

- Trennung von **Modell** und **Benutzerschnittstelle**
 - **Beispiel:** Kommandozeilenaufruf `read -v my.data`

```
int main(int argc, char *argv[]) {
    Operation op = OP_NONE;
    switch(input) {
        case 'v':
            op      = OP_VALIDATE;
            break;
        ...
        default:
            err     = ERR_INVALID_OP;
    }
    ...
    perform(op, fname);
    ...
    return err;
}
```

```
typedef enum Operation {
    OP_NONE      = 0,
    OP_VALIDATE  = 1,
    OP_LOAD      = 2,
    ...
} Operation;
```

```
void perform(Operation op, char *file) {
    /* ... */
}
```

Vorteile:

- *Einfache/übersichtliche Wartbarkeit*
 - ↳ *Benutzer-Schnittstelle (Eingabe)*
 - ↳ *Code-Erweiterung*

- Eigener Aufzählungstyp abstrahiert von den Festlegungen für die Benutzerschnittstelle, Benutzerschnittstelle und `perform()` können unabhängig voneinander variiert werden