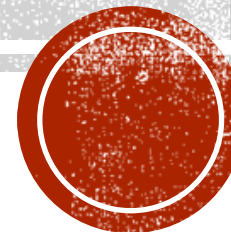


ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ ПРОГРАМСКИ ЈЕЗИК ЈАВА – 1

Решавање проблема помоћу рачунара



СОФТВЕР

- Скуп програма рачунара.
- Креирање софтвера је сложен процес.
- Софтверско инжењерство.
- Захтеви:
 - ефикасност,
 - поузданост,
 - користност,
 - изменљивост,
 - преносивост
 - итд.

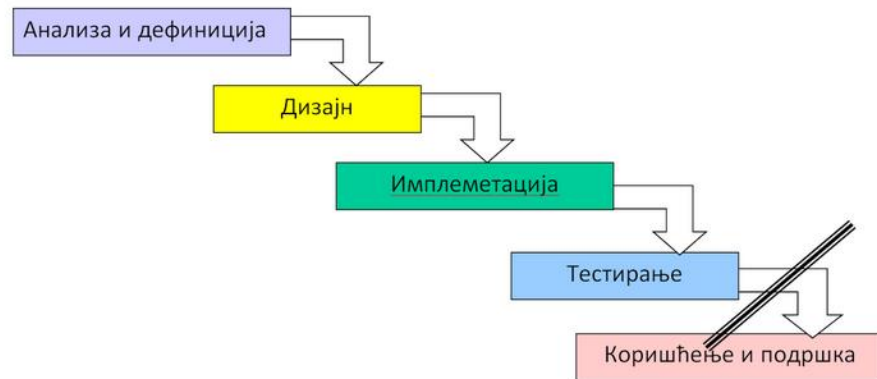
РЕШАВАЊЕ ПРОБЛЕМА

МОДЕЛ ВОДОПАДА

- Један од најстаријих и најчешће коришћених модела.
- Четири фазе:
 1. анализа и дефиниција проблема,
 2. дизајн,
 3. имплементација (кóдирање) и
 4. тестирање.
- Додатна фаза коришћења и подршке.

РЕШАВАЊЕ ПРОБЛЕМА (2)

МОДЕЛ ВОДОПАДА



ФАЗА АНАЛИЗЕ И ДЕФИНИЦИЈЕ ПРОБЛЕМА МОДЕЛ ВОДОПАДА

- Захтеви које мора задовољавати новокреирани софтвер.
- Без прецизних захтева производ може да „промаши“ очекивања корисника.
- Две подфазе:
 1. планирање:
 - студија изводљивости,
 - речник термина,
 - процена цене
 - прелиминарна спецификација
 - израда пројектног плана
 2. дефинисање:
 - детаљна спецификација захтева,
 - дефинисање корисничког интерфејса,
 - упутство за кориснике.

ФАЗА ДИЗАЈНА МОДЕЛ ВОДОПАДА

- Пројектовање (моделовање) будућег софтверског производа.
 - Језик за моделирање UML (енг. Unified Modeling Language).
 - Псеудојезици, дијаграми итд.
- Битне одлуке:
 - одабир метода пројектовања,
 - вођење рачуна о архитектури софтвера (слојевита, клијент-сервер, веб-сервиси итд.)
- Мора бити у складу са претходном фазом.

ФАЗА ИМПЛЕМЕНТАЦИЈЕ

МОДЕЛ ВОДОПАДА

- Писање програма коришћењем неког (неких) програмских језика.
- Вођена моделом из претходне фазе.
- Рутинска за искусног програмера или програмере.
- Повезивање модула у различитим програмским језицима (ако се користи више).

ФАЗА ТЕСТИРАЊА

МОДЕЛ ВОДОПАДА

- Да ли софтвер испуњава раније постављене захтеве.
- Примена софтвера и упоређивање резултата са очекиваним (тачним).
 - Тест примери.
- Одабир карактеристичних улаза и сценарија употребе је кључан.
- Тестери.
- У случају грешака, следи корекција софтвера (фаза имплементације).
 - Водопадни модел не предвиђа ову могућност.
 - Постоје и други модели, водопадни је традиционалан и најједноставнији.
- Друге технике:
 - формална верификација софтвера,
 - комбиновани приступ (верификација + тестирање).

ФАЗА КОРИШЋЕЊА И ПОДРШКЕ

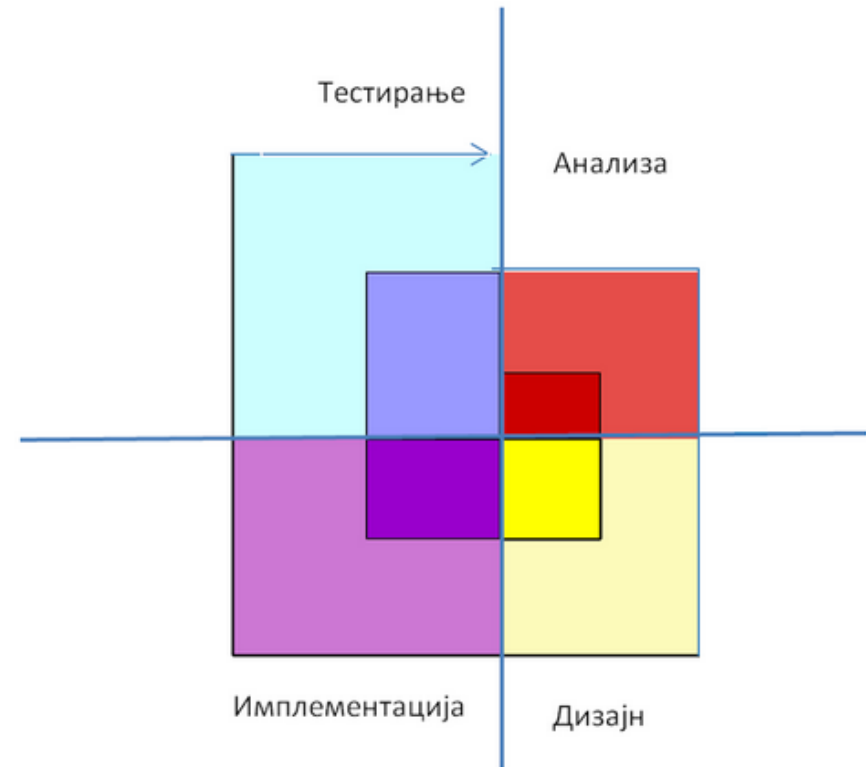
МОДЕЛ ВОДОПАДА

- Након успешног тестирања.
- Испоручивање крајњим корисницима.
 - Софтвер.
 - Документација за различите типове корисника.
- Измене у току коришћења (одржавање):
 - прилагођавања,
 - отклањање неочекиваних грешака.

РЕШАВАЊЕ ПРОБЛЕМА

СПИРАЛНИ МОДЕЛ

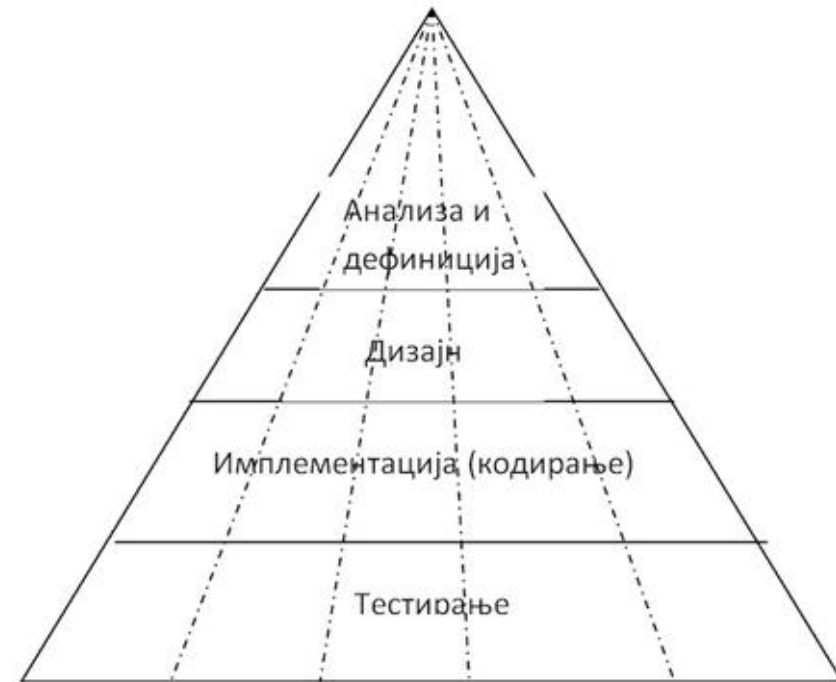
- Констатовање неиспуњености захтева у фази тестирања.
- Поновни пролазак кроз све претходне фазе и ново тестирање.
- Један пролаз кроз све четири фазе је тзв. итерација.
- Након довољно итерација иде се на фазу коришћења.
- Скупљи финансијски и временски.
- Више одговара реалности.



РЕШАВАЊЕ ПРОБЛЕМА (2)

СПИРАЛНИ МОДЕЛ

- Просечно утрошено време по фазама:
 - анализа и дефиниција ... 20%,
 - дизајн.....15%,
 - имплементација.....20%,
 - тестирање.....45%.
- Сложени програми захтевају опсежно тестирање.
 - И до неколико стотина хиљада линија кода.
 - Дешава се детекција грешака и након неколико година успешне употребе.



ЈЕЗИЧКИ ПРОЦЕСОРИ

- Програм за обраду језика.
- Примери употребе:
 - превођење са једног говорног на други,
 - са једног програмског на други програмски.
- Крајњи (циљни) језик рачунара је бинарни, тј. машински.
- Програмски језици:
 - виши програмски језици (даљи по апстракцији од машинских).
 - нижи (машински оријентисани, за конкретну машину).
- Превођење са машинског оријентисаног на машински релативно лако.
- Са вишег програмског језика на машински је теже.

АСЕМБЛЕРИ

- Језички процесори за превођење са машински оријентисаних на машински.
- Процес превођења се зове асемблирање.

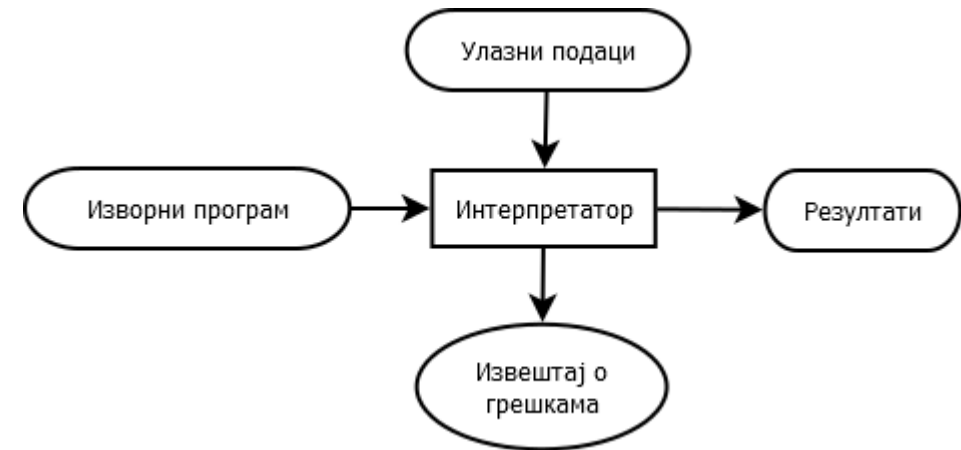
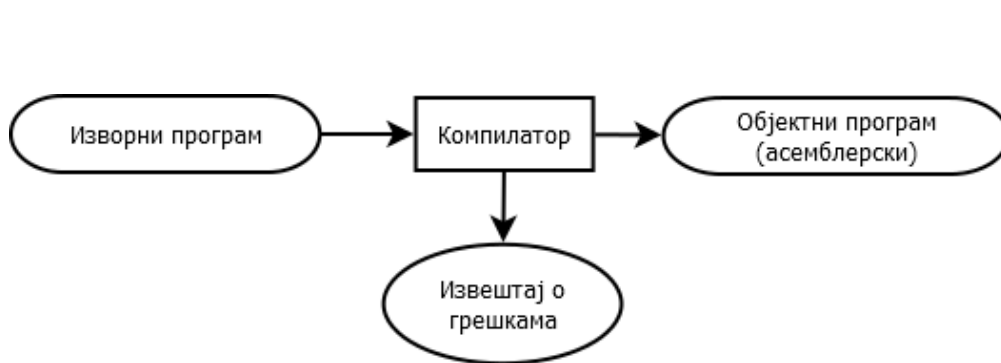


- Изворни програм се преводи асемблером у објектни.
 - Објектни је бинаран, али није још спреман за извршавање.
- Пунилац (повезивач) објектни смешта на одређено место у меморији.
 - Тако настаје извршни програм.

ПРЕВОЂЕЊЕ ВИШИХ ПРОГРАМСКИХ ЈЕЗИКА

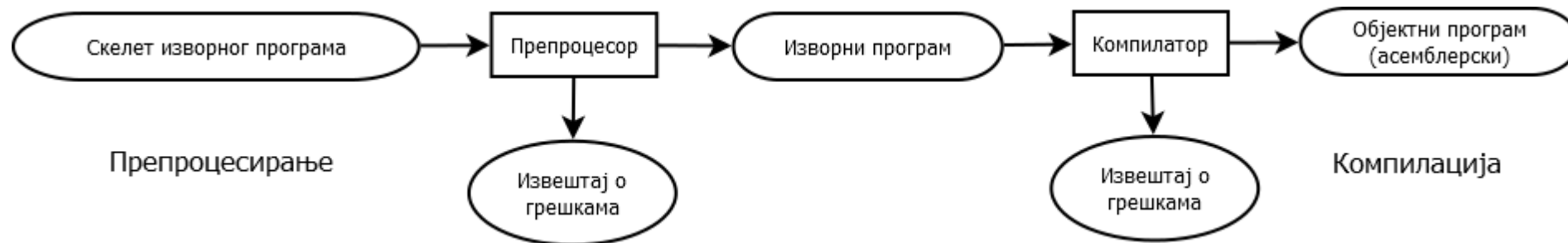
- Два стандардна приступа превођења виших програмских језика на машински:
 - компилација и
 - интерпретација.
- Компилација.
 - Процес превођења и извршавања временски раздвојени.
 - Компајлери.
- Интерпретација.
 - Процес превођења и извршавања временски повезани.
 - Преводе се део наредби па се изврши, па се преводе опет део па изврши итд.
 - Интерпретатори или интерпретери.

КОМПИЛАЦИЈА И ИНТЕРПРЕТАЦИЈА



КОМПИЛАЦИЈА СА ПРЕПРОЦЕСИРАЊЕМ

- У програму се могу наћи конструкције које не припадају програмском језику.
- Додатном обрадом се своде на конструкције тог језика.
- Тај процес се назива препроцесирање.



ОСТАЛИ ПРИСТУПИ

- Програмски језици могу имати и компилаторе и интерпретаторе.
 - На основу начина употребе се може користити једно или друго (нпр. Python).
- Али се најчешће дизајнирају имајући у виду искључиво једно или друго.
- Постоје и језици код којих се комбинују компилација и интерпретација.
 - Програмски језик Јава је најбољи пример.

ПИТАЊА И ЗАДАЦИ

1. Упоредити водопадни и спирални модел креирања софтвера.
По чему су слични, по чему се разликују?
2. Упоредити процес компилације програма и интерпретације програма.
Истражити који програмски језици врше компилацију, а који интерпретацију.
3. Шта се подразумева под процесом претпроцесирања?
4. Описати процес асемблирања.