**Switch Statement(Switch Case)**

Dəyişənin sahib olduğu dəyərə uyğun olaraq müəyyən bir əməliyyat icra olunacaqsa, else if şərt operatoru əvəzinə Switch Case yazılmalıdır. Case keyword qarşısında yoxlanılacaq dəyişənin tipinə uyğun dəyərlər yazılır və dəyər uyğun gələrsə işə düşəcək kod bloku qeyd olunur. Default keyword vasitəsi ilə yoxlanış zamanı heç bir dəyər uyğun gəlməzsə default olaraq icra olunmasını istədiyimiz əməliyyatı yazabilərik. Lakin belə bir məcburiyyət yoxdur.

switch(expression or variable)

{

case value1:

Statement1

break;

case valueN:

StatementN

break;

default:

StatementDefault

break;

}

Compile time error olan Fall-through error-un qarşısını almaq üçün break keyword istifadə edilir. Break keyword, case-in tamamlandığını və switch case-in bitdiyini göstərir. Lakin bunun qarşısını sadəcə break keyword ilə yox, goto case value, return, throw və continue keyword-lərdən də istifadə edə bilərik. Və ya case daxili boş buraxılır, əgər value uyğun gələrsə ondan sonra gələn case işə düşür.

Explicit Fall-through olaraq bilinən goto case value, müəyyən əməliyyatdan sonra başqa bir case-i işə salmaq üçün istifadə edilir. Break keyword-ə ehtiyac qalmır

Müəyyən bir case-də əməliyyat icra etmək əvəzinə Exception throw edilə bilər. Bu zaman break keyword throw Exception ilə əvəz olunur.

Method daxilində yazılmış olan switch case-də, hansısa case uyğun gəldiyi təqdirdə bir başa olaraq method-u bitirmək return keyword istifadə edilir.

Loop daxilində yazılmış olan switch case-də, hansısa hala uyğun gəldiyi təqdirdə case-i bitirib döngüyə başa qayıtmaq üçün continue keyword istifadə edilir. Break keyword-ə ehtiyac qalmır

**Switch Expression**

C# 8 ilə birlikdə gəlmiş olan Switch Expression ilə biz, müxtəlif case-lərə uyğun olaraq sadəcə dəyər mənimsətmə prosesi həyata keçirəcəyimiz zaman Switch Case və ya Else If Condition əvəzinə daha asan və qısa yazılışa sahib olan Switch Expression-dan istifadə edəcəyik.

Nəticədə case-ə uyğun olaraq verəcəyimiz dəyəri mənimsəcədək. Burada default keyword, underline ilə işarələnir. Switch Case-də olduğu kimi Default halını yazmaq məcburiyyətində deyilik. Case-ə uyğun olaraq Exception throw da mümkündür.

int dayCode = 5;

string dayStr = dayCode switch

{

1 => “Monday”,

…

7 => “Sunday”,

\_ => “DefaultValue”

}

Həmçinin aşağıdaki şəkildə yazılışı da mövcuddur.

public string GetDayStr(int dayCode) =>

dayCode switch

{

1 => “Monday”,

…

7 => “Sunday”,

\_ => “DefaultValue”

};

**Pattern Matching with Switch Statement and Switch Expression.**

C# 9 ilə birlikdə gəlmiş Relational Patterns, case-lərdə Relational operatorları istifadə etmək üçündür və aşağıdaki şəkildədir.

public string ClassifyNumber(int number) => number switch

{

< 0 => "Negative",

0 => "Zero",

> 0 => "Positive"

};

C# 9 ilə birlikdə gəlmiş Logical Patterns, case-lərdə Logical operatorları istifadə etmək üçündür və aşağıdaki şəkildədir.

public string CheckRange(int number) => number switch

{

> 0 and <= 10 => "Between 1 and 10",

> 10 or < -10 => "Outside range",

not 0 => "Non-zero",

\_ => "Zero"

};

public string CheckRange(int number)

{

switch(number)

{

case > 0 and <= 10: return "Between 1 and 10";

case > 10 or < -10: return "Outside range";

case not 0: return "Non-zero";

default: return "Zero";

};

}

When keyword ilə isə aşağıdaki şəkildə olacaq.

public string CheckRange(int number) => number switch

{

int num when num > 0 && num <= 10 => "Between 1 and 10",

int num when num > 10 or num < -10 => "Outside range",

int num when num != 0 => "Non-zero",

\_ => "Zero"

};

public string CheckAgeCategory(int age)

{

switch (age)

{

case int n when n < 13:

return "Child";

case int n when n >= 13 && n < 20:

return "Teenager";

default:

return "Unknown age category";

}

}

\*Pattern matching Property pattern ilə