

Семинар по „Увод в програмирането“

Условни конструкции

(if-else statements)

1. Какво представлява if конструкцията?

Условни конструкции използваме, когато искаме блок от код да се изпълни, само когато дадено условие е изпълнено. Има три вида конструкции:

1) If конструкция

```
if (condition)
{
    //block of code
}
```

Условието в скобите се проверява и ако е вярно се изпълнява кода в тялото на конструкцията, ако не програмата продължава изпълнението си.

Пример:

```
1  #include <iostream>
2
3  int main()
4  {
5      int number = 2;
6
7      if (number % 2 == 0)
8      {
9          std::cout << "Number is even!\n";
10     }
11
12     return 0;
13 }
```

2) If-else конструкция

```
if (condition)
{
    // this block of code will execute if condition is true
}
else
{
    // this block of code will execute if condition is false
}
```

Пример:

```
1      #include <iostream>
2
3      int main()
4      {
5          int number;
6
7          std::cin >> number;
8
9          if (number % 2 == 0)
10         {
11             std::cout << "Number is even!\n";
12         }
13         else
14         {
15             std::cout << "Number is odd!\n";
16         }
17
18         return 0;
19     }
```

3) If-else if-else конструкция

```
if (/* condition1 */)
{
    // this block of code will execute if condition is true
}
else if (/* condition2 */)
{
    // this block of code will execute if the condition2 is true
}
else
{
    // this block of code will execute if condition1 and condition2 are false
}
```

Пример:

```
1  #include <iostream>
2
3  int main()
4  {
5      int number;
6
7      std::cin >> number;
8
9      if (number > 0)
10     {
11         std::cout << "Number is positive!\n";
12     }
13     else if (number < 0)
14     {
15         std::cout << "Number is negative!\n";
16     }
17     else
18     {
19         std::cout << "Number is zero!\n";
20     }
21
22     return 0;
23 }
```

2. Switch конструкция

1) Какво представлява switch?

Switch ни позволява да изпълним даден блок от код, чийто случай съвпада с условието в скобите. При съвпадение се изпълняват дадените операции докато не се срещне оператора **break**.

```
switch (/*expression*/)
{
case /* constant-expression1 */:
    /* this code will execute if constant-expression1 is equal to expression */
    break;
case /* constant-expression2 */:
    /* this code will execute if constant-expression2 is equal to expression */
    break;
    .
    .
default:
    /* code to be executed if none of the above cases match the expression */
    break;
}
```

Пример:

```
1      #include <iostream>
2
3      int main()
4      {
5          int dayOfWeek;
6
7          std::cin >> dayOfWeek;
8
9          switch (dayOfWeek)
10         {
11             case 1:
12                 std::cout << "It's Monday!\n";
13                 break;
14             case 2:
15                 std::cout << "It's Tuesday!\n";
16                 break;
17             case 3:
18                 std::cout << "It's Wednesday!\n";
19                 break;
20             case 4:
21                 std::cout << "It's Thursday!\n";
22                 break;
23             case 5:
24                 std::cout << "It's Friday!\n";
25                 break;
26             case 6:
27                 std::cout << "It's Saturday!\n";
28                 break;
29             case 7:
30                 std::cout << "It's Sunday!\n";
31                 break;
32             default:
33                 std::cout << "Wrong day of week!\n";
34                 break;
35         }
36
37         return 0;
38     }
```

3. Тернарен оператор

Синтаксис:

```
<condition> ? <expression1> : <expression2>
```

Ако условието е вярно, се изпълнява първия израз(expression1), ако е грешно се изпълнява втория израз(expression2).

Пример:

```
1      #include <iostream>
2
3      int main()
4      {
5          int age;
6          std::cin >> age;
7
8          std::cout << (age >= 18 ? "Adult\n" : "Minor\n");
9
10         return 0;
11     }
```