7 COMPUTER PROGRAMMING

CONTROL Shift

CONTENTS

□ CONTROL shift

- ■ BREAK
- ■ BREAK LABEL
- ■ CONTINUE
- ■ CONTINUE LABEL
- ■ RETURN

Control Shift - BREAK statement

- □ **Provides BREAK, CONTINUE, RETURN commands to shift the control.**
- □ **BREAK command**
  - ■ Apply to a switch statement outside in the switch statement.
  - ■ Apply to a iteration statement outside in the iteration loop.
  - ■ Advanced format of the conventional goto statement

```
class ForBreak {
  public static void main(String args[]) {
    for(int i=0; i<100; i++) {
      if(i == 10) break;
      System.out.println(i + "자바의 세계로 오세요! " );
    }
    System.out.println("Break에 의하여 for 문이 중단되었습니다.");
  }
}
```

```
class WhileBreak {
  public static void main(String args[]) {
    int i = 0;
    while(i < 100) {
      if(i == 10) break;
      System.out.println(i + "자바의 세계로 오세요! " );
      i++;
    }
  System.out.println("Break에 의하여 While 문이 중단되었습니다.");
  }
}
```

Control Shift - BREAK statement

```
class NestedForBreak {
  public static void main(String args[
]) {
    int i, j;
    for(i=1 ; i<10 ; i++) {
      for(j=1 ; j<i ; j++)  {
        if (j > 6) break;
          System.out.print(" * ");
      }
      System.out.println();
    }
  }
}
```

Control Shift - BREAK LABEL statement

□ **Only limited use of nested state blocks**

```
class BreakTest {
  public static void main(String args[]) {
     boolean t = true;
     first : {
        second : {
             third : {
                    System.out.println("third 블록 'break' 문장전");
                    if(t) break second;
                    System.out.println("third 블록 'break' 문장후");
              }
            System.out.println("second 블록 문장");
        }
      System.out.println("first 블록 문장");
     }
  }
}
```

Control Shift – BREAK LABEL statement

```
class BreakErr {
  public static void main(String args[]) {
     one : for(int i=0; i<3; i++)  {
       System.out.print("Pass " + i + ": ");
     }
     for(int j=0; j<100; j++)  {
          if(j == 10) break one;
          System.out.print(j + " ");
     }
  }
}
```

Control Shift - CONTINUE statement

- □ used to send the control to the first statement at a particular position of iteration statements

```
class ContinueTest {
  public static void main(String args[]) {
    for(int i=0; i<10; i++) {
      if (i%2 == 0) continue;
      System.out.println(i + " 자바의 세계로 오세요! " );
    }
  }
}
```

Control Shift - CONTINUE LABEL statement

□ In the nested FOR loop, move the control to the specified label.

```
class ContinueLabelTest {
  public static void main(String args[]) {
    outer : for (int i=0; i<10; i++) {
                  for(int j=0; j<10; j++)  {
                     if(j > i)  {
                        System.out.println();
                        continue outer;
                     }
                  System.out.print(“ “ + (i*j));
                 }
              }
   }
}
```

Control Shift – RETURN statement

□ **Return control to the position to call the method**

```
class Return {
  public static void main(String args[ ]) {
    boolean t = true;
    System.out.println("Hot 자바!");
    if(t) return;
        System.out.println("자바를 배워봅시다.");
  }
}
```

CONCLUDE

□ CONTROL shift

- ■ BREAK
- ■ BREAK LABEL
- ■ CONTINUE
- ■ CONTINUE LABEL
- ■ RETURN