Enumerationer

- Vanligt problem: En variabel får endast ha vissa tillåtna värden
- Exempel: Veckodagar, månader, svarsalternativ i en meny...
- Inte bra att använda “magiska nummer”
  - gör koden svårläst och risken för fel ökar

```
int choice = getUserChoice();
if (choice == 0) { calculateCirc(); }
if (choice == 1) { calculateArea(); }
```

- Ett bättre sätt är att definiera konstanter:

```
const int CALCCIRCUM = 0;
const int CALCAREA = 1;

int choice = getUserChoice();
if (choice == CALCCIRCUM) { calculateCirc(); }
if (choice == CALCAREA) { calculateArea(); }
```

Enumerationer

- Ännu bättre sätt: Definiera en namngiven mängd med namngivna heltalskonstanter => *enumeration (“uppräkning”)*
- Som standard motsvarar den första konstanten heltalet 0, följande konstant 1, osv.

```
enum menuChoices { CALCCIRCUM, CALCAREA };
enum menuChoices choice = getUserChoice();
if (choice == CALCCIRCUM) { calculateCirc(); }
if (choice == CALCAREA) { calculateArea(); }
```

- Större exempel: **`gradecalc.c`** i exempelmappen på Moodle
- Visar också hur man använder **`switch..case`** för att kontrollera enumerationens värde

Enumerationer

- Varje enumeration är egentligen ett heltalsvärde
- Anges inga explicita värden tilldelas den första enumerationen värdet 0, den andra enumerationen värdet 1, osv.
- Kan också ange ett eget startvärde för numreringen:

```
enum month { JANUARY = 1, FEBRUARY, MARCH, .... };
```

- ...och tilldela separata värden för en del eller alla enumerationer

```
enum priority { LOW = 0,
                MEDIUM = 50,
                ALMOST_MEDIUM,  // får värdet 51
                HIGH = 100 };
```

Enumerationer

- Har tidigare sett exempel på hur **`typedef`** kan användas för att skapa alias för en **`struct`**
- Kan också använda **`typedef`** för att förenkla användningen av enumerationer:

```
typedef enum {
    MONDAY=1, TUESDAY, WEDNESDAY, THURSDAY,
        FRIDAY, SATURDAY, SUNDAY
} weekday;

weekday w = TUESDAY;
```

Enumerationer för tillstånd

- En del programmeringsuppgifter kan lösas genom att observera att programmet kan befinna sig i olika *tillstånd* (eng. *states*)
- Antalet tillstånd är begränsat => kan använda en enumeration för att räkna upp de tillstånden och skapa en *tillståndsvariabel* av denna typ för att hålla reda på det aktuella tillståndet
- Typiska exempel: (nätverks)kommunikation enligt ett visst protokoll, (delar av) spel...

Enumerationer för tillstånd

```
typedef enum
{
    HEALTHY,
    WOUNDED,
    NEAR_DEATH,
    DEAD
} ai_status;

ai_status status = HEALTHY;
...
switch (status)
{
    case HEALTHY: { normal_attack(); }
    case WOUNDED: { flee(); }
    case NEAR_DEATH: { kamikaze_attack(); }
}
```