

```

// Ad-hoc Scanner
int scanner()
{
    char c=' ';
    int tokenclass=0; // tokenclass = 1 für Zahl, 2 für ==-Operator,
                     // 3 für Zuweisung =, 4 für Bezeichner, 5 für +-Operator 0 = ' '

    int tokenval;
    c = getchar();
    switch (c)
    {
        case ' ':
        case '\n': // Leerzeichen und Zeilenende ignorieren
            break;
        case '0':
        case '1':
        case '2':
        case '3':
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
            tokenval = c - '0';
            c = getchar();
            while ((c >= '0') && (c <= '9'))
            {
                tokenval = tokenval * 10 + c - '0';
                c = getchar();
            }
            ungetc(c, stdin); // Zeichen gehört nicht zu Zahl
            printf("Ganzzahl erkannt Wert =%i\n", tokenval);
            tokenclass = 1; // Tokenklasse 1 ist Ganzzahl
            break;
        case '=':
            if ((c = getchar()) == '=')
            {
                tokenclass = 2; // Tokenklasse 2 ist ==-Operator
                printf("Ist gleich-Operator == erkannt\n");
            }
            else
            {

```

Beispiel:

677

1. c = '6'

tokenval = 6

c = '7'

tokenval = 6 · 10 + 7 = 67

tokenclass = 1 (Zahl)

tokenval = 67

```

        ungetc(c, stdin); // Zeichen gehört nicht zu Zuweisung
        tokenclass = 3; // Tokenklasse 3 ist Zuweisung-Operator
        printf("Zuweisungsoperator erkannt\n");
    }
    break;
case 'a':
case 'b':
case 'c':
case 'd':
case 'e': //usw. bis 'z'
    while (c >= 'a' && c <= 'z')
        c = getchar();
    ungetc(c, stdin); // Zeichen gehört nicht zu Bezeichner
    printf("Bezeichner erkannt\n");
    tokenclass = 4; // Tokenklasse 4 ist Bezeichner
    break;
case '+': tokenclass = 5; // Tokenklasse 5 ist +-Operator
    printf("+ Operator erkannt\n");
    break;
default:
    tokenclass = 99;
    printf("Fehler");
}
return tokenclass; // erkannte token-Klasse zurückgeben
}

```

hier sind für Bezeichner
nur Kleinbuchstaben erlaubt