

I. Enum (тип изброен)

Кога го използваме? – когато имаме някаква променлива и предварително ни е известно множеството от стойностите, които тя може да приема, с enum можем да си моделираме променливата. Най-често enum се използва в комбинация със switch.

Пр.: а) enum Weekday {MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY};

б) enum Season {WINTER, SPRING, SUMMER, FALL};

в) enum RealEstate {HOUSE, FLAT, PLOT};

г) enum ZodiacSign {ARIES, TAURUS, GEMINI, CANCER, LEO, VIRGO, LIBRA, SCORPIO, SAGGITARIUS, CAPRICORN, AQUARIUS, PISCES};

Синтаксис:

- enum <име_на_тип> { <идентификатор₁>=[<константен_израз₁>]опц,
 <идентификатор₂>=[<константен_израз₂>]опц, ...,
 <идентификатор_n>=[<константен_израз_n>]опц }, където:

<име_на_тип> е името, което даваме на дефинирания от нас тип (weekday, month, vowel, etc.);

<идентификатор_i> е стойност, която нашият тип ще може да приема (ако типът ни е month, можем да сложим january, february,...,december).

[<константен_израз>]опц, е числова константа (код), който съответства на нашия идентификатор и чрез която можем да викаме или можем да присвояваме на променлива от дефинирания тип.

Особености:

Константа на enum-а наричаме една от стойностите, които може да приема. За примера в б) горе, константите на enum-а са WINTER, SPRING, SUMMER и FALL.

- Стойностите, които променлива от дефинирания от нас тип може да приема, са САМО ТЕЗИ, които изрично сме обявили в идентификаторите в enum-а.

Пр.:

```
enum Weekday {SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY};
Weekday day;
//множество от стойности, които може да приема day:
//SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
```

- Пишем стойностите вътре в enum-а с главни букви. Ще се компилира и с малки, но не е добра практика.
- Константите на enum-а трябва да бъдат уникални в рамките на този enum и в рамките на програмата.

Пр.:

```
//няма да го подчертае като грешка, но няма да се компилира:
enum Month {JANUARY, JULY}
enum SummerMonth {JULY}
```

- Не е възможно въвеждането на променлива от дефинирания от нас тип!

Пр.:

```
//НЕ МОЖЕ!
enum Weekday {SUNDAY=7, MONDAY=1, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY=6};
Weekday a;
cin >> a;
```

Семантика:

- Всяка константа на enum-а вътрешно се свързва с код.
- Когато изкарваме променлива от дефинирания от нас тип на конзолата, се изкарва кодът, с който тя е свързана.

- В [константен израз]опц. можем да напишем някаква числова константа, която ще играе ролята на код за съответната константа на enum-а.

Пр.:

```
enum Weekday {SUNDAY = 3, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY};
Weekday day;
```

- Ако никъде в enum-а няма такава константа със свързан с нея код, кодовете на съответните стойности по подразбиране стават последователно 0, 1, ..., <брой_идентификатори-1>:

Пр.:

```
enum Weekday {SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY};
//SUNDAY - 0;
//MONDAY - 1;
//...
//SATURDAY - 6;
```

- Ако всички идентификатори са свързани със стойности, вътрешното представяне (кодът) на всяка константа на enum-а е указаната стойност:

Пр.:

```
enum Weekday {SUNDAY=7, MONDAY=1, TUESDAY=2, WEDNESDAY=3, THURSDAY=4, FRIDAY=5, SATURDAY=6}
//MONDAY - 1;
//TUESDAY - 7;
//...
//SATURDAY - 6.
```

- Ако някои идентификатори са свързани със стойности, а други – не са, стойностите на тези, които нямат свързани стойности, се определят от стойността на кода на предишния идентификатор +1.

Пр.:

```
enum Weekday { SUNDAY = 7, MONDAY = 1, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY = 13, SATURDAY };
cout << THURSDAY << " "<< SATURDAY;//4 14
```

- Как да изведем кода на константа на enum-а на конзолата?

Пр.:

```
enum Weekday { SUNDAY = 7, MONDAY = 1, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY = 13, SATURDAY };
cout << SUNDAY; //7
cout << (int)SUNDAY; //7 (записът е еквивалентен на горния);
```

Задача. Да се напише програмен фрагмент, който извежда с думи константите на типа Weekday.

```
enum Weekday {SUNDAY=7, MONDAY=1, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY=6};
Weekday d = FRIDAY;
switch (d)
{
case SUNDAY: cout << "SUNDAY \n"; break;
case MONDAY: cout << "MONDAY \n"; break;
case TUESDAY: cout << "TUESDAY \n"; break;
case WEDNESDAY: cout << "WEDNESDAY \n"; break;
case THURSDAY: cout << "THURSDAY \n"; break;
case FRIDAY : cout << "FRIDAY \n"; break;
case SATURDAY: cout << "SATURDAY \n";
}
```