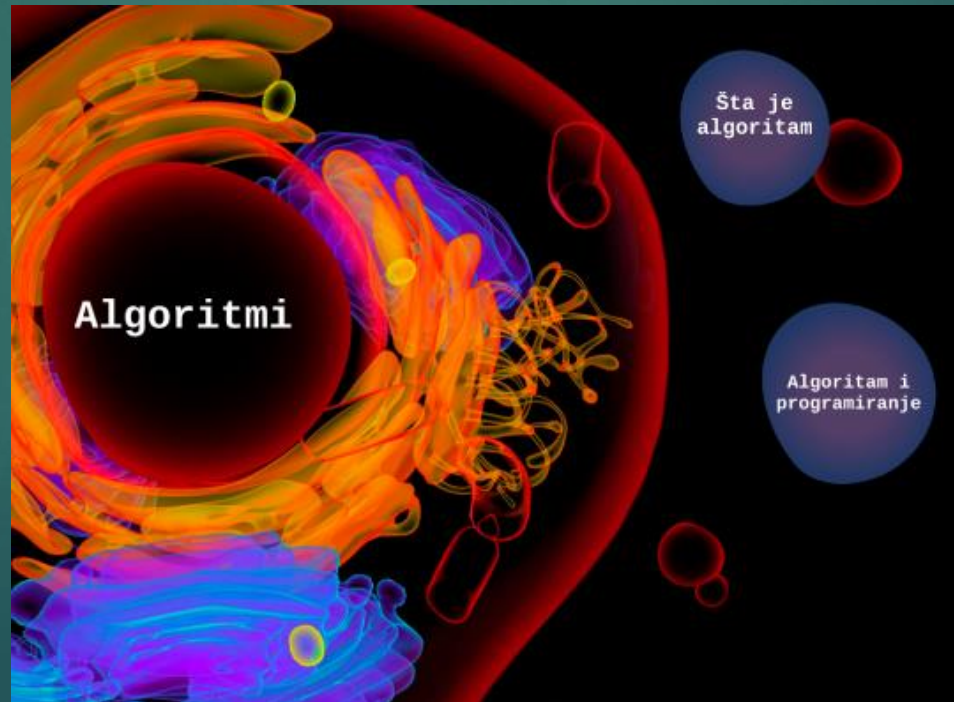


ЧАС 8

КЛАСИФИКАЦИЈА СТРУКТУРА АЛГОРИТАМА 2.



ЦИКЛИЧНЕ АЛГОРИТАМСКЕ СТРУКТУРЕ

- ▶ Општа карактеристика *цикличних алгоритамских структура* јесте вишеструко извршавање једног или више алгоритамских корака при једном извршавању алгоритма.
- ▶ Инструкције циклуса омогућавају да се одређена група инструкција систематски понавља док се не испуни услов за прекидање понављања.
- ▶ Циклична структура се још зове и *петља*, а услов за излазак из циклуса се зове *излазни критеријум*.
- ▶ Излазни критеријум је најчешће или број извршених циклуса или достигнута тачност у рачунању.

ОСНОВНЕ ВРСТЕ ЦИКЛУСА

- ▶ Основне врсте циклуса или петљи (енг. *loop*) су следеће:
 1. Петље са експлицитним бројачем,
 2. Петље са условним излазом,
 3. Петље са имплицитним бројачем,
 4. Бесконачне петље са и без механизма условног излаза,
 5. Петље по елементима скупа.
- ▶ Ми ћемо детаљно обрадити само прве две групе цикличних структура.

ГНЕЖЋЕЊЕ (ПЕТЉА У ПЕТЉИ)

- ▶ Циклуси могу да обухватају и друге циклусе, тзв. *гнежђење*.
- ▶ У том случају важи следеће основно правило за петље:
Спољна петља мора да обухвата све инструкције сваке унутрашње петље.

СПЕЦИФИЧНОСТ ЦИКЛИЧНЕ СТРУКТУРЕ

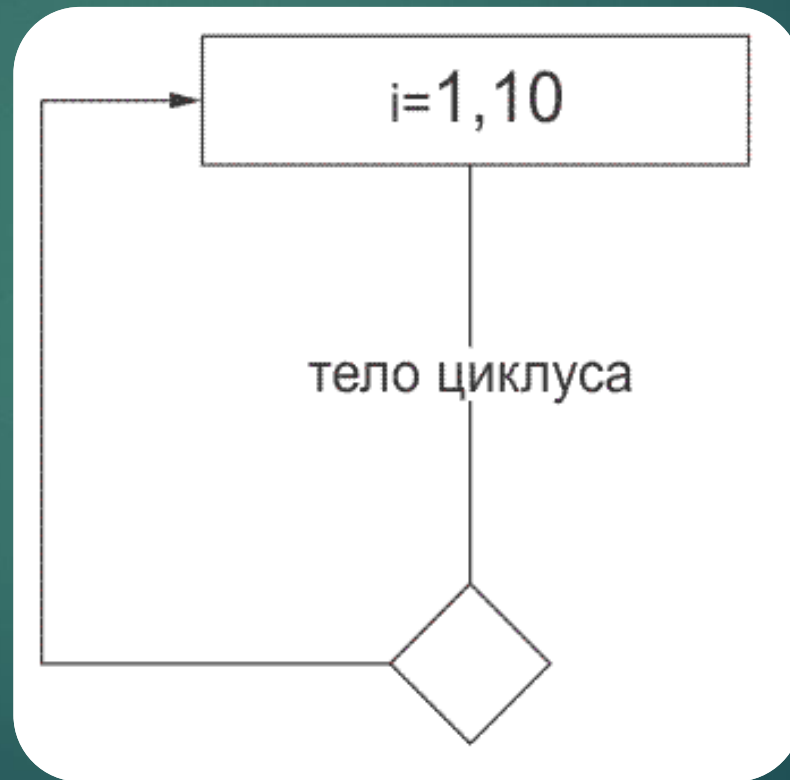
- ▶ Цикличне структуре се разликују од других структура по томе што се код ових структура део програма понавља више пута.
- ▶ Уколико се део програма који се понавља састоји од више од једног корака редослед истих не сме да се мења, већ увек остаје исти.
- ▶ Свака циклична структура има **почетак** и **крај циклуса**.
- ▶ Све између се назива **телом циклуса**.

БРОЈАЧКИ ЦИКЛУСИ

- ▶ Карактеристични алгоритми ове врсте су бројачки циклуси при чему бројање може бити унапред или уназад.
- ▶ *Унапред* – подразумева од почетне до крајње вредности са фиксним позитивним кораком, при чему је излазни критеријум податак да ли је достигнута или премашена крајња вредност.
- ▶ Бројање *уназад* подразумева кретање од крајње до почетне вредности са фиксним негативним кораком.

БРОЈАЧКИ ЦИКЛУСИ

- Бројачки циклуси се користе онда када знамо колико тачно пута се нешто понавља.



ПРИМЕР БРОЈАЧКОГ ЦИКЛУСА

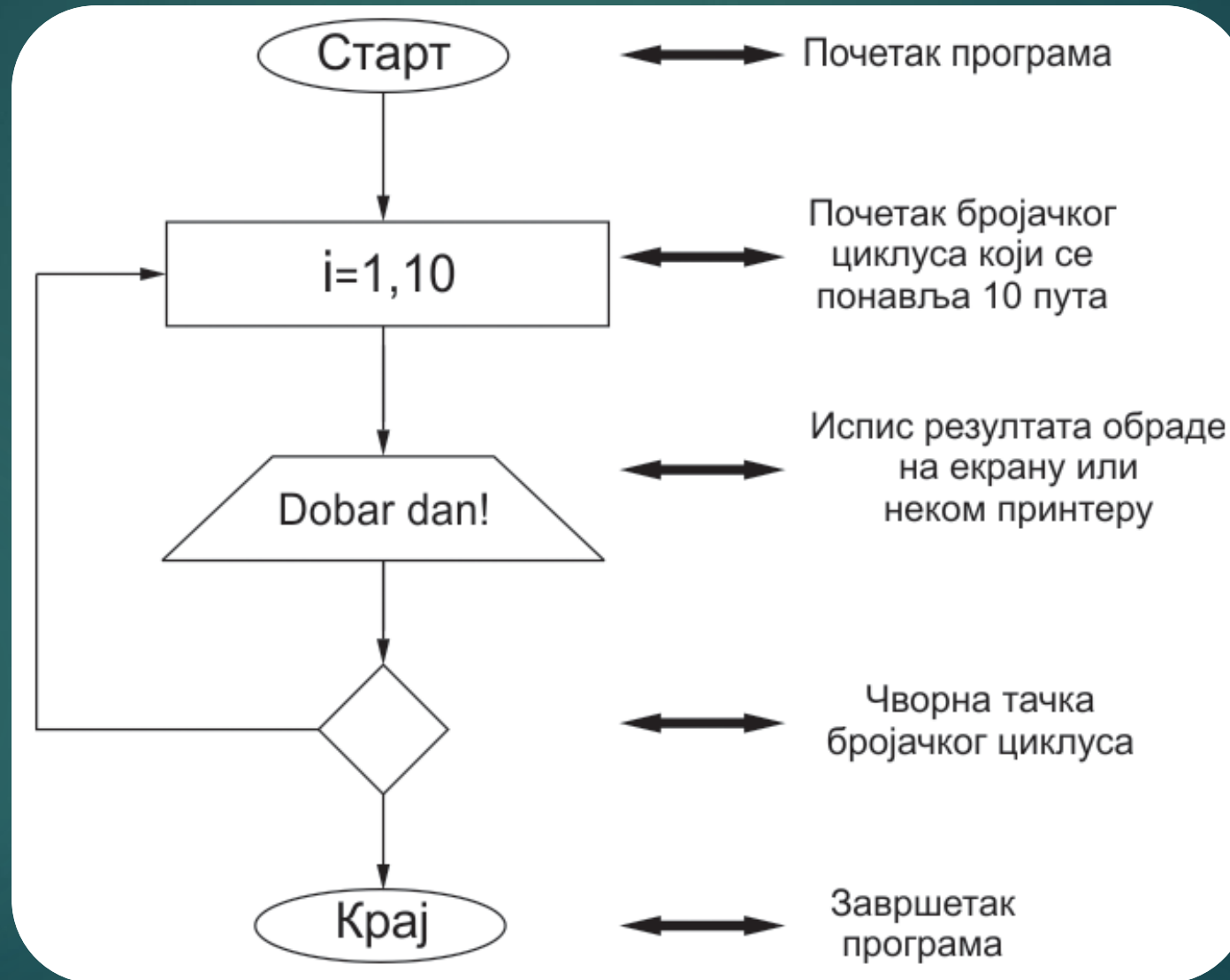
8

час

ДЕСЕТ ПУТА ОДШТАМПАТИ РЕЧЕНИЦУ:

„Добар дан.“

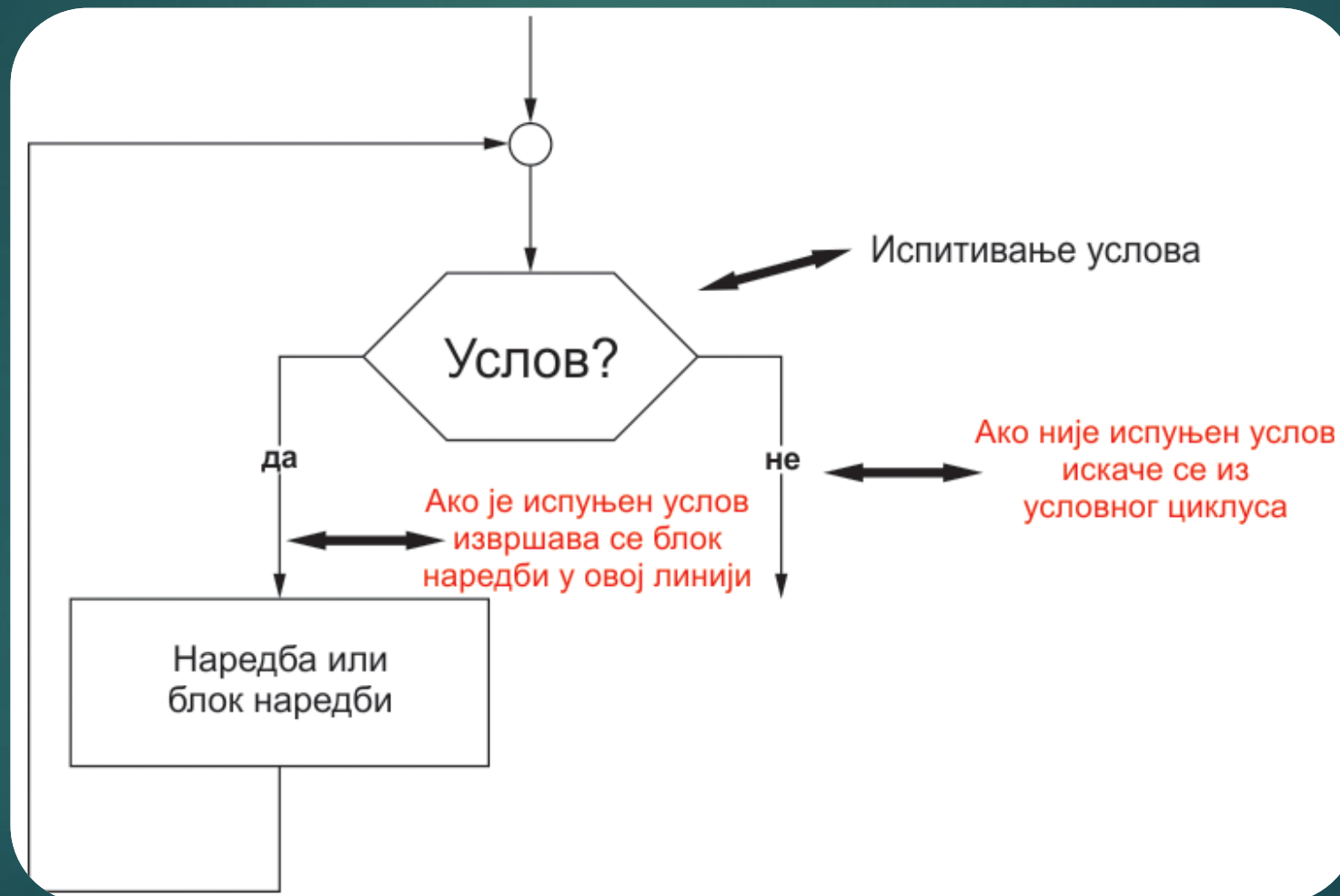
► Алгоритам за овај проблем ће имати следећи облик:



УСЛОВНИ ЦИКЛУСИ

- ▶ *Условни циклуси* се користе онда када не знамо колико пута ће нешто да се понови, већ постоји одређени услов који, када буде испуњен, значи и крај циклуса.
- ▶ Значајно је рећи да постоје *услови* који могу бити *на почетку циклуса*, али и *услови* који се налазе *на крају циклуса*.
- ▶ Онога тренутка када услов више није испуњен „искаче се“ из петље и наставља са извршењем задатка.

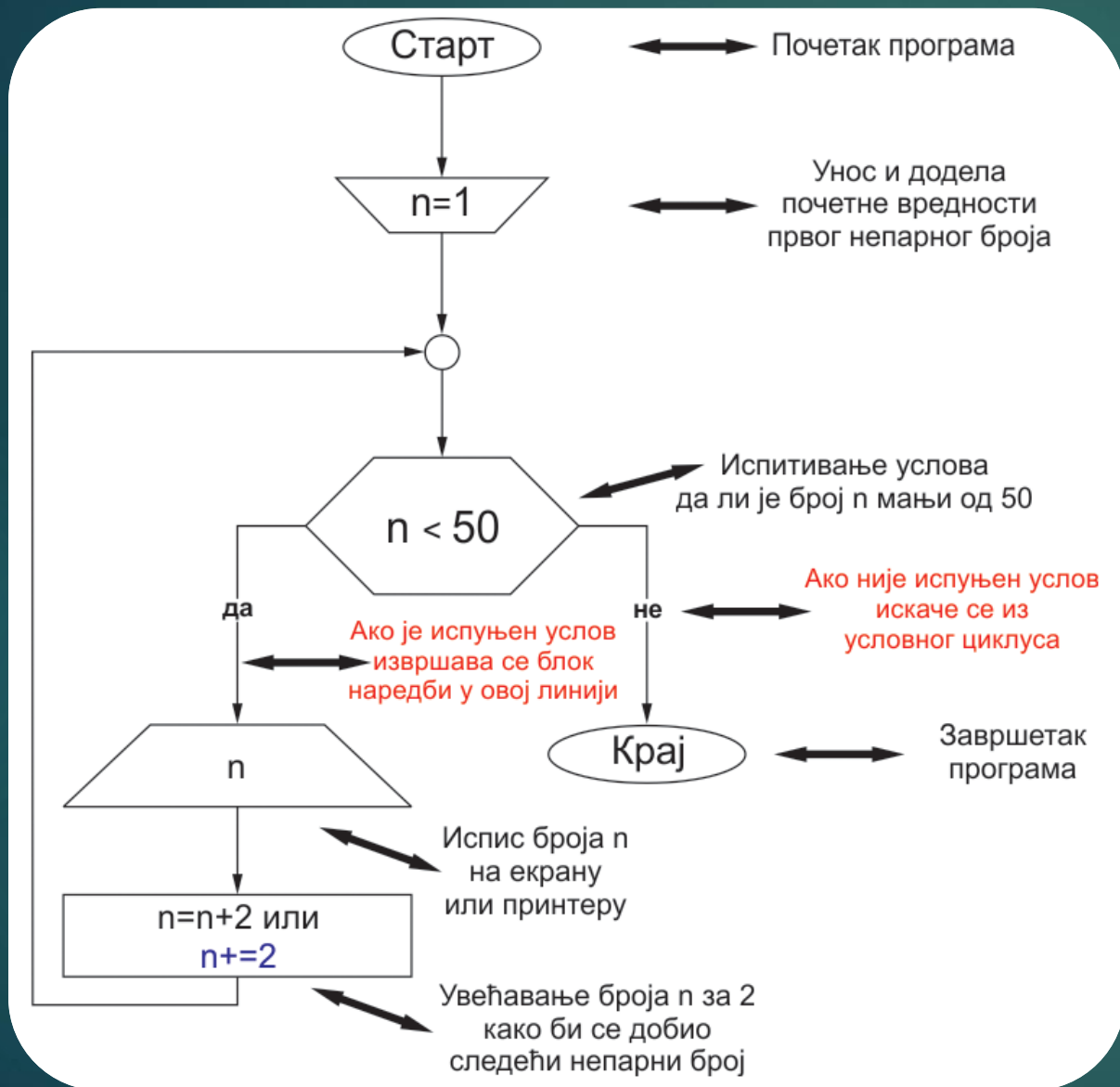
ЦИКЛИЧНА СТРУКТУРА СА УСЛОВОМ НА ПОЧЕТКУ ЦИКЛУСА



ПРИМЕР ЦИКЛИЧНЕ СТРУКТУРЕ СА УСЛОВОМ НА ПОЧЕТКУ ЦИКЛУСА

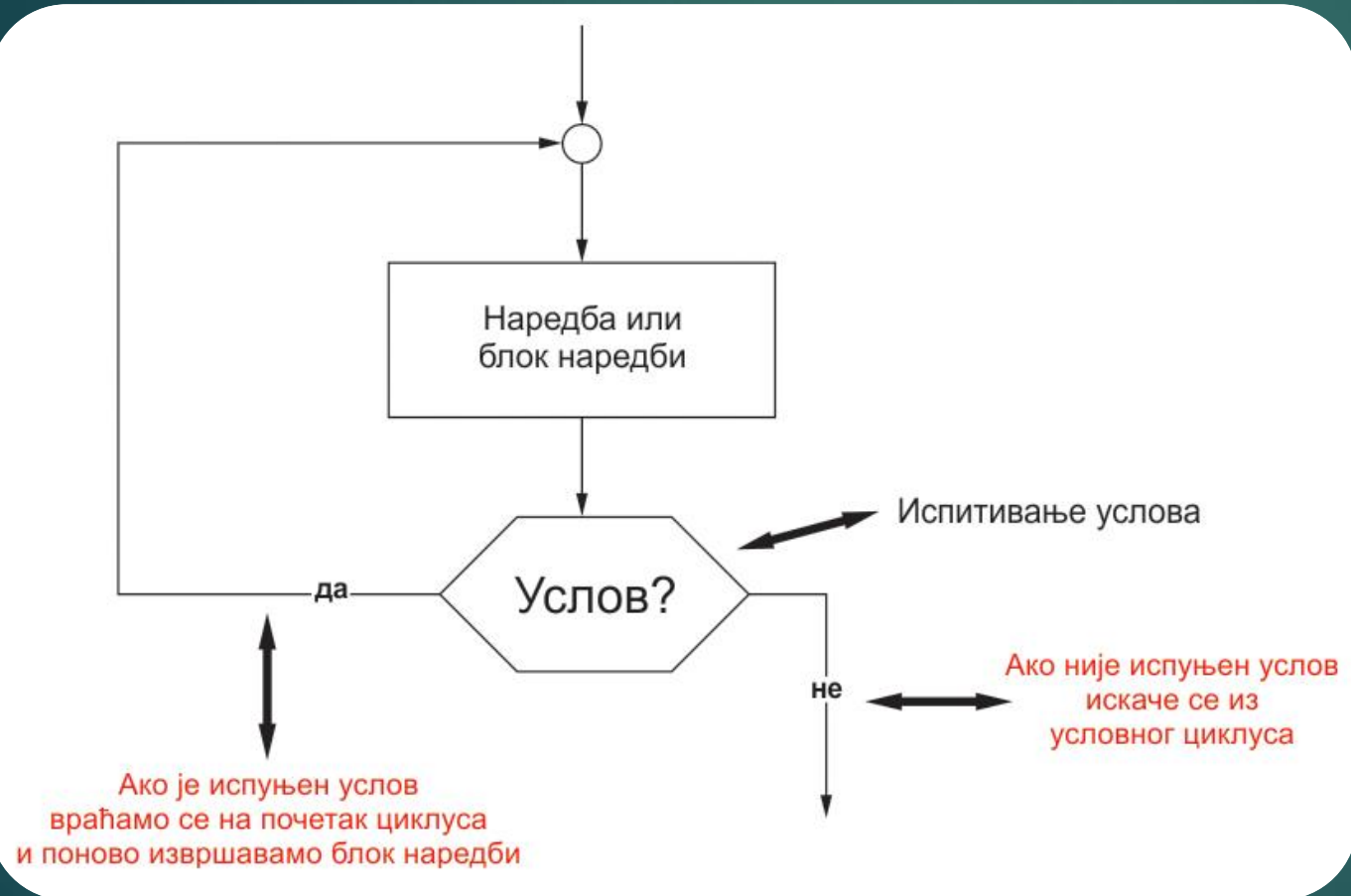
САСТАВИТИ АЛГОРИТАМ ПРОГРАМА
КОЈИ ЋЕ ШТАМПАТИ СВЕ НЕПАРНЕ
БРОЈЕВЕ МАЊЕ ОД 50.

► Алгоритам за овај проблем ће имати следећи облик:



► У обради су приказана два начина исписа увећања непознате за два: $n=n+2$ и $n+=2$. Оба начина су валидна, а чешће се примењује други испис (по обрнутој пољској нотацији).

ЦИКЛИЧНА СТРУКТУРА СА УСЛОВОМ НА КРАЈУ ЦИКЛУСА



➤ Уколико се наредбе најпре изврше, а потом испитује услов који означава прекид петље реч је о **условном циклусу са условом на крају циклуса**.

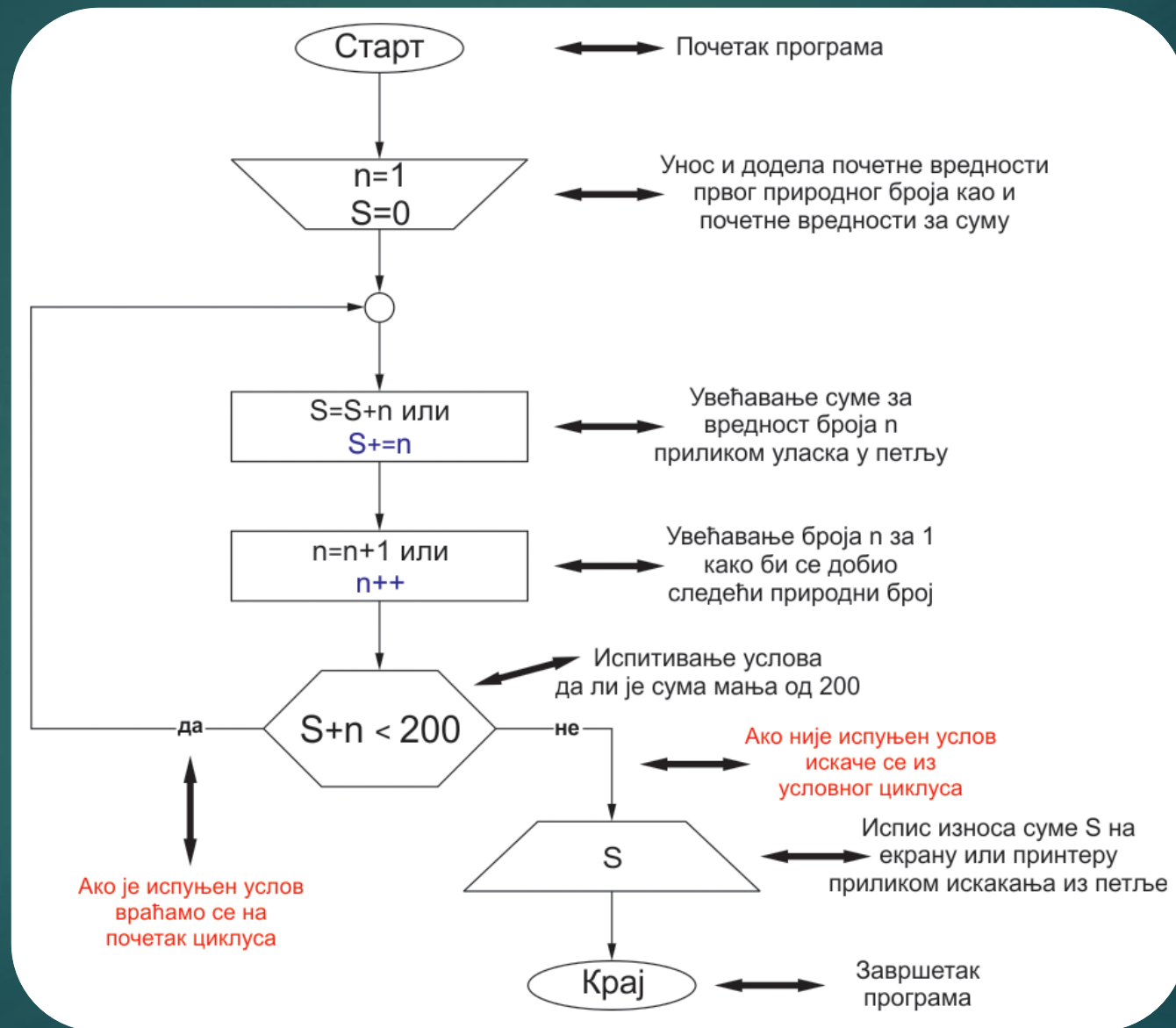
и поново извршавамо блок наредби
враћамо се на почетак циклуса
Ако је испуњен услов

условног циклуса

ПРИМЕР ЦИКЛИЧНЕ СТРУКТУРЕ СА УСЛОВОМ НА КРАЈУ ЦИКЛУСА

САСТАВИТИ АЛГОРИТАМ ПРОГРАМА
КОЈИ ЋЕ НАПРАВИТИ ЗБИР ПРИРОДНИХ
БРОЈЕВА ПО РЕДОСЛЕДУ СВЕ ДОК ЈЕ
ЗБИР МАЊИ ОД 200.

► Алгоритам за овај проблем ће имати следећи облик:



- У овом примеру смо видели да морамо да доделимо почетну вредност суми пре уласка у петљу. Та почетна вредност је увек $S=0$.
- Почетна вредност за број n је 1 пошто је 1 први природни број!
- У петљи се сума увећава за вредност броја n коју он има приликом уласка у петљу (у првом пролазу је то вредност 1 , у другом 1 увећано за $1 = 2, \dots$)
- Такође, видели смо и два начина исписа за увећавање за 1 ($n=n+1$ или $n++$).