**Simple Conditional Statements – Material Summary**

* **Оператори за сравнение:** ==, !=, >=, >, <=, <
* **Сравняване на текстове:** 
  + Сравнява се дължината
  + Сравнява се съдържанието
* **Булев тип данни:** bool
* **Булева променлива**
  + **bool isValid = true;**
  + **bool isCorrect = false;**
  + **bool isGreater = 5 < 9;**
  + **bool isHigher = 10 > 5;**
* **Булев израз:** 
  + израз, в който има оператор за сравнение
  + може да има две стойности: **true** или **false**
* **Блок от код:** обособява се с отворена къдрава скоба **{** и затворена къдрава скоба **}**
* **Условна конструкция:** if – else

if (булев израз)

{

//код за изпълнение ако булевия израз е true

}

else

{

//код за изпълнение ако булевия израз е false

}

* **Серия от проверки:** if – else if – else

if (булев израз)

{

//код за изпълнение ако булевия израз е true

}

else if (булев израз)

{

//код за изпълнение ако горния булев израз е false

}

else

{

//код за изпълнение ако всички булеви изрази са false

}

* **Живот на променлива:** 
  + Променливата се създава в даден блок от код, когато и се зададе име, тип и стойност
  + Променливата може да се използва в блока от код, в който е създадена и във всички вътрешни блокове от код
  + Променливата не може да се използва извън блока от код, в който е създадена
* **Дебъгване:** проследяване на изпълнението на кода ред по ред с цел проследяване на логиката и отстраняване на грешки
  + Поставяме breakpoint (поставя се в сивото поле) на реда, от който искаме да започнем дебъгването
  + Стартираме процесът по дебъгване (F5)
  + В жълто е оцветен редът, който предстои да се изпълни
  + Изпълняване на реда, който предстои да се изпълни с F10