bottle" + s + " of beer on the wall.\n");
        }
    }
}
```

Logisch: Prolog

```prolog
bottles :-
    bottles(99).

bottles(1) :-
    write('1 bottle of beer on the wall, 1 bottle of beer,'), nl,
    write('Take one down, and pass it around,'), nl,
    write('Now they are all gone.'), nl,!.
bottles(X) :-
    write(X), write(' bottles of beer on the wall,'), nl,
    write(X), write(' bottles of beer,'), nl,
    write('Take one down and pass it around,'), nl,
    NX is X - 1,
    write(NX), write(' bottles of beer on the wall.'), nl, nl,
    bottles(NX).
```

Quelle: 99-bottles-of-beer.net/language-prolog-965.html, by M@

Funktional: Haskell

```haskell
bottles 0 = "no more bottles"
bottles 1 = "1 bottle"
bottles n = show n ++ " bottles"

verse 0  = "No more bottles of beer on the wall, no more bottles of beer.\n"
          ++ "Go to the store and buy some more, 99 bottles of beer on the wall."

verse n  = bottles n ++ " of beer on the wall, " ++ bottles n ++ " of beer.\n"
          ++ "Take one down and pass it around, " ++ bottles (n-1)
                                        ++ " of beer on the wall.\n"

main      = mapM (putStrLn . verse) [99,98..0]
```

Brainfuck

Programmiersprache Lox

```
fun fib(x) {
    if (x == 0) {
        return 0;
    } else {
        if (x == 1) {
            return 1;
        } else {
            fib(x - 1) + fib(x - 2);
        }
    }
}

var wuppie = fib;
wuppie(4);
```

Wrap-Up

- Compiler übersetzen formalen Text in ein anderes Format

- Berücksichtigung von unterschiedlichen
  - Sprachkonzepten (Programmierparadigmen)
  - Typ-Systemen
  - Speicherverwaltungsstrategien
  - Abarbeitungsstrategien

LICENSE