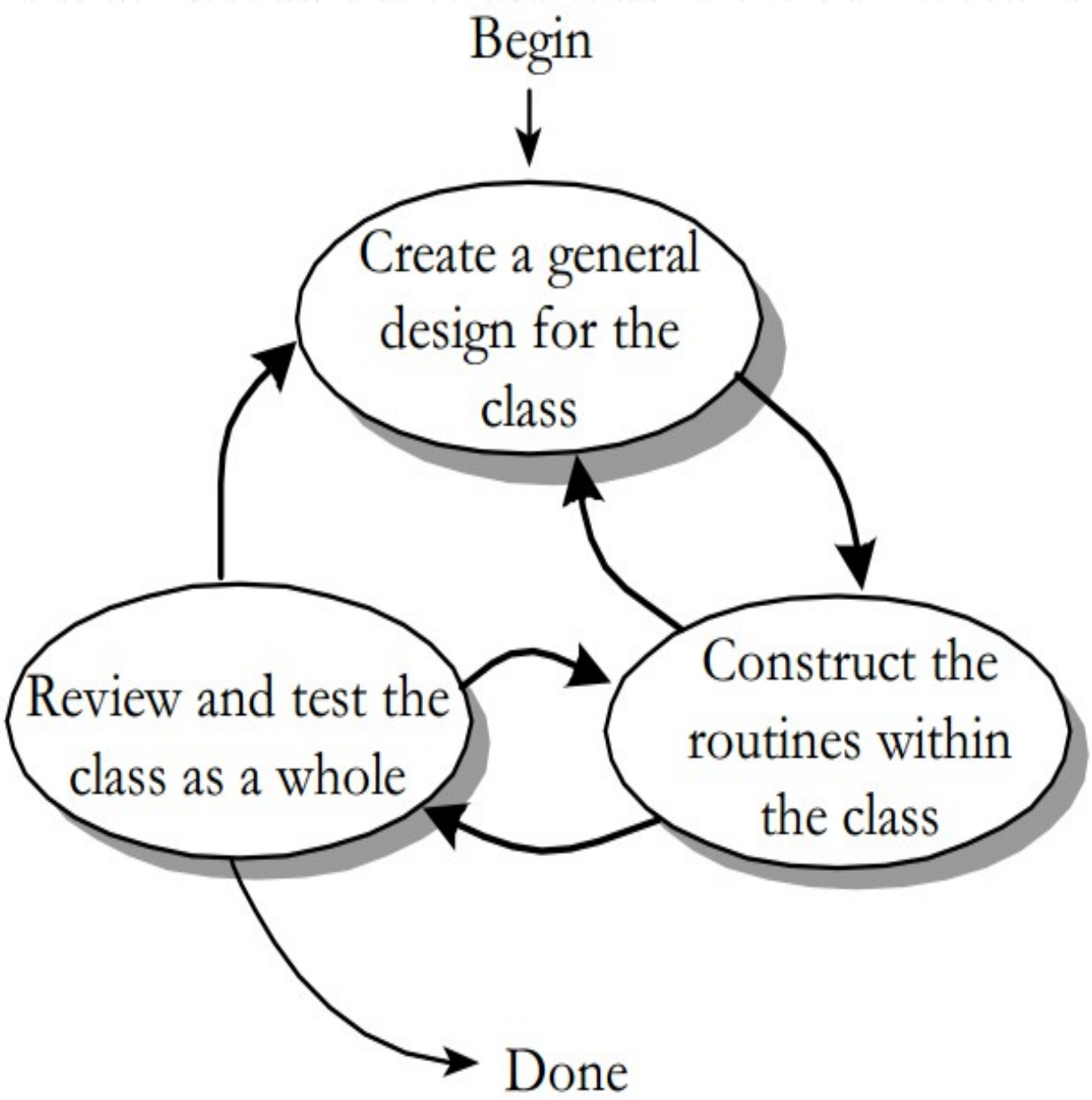
Pseudokod
Proces programiranja

Uvod

- Programiranje u malom
- Pseudocode programming process (PPP)
- Ekspert programeri preskacu ovo poglavlje
- Puna eksploatacija procesa
- PPP nije samo procedura za kreiranje klasa i rutina

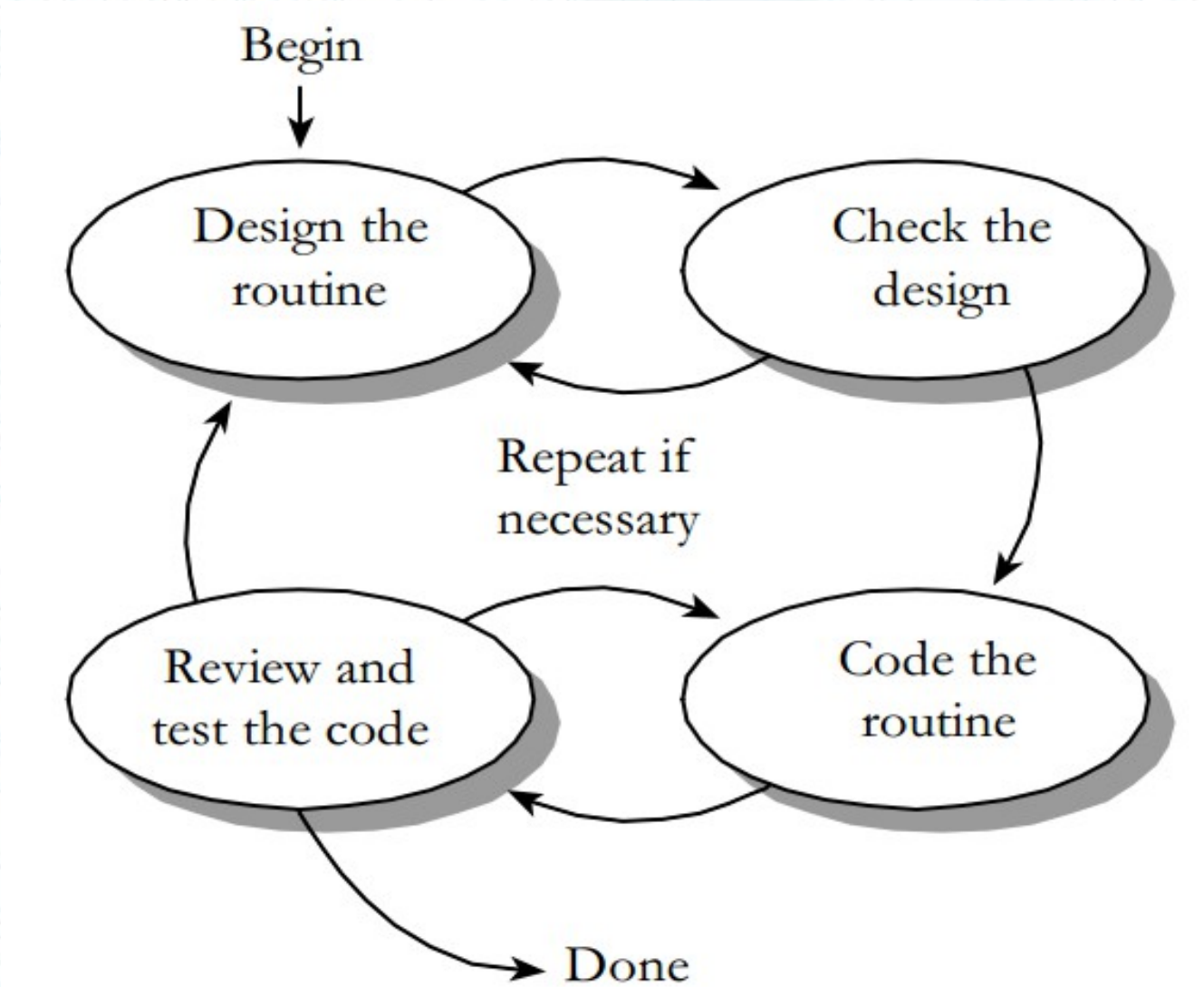
Rezime koraka u izgradnji klasa i rutina

- Obicno je u pitanju iterativni proces
- Kreiranje generalnog dizajna klase
- Enumerisanje i kreiranje specificnih rutina klase
- Provera konstrukcije klase kao celine


Begin
Create a general design for the class
Construct the routines within the class
Review and test the class as a whole
Done

Koraci u izgradnji rutina

- Mnoge rutine su jednostavne za implementaciju
- Za ostale, komplikovanije, potreban je sistematicni pristup
- Dizajn rutine
- Provera dizajna
- Kodiranje rutine
- Provera koda


Begin
Design the routine
Check the design
Repeat if necessary
Review and test the code
Code the routine
Done

Pseudocode for Pros

- Naredbe odgovaraju govornom jeziku (srpskom, engleskom...)
- Izbegavati sintaksne elemente ciljanog programskog jezika
- Pisati na nivou namere
- Dovoljno nizak nivo da bi generisanje koda bilo gotovo automatski

Primer loseg pseudokoda

```
increment resource number by 1

allocate a dlg struct using malloc

if malloc() returns NULL then return 1

invoke OSrsrc_init to initialize a resource for the operating system

*hRsrcPtr = resource number

return 0
```

Primer dobrog pseudokoda

```
Keep track of current number of resources in use
If another resource is available
   Allocate a dialog box structure
   If a dialog box structure could be allocated
      Note that one more resource is in use
      Initialize the resource
      Store the resource number at the location provided by the caller
   Endif
Endif
Return TRUE if a new resource was created; else return FALSE
```

Prednosti dobrog dizajna

- Lakse razmatranje
- Iterativno preciscavanje
- Lakse menjanje
- Manje komentarisanja
- Lakse odrzavanje od drugih formi

Konstrukcija rutina

- Dizajn rutine
- Kodiranje rutine
- Provera koda
- Ciscenje ostataka
- Ponoviti ako je potrebno

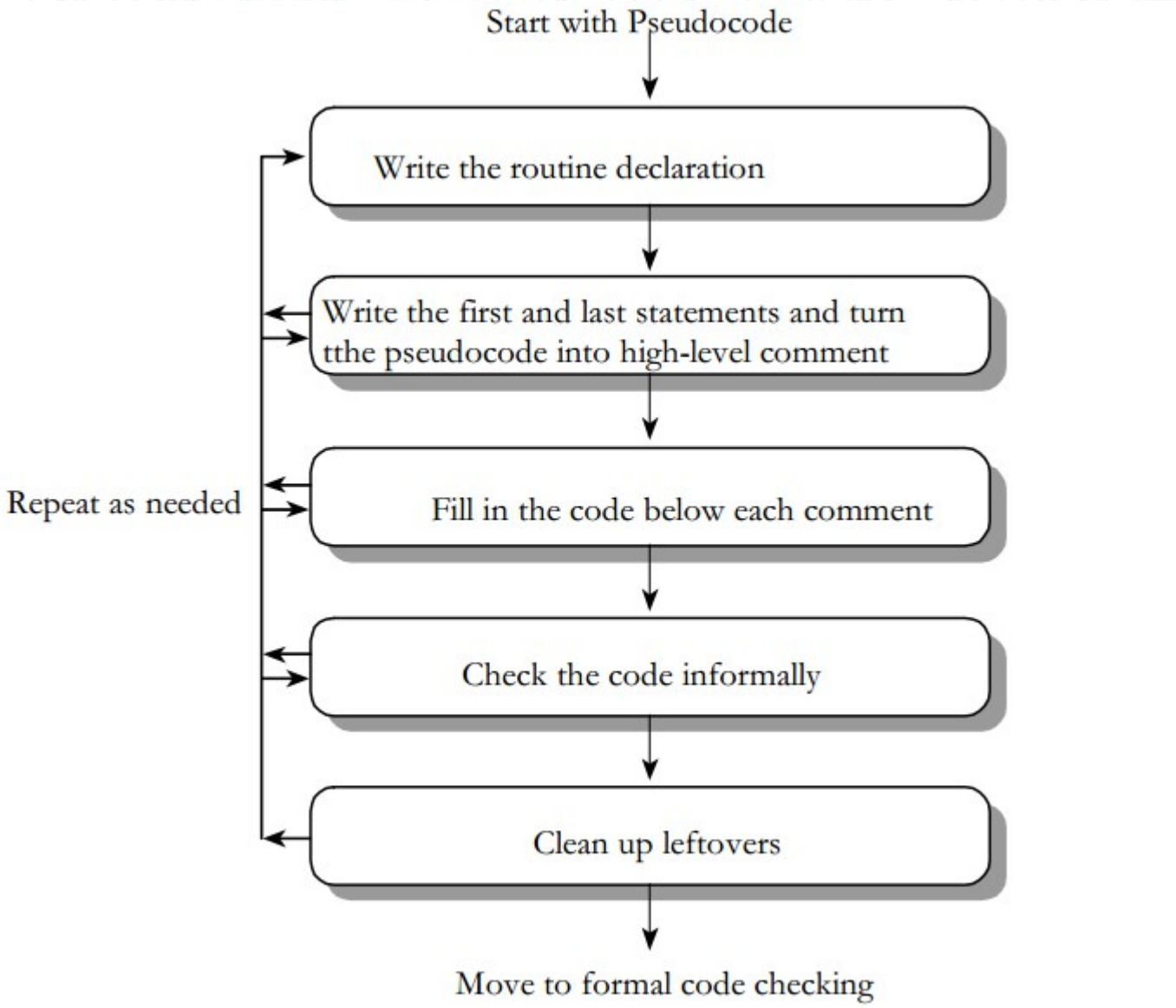
Dizajn rutine

- Proveriti preduslove
- Definisati problem koji rutina treba da resi
- Imenovati rutinu
- Odluka o testiranju rutine
- Rukovanje greskama
- Efikasnost
- Dostupna funkcionalnost u bibliotekama
- Istraziti algoritme i strukture podataka
- Napisati pseudokod

Dizajn rutine

- Razmisljati o podacima
- Proveriti pseudokod

Kodiranje rutine



Kodiranje rutine

```
/* This routine outputs an error message based on an error code
supplied by the calling routine. The way it outputs the message
depends on the current processing state, which it retrieves
on its own. It returns a value indicating success or failure.
*/

Status ReportErrorMessage(
   ErrorCode errorToReport
)
set the default status to "fail"
look up the message based on the error code

if the error code is valid
   if doing interactive processing, display the error message
   interactively and declare success

   if doing command line processing, log the error message to the
   command line and declare success

if the error code isn't valid, notify the user that an
internal error has been detected

return status information
```

Kodiranje rutine

```
/* This routine outputs an error message based on an error code
supplied by the calling routine. The way it outputs the message
depends on the current processing state, which it retrieves
on its own. It returns a value indicating success or failure. */

Status ReportErrorMessage(
   ErrorCode errorToReport
   ) {
   // set the default status to "fail"
   // look up the message based on the error code
   // if the error code is valid
      // if doing interactive processing, display the error message
      // interactively and declare success

      // if doing command line processing, log the error message to the
      // command line and declare success

   // if the error code isn't valid, notify the user that an
   // internal error has been detected

   // return status information
}
```

Kodiranje rutine

```
/* This routine outputs an error message based on an error code
supplied by the calling routine. The way it outputs the message
depends on the current processing state, which it retrieves
on its own. It returns a value indicating success or failure.
*/

Status ReportErrorMessage(
   ErrorCode errorToReport
   ) {
   // set the default status to "fail"
   Status errorMessageStatus = Status_Failure;

   // look up the message based on the error code
   Message errorMessage = LookupErrorMessage( errorToReport );

   // if the error code is valid
   if ( errorMessage.ValidCode() ) {
      // determine the processing method
      ProcessingMethod errorProcessingMethod = CurrentProcessingMethod();

      // if doing interactive processing, display the error message
```

Proveravanje koda

- “Mentalno” proveravanje
- Kompajliranje rutine
- Debagovanje
- Testiranje koda
- Uklanjanje gresaka

Ciscenje ostataka

- Provera interfejsa rutine
- Provera kvaliteta dizajna
- Provera podataka rutine
- Provera naredbi i logike rutine
- Provera layouta rutine
- Provera dokumentacije rutine
- Uklanjanje suvisnih komentara

Alternative PPP-u

- Test-first development
- Design by contract
- Hacking?

Checklist

- Da li su preduslovi zadovoljeni?
- Da li je problem koji treba resiti definisan?
- Da li je dizajn visokog nivoa dovoljno jasan da klase i rutine dobiju dobro ime?
- Da li ste razmisljali o testiranju klasa i rutina?
- Da li ste razmisljali o efikasnosti u smislu stabilnog interfejsa i citljive implementacije, ili u smislu zadovoljavanja rokova i budzeta?

Checklist

- Da li ste proverili standardne i druge biblioteke za vec gotov kod?
- Da li ste koristili literaturu za algoritme?
- Da li ste koristili detaljni pseudokod za svaku rutinu?
- Da li ste proverili “usmeno” pseudokod? Da li je lak za razumevanje?
- Da li ste obratili paznju na stvari koje bi mogli da vrate u dizajn(npr globalni podaci)?

Checklist

- Da li ste precizno preveli pseudokod u kod na ciljanom jeziku?
- Da li ste primenili PPP rekurzivno, razbijajuci rutine u manje ako je potrebno?
- Da li ste dokumentovali pretpostavke?
- Da li ste uklonili suvisne komentare?
- Da li ste razumeli kod? Da li je kod tezak za razumevanje?

KRAJ