

Korišćenje naredbi uslova

Ivan Milić 1099/2014

Kondicionalni

- ▶ Kondicionali odnosno naredbe uslova (grananja) predstavljaju naredbe koje kontrolišu izvršavanje drugih naredbi
- ▶ *if (else)*
- ▶ *switch (case)*

if naredbe

- ▶ Postoje različite vrste *if* naredbi:
- ▶ *if* odnosno *if-then* naredba
- ▶ *if-then-else*
- ▶ *if-then-else-if*

Obična *if-then* naredba

- ▶ Kada pišemo *if* naredbu, treba da pratimo sledeće smernice:
 1. Utvrditi korektnost grananja prema jednakosti ($>$, $>=$, $<$, $<=$)
 2. Pisati prvo regularnu putanju kroz kod; nakon toga neuobičajene slučajeve
 3. Staviti normalni slučaj izvršavanja nakon *if*, a ne nakon *else* naredbe

```
OpenFile( inputFile, status )
If ( status = Status_Error ) Then
  error case   |   errorType = FileOpenError
Else
  nominal case |   ReadFile( inputFile, fileData, status )
                |   If ( status = Status_Success ) Then
  nominal case |     SummarizeFileData( fileData, summaryData, status )
                |     If ( status = Status_Error ) Then
  error case   |       errorType = ErrorType_DataSummaryError
                |     Else
  nominal case |       PrintSummary( summaryData )
                |       SaveSummaryData( summaryData, status )
                |       If ( status = Status_Error ) Then
  error case   |         errorType = ErrorType_SummarySaveError
                |       Else
  nominal case |         UpdateAllAccounts()
                |         EraseUndoFile()
                |         errorType = ErrorType_None
                |       End If
                |     End If
                |   Else
                |     errorType = ErrorType_FileReadError
                |   End If
End If
```

```
OpenFile( inputFile, status )
If status = Status_Success Then
  nominal case ReadFile( inputFile, fileData, status )
  If status = Status_Success Then
    nominal case SummarizeFileData( fileData, summaryData, status )
    If status = Status_Success Then
      nominal case PrintSummary( summaryData )
      SaveSummaryData( summaryData, status )
      If status = Status_Success Then
        nominal case UpdateAllAccounts()
        EraseUndoFile()
        errorType = ErrorType_None
      Else
        error case errorType = ErrorType_SummarySaveError
      End If
    Else
      error case errorType = ErrorType_DataSummaryError
    End If
  Else
    error case errorType = ErrorType_FileReadError
  End If
Else
  error case errorType = ErrorType_FileOpenError
End If
```

Smernice za *if-then* (nastavak)

4. Ispratiti *if* klauzu sa naredbom koja ima značenje

```
if ( SomeTest )  
    ;  
else {  
    // do something  
    ...  
}
```

```
if ( ! SomeTest ) {  
    // do something  
    ...  
}
```

5. Uzeti u obzir i *else* klauzu

```
// if color is valid  
if ( COLOR_MIN <= color && color <= COLOR_MAX ) {  
    // do something  
    ...  
}  
else {  
    // else color is invalid  
    // screen not written to -- safely ignore command  
}
```

Smernice za *if-then* (nastavak)

- 6. Testirati korektnost *else* klauze (a ne samo *if*)
- 7. Proveriti da li je došlo do obrtanja *if* i *else* klauza (česta greška)

Lanac *if-then-else* naredbi

```
if ( inputCharacter < SPACE ) {
    characterType = CharacterType_ControlCharacter;
}
else if (
    inputCharacter == ' ' ||
    inputCharacter == ',' ||
    inputCharacter == '.' ||
    inputCharacter == '!' ||
    inputCharacter == '(' ||
    inputCharacter == ')' ||
    inputCharacter == ':' ||
    inputCharacter == ';' ||
    inputCharacter == '?' ||
    inputCharacter == '-'
    ) {
    characterType = CharacterType_Punctuation;
}
else if ( '0' <= inputCharacter && inputCharacter <= '9' ) {
    characterType = CharacterType_Digit;
}
else if (
    ( 'a' <= inputCharacter && inputCharacter <= 'z' ) ||
    ( 'A' <= inputCharacter && inputCharacter <= 'Z' )
    ) {
    characterType = CharacterType_Letter;
}
```

Smernice za *if-then-else*

1. Uprostiti komplikovane testove sa pozivima *boolean* funkcija

```
if ( IsControl( inputCharacter ) ) {  
    characterType = CharacterType_ControlCharacter;  
}  
else if ( IsPunctuation( inputCharacter ) ) {  
    characterType = CharacterType_Punctuation;  
}  
else if ( IsDigit( inputCharacter ) ) {  
    characterType = CharacterType_Digit;  
}  
else if ( IsLetter( inputCharacter ) ) {  
    characterType = CharacterType_Letter;  
}
```

Smernice za *if-then-else* (nastavak)

2. Staviti prvo najčešće slučajeve

```
if ( IsLetter( inputCharacter ) ) {  
    characterType = CharacterType_Letter;  
}  
else if ( IsPunctuation( inputCharacter ) ) {  
    characterType = CharacterType_Punctuation;  
}  
else if ( IsDigit( inputCharacter ) ) {  
    characterType = CharacterType_Digit;  
}  
else if ( IsControl( inputCharacter ) ) {  
    characterType = CharacterType_ControlCharacter;  
}
```

Smernice za *if-then-else* (nastavak)

3. Potruditi se da svi slučajevi budu pokriveni

```
if ( IsLetter( inputCharacter ) ) {  
    characterType = CharacterType_Letter;  
}  
else if ( IsPunctuation( inputCharacter ) ) {  
    characterType = CharacterType_Punctuation;  
}  
else if ( IsDigit( inputCharacter ) ) {  
    characterType = CharacterType_Digit;  
}  
else if ( IsControl( inputCharacter ) ) {  
    characterType = CharacterType_ControlCharacter;  
}  
else {  
    DisplayInternalError( "Unexpected type of character detected." );  
}
```

Smernice za *if-then-else* (nastavak)

4. Zameniti *if-then-else* lance sa drugim konstruktima ako ih vaš jezik podržava

```
Select Case inputCharacter
  Case "a" To "z"
    characterType = CharacterType_Letter
  Case " ", ",", ".", "!", "(", ")", ":", ";", "?", "-"
    characterType = CharacterType_Punctuation
  Case "0" To "9"
    characterType = CharacterType_Digit
  Case FIRST_CONTROL_CHARACTER To LAST_CONTROL_CHARACTER
    characterType = CharacterType_Control
  Case Else
    DisplayInternalError( "Unexpected type of character detected." )
End Select
```

case naredbe

- ▶ *case* (odnosno *switch*) naredba je konstrukt koji veoma varira od jezika do jezika
- ▶ *Visual Basic* podržava *case* i pruža notaciju za izražavanje opsega i kombinacija vrednosti
- ▶ *C++* i *Java* podržavaju *case* samo za po jednu vrednost
- ▶ Mnogi skript jezici uopšte ne podržavaju *case* i stoga se moraju koristiti *if-then-else-if* lanci

Smernice za *case*

Odabir najefektivnijeg redosleda slučajeva

1. Složiti slučajeve po abecedi ili numerički (svi su podjednako važni)
2. Staviti prvo normalne slučajeve
3. Složiti slučajeve po učestalosti

Smernice za *case* (nastavak)

Saveti za korišćenje *case* naredbi

1. Ne praviti veštačke promenljive kako bi se iskoristila *case* naredba

```
action = userCommand[ 0 ];
switch ( action ) {
    case 'c':
        Copy();
        break;
    case 'd':
        DeleteCharacter();
        break;
    case 'f':
        Format();
        break;
    case 'h':
        Help();
        break;
    ...
    default:
        HandleUserInputError( ErrorType.InvalidUserCommand );
}
```

```
if ( UserCommand.equals( COMMAND_STRING_COPY ) ) {
    Copy();
}
else if ( UserCommand.equals( COMMAND_STRING_DELETE ) ) {
    DeleteCharacter();
}
else if ( UserCommand.equals( COMMAND_STRING_FORMAT ) ) {
    Format();
}
else if ( UserCommand.equals( COMMAND_STRING_HELP ) ) {
    Help();
}
...
else {
    HandleUserInputError( ErrorType.InvalidCommandInput );
}
```


Smernice za *case* (nastavak)

2. Održati jednostavnost akcija svakog slučaja
3. Kodirati poslednji preostali slučaj kao poseban slučaj (kodiranjem poslednjeg preostalog slučaja kao *default* slučaja, gubi se mogućnost detektovanja grešaka u *default* klauzi)
4. Koristiti *default* klauzu za detekciju grešaka

```
switch ( commandShortcutLetter ) {  
    case 'a':  
        PrintAnnualReport();  
        break;  
    case 'p':  
        // no action required, but case was considered  
        break;  
    case 'q':  
        PrintQuarterlyReport();  
        break;  
    case 's':  
        PrintSummaryReport();  
        break;  
    default:  
        DisplayInternalError( "Internal Error 905: Call customer support." );  
}
```

Smernice za *case* (nastavak)

5. Izbegavati propadanje kroz kraj *case* naredbe

```
switch ( InputVar )
{
    case 'A': if ( test )
        {
            // statement 1
            // statement 2
        }
    case 'B': // statement 3
        // statement 4
        ...
    }
    ...
    break;
    ...
}
```

Smernice za *case* (nastavak)

6. Identifikovati i jasno označiti mesta na kojima dolazi do propadanja na kraju *case* naredbe

```
switch ( errorDocumentationLevel ) {  
    case DocumentationLevel_Full:  
        DisplayErrorDetails( errorNumber );  
        // FALLTHROUGH -- Full documentation also prints summary comments  
  
    case DocumentationLevel_Summary:  
        DisplayErrorSummary( errorNumber );  
        // FALLTHROUGH -- Summary documentation also prints error number  
  
    case DocumentationLevel_NumberOnly:  
        DisplayErrorNumber( errorNumber );  
        break;  
  
    default:  
        DisplayInternalError( "Internal Error 905: Call customer support." );  
}
```

Ključne tačke

- ▶ Za jednostavne *if-else* naredbe, obratiti pažnju na redosled *if* i *else* klauza, pogotovo ako procesuiraju mnogo grešaka. Utvrditi da je put regularnog slučaja čist.
- ▶ Za *if-then-else* lance i *case* naredbe, izabrati redosled koji maksimizuje čitkost.
- ▶ Koristiti *default* klauzu u *case* naredbi ili poslednji *else* u lancu *if-then-else* naredbi za hvatanje grešaka.
- ▶ Izabrati konstrukt kontrole koji je najprikladniji svakoj pojedinačnoj sekciji koda.

Hvala na pažnji!