Објектно орјентисано програмирање

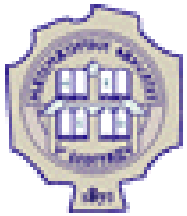
Владимир Филиповић
vladaf@matf.bg.ac.rs
Александар Картељ
kartelj@matf.bg.ac.rs

Предефинисани типови и објекти у програмском језику Јава

Владимир Филиповић
vladaf@matf.bg.ac.rs
Александар Картељ
kartelj@matf.bg.ac.rs

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs



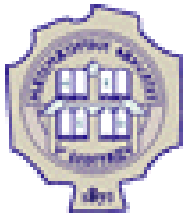
Улазна тачка програма

- Целокупан Јава програмски код се налази у класама.
- Улазна тачка Јава програма је метод main(), тј. извршавање програма почиње извршавањем ове методе (функције)
  - да би се програм могао извршити, метод main() мора бити означен кључном речи public
  - да би се програм могао извршити, метод main() мора бити означен кључном речи static
  - метод main() не сме да врати ништа, тј. мора вратити void
  - на крају, метод мора имати један параметар који омогућује да се прочитају вредности прослеђена преко командне линије приликом покретања програма
- Дакле, заглавље овог метода мора да има следећи облик:

```
public static void main(String[] argumentiKomandneLinije)
```

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

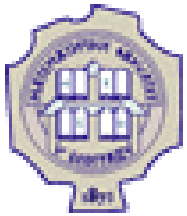


Функционална декомпозиција програма

- Статички метод main() обично није једини метод који се налази у класама.
- Обично се делови функционалности издвајају у посебне целине.
- Један начин да се то постигне је тзв. функционална декомпозиција.
  - у том случају се реализација издвојених функционалности измешта из метода main() у посебни издвојени метод (такође маркиран кључном речју static)
  - у методу main() остаје само позив тог издвојеног статичког метода
- Поступак декомпозиције се може даље наставити, ако има потребе за тим

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs



Функционална декомпозиција програма (2)

- Код функционалне декомпозиције, статички методи омогућавају да се дугачак код разбије у мање целине и на тај начин доприносе прегледности кода.
- Основна структура статичког метода:

povratniTip static imeMetoda(arg_1, arg_2,…,arg_n)

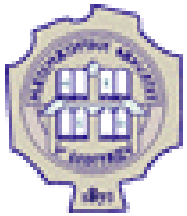
```
{
        Kod metoda
}
```

Telo metoda

- Чак и ако нема аргумената, заграде су обавезне и у декларацији и у позиву
- Ако метод треба да врати вредност, то се постиже тако што се у телу метод користи наредба return

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

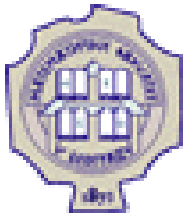


Функционална декомпозиција програма (3)

- Приликом позива се дешава следеће:
  1. евалуирају се аргументи из позива метода
  2. израчуната вредност аргумената замењује параметре метода, према редоследу навођења – вредност првог аргумента замењује вредност првог параметра, други аргумент замењује други параметар итд.
     - Да би то функционисало, а с обзиром да је Јава строго типизиран језик, потребно је:
       - да се број параметара у дефиницији метода поклапа са бројем аргумената методе и
       - да тип сваког од аргумената буде у сагласности са типом одговарајућег параметра

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

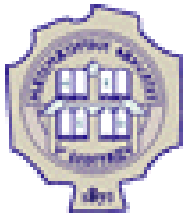


Функционална декомпозиција програма (4)

- Приликом позива се дешава следеће:
  4. извршавају се редом наредбе у телу метода
  5. извршавање тела метода се завршава када се дође до извршавања наредбе return или до извршавања последње наредбе у телу метода – тада наступа повратак у позивајући метод
     - ако је извршавање метода из претходне тачке завршено наредбом return, онда је резултат рада функције вредност израза који следи иза те наредбе и тип тог израза мора да одговара повратном типу из декларације метода.
  6. приликом повратка у позивјући метод, извршавање се наставља од наредбе која следи иза наредбе позива

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

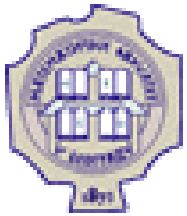


Рекурзија

- Рекурзија (лат. recursio - враћање) у означава поступак или функцију (метод) који у својој дефиницији користе сами себе
- Другим речима, уколико неки поступак захтева да делови проблема које је раздвојио од других бивају независно подвргнути истом том поступку, тај поступак је рекурзиван
- У сваком рекурзивном поступку се рауликују две целине:
  - свођење проблема датог типа и димензије на проблем/проблема истог типа али ниже димензије (тзв. рекурзивни корак)
  - завршетак поступка када је димензија проблема мала (тзв. излаз из рекурзије)
- Разликују се саморекурзија (метод се директно своди на себе) и узајамна рекузија (свођење иде преко других метода)

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

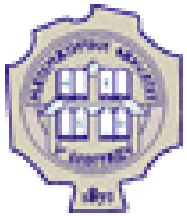


Класа System

- Класа System садржи више корисних објеката и метода
- Ова класа, тј. њена функционалност је одмах на располагању програмеру
- Нема могућности да се креира примерак ове класе
- Пун назив ове класе је java.lang.System
  - она се налази у пакету lang унутар пакета java и на њу се може референсирати коришћењем пуног имена тј. помоћу java.lang.System
  - међутим, за њу, као и за остале класе из пакета java.lang (и само за њих) важи да не мора да се користи пуно име, већ може да се референсише на класу коришћењем скраћеног назив класе тј. помоћу System.

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

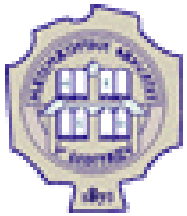


Класа System (2)

- Класа System, поред осталих фунционалности, обезбеђује и следеће могућности:
  - стандардни излаз, објекат System.out
  - ток за приказ порука о грешкама, System.err
  - мерење протеклог времена помоћу часовника рачунара
  - интеракција са сакупљачем отпадака
  - приступ спољашње дефинисаним особинама и променљивима окружења
  - директан завршетак рада програма
  - стандардни улаз System.in
  - помоћни метод за брзо копирање меморијских блокова, тј. делова низа
  - механизам за учитавање датотека и библиотека

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

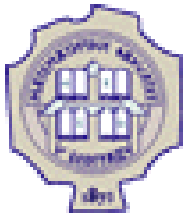


Класа Object

- Класа java.lang.Object је корен дрвета који представља хијерархију класа.
  - дакле, за сваку од Јава класа, класа Object је њихова надкласа – ако не директна надкласа, онда наткласа наткласе итд. – ако се прође кроз надкласе, на врху се налзи класа Object
  - Сви објекти, укључујући и низове и енумерисане типове, имплементирају методе који су дефинисани у класи Object
  - Методи дефинисани у класи Object се могу применити на било који објекат (тј. примерак било које Јава класе)

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs



Класа String

- Класа java.lang.String представља ниске, које обмотавају (енкапсулирају) низове знакова
- Сви ниска литерали у програмском језику јава, као што је нпр. "abc", су имплементирани као примерци ове класе
- Ниске тј. примерци класе String су имутабилни (константни) и када се ови објекти једном креирају њихове вредности не могу да се мењају
- Ниске су објекти, па над нискама могу да се примене све методе које су дефинисане у класи Object, при чему се добиају другачији одговори
- Над примерцима класе String могу да се користе методе којима се реализују све уобичајене операције над нискама (налепљивање, претрага, замена, форматирање и сл.)

Математички факултет

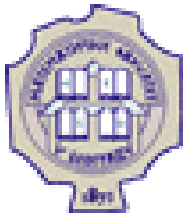
{vladaf, kartelj}@matf.bg.ac.rs

Класа StringBuilder

- Како су примерци класе String су имутабилни (константни), то узастопно налепљивање ниски коришћењем метода класе String није ефикасно
- За ефикасно налепивање ниски се, стога користи класа StringBuilder чији су објекти мутабилни
  - метод append() ове класе, због мутабилности објекта који садржи ниску, омогућава брже надовезивање
  - на располагењу су и методи аналогни оним који постоје у класи String, као и метод insert() за уметање на дату позицију у ниски
  - метод setCharAt() омогућава промену знака ниске на датој позицији „на лицу места“, без непотребног копирања.

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

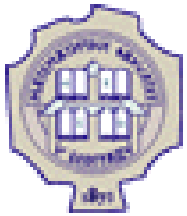


Класа Integer

- Класа java.lang.Integer омотава вредност примитивног типа int у објекат
- Објекат примерак класе Integer у себи садржи поље типа int у ком се чува вредност тог Integer објекта
- Примерци класе Integer су имутабилни - када се овакав објекат једном креира њихова вредност не може да се мења
- Над примерцима класе Integer могу да се примене методи дефинисани у класи Object, са другачијим резултатом
- Допуштена је додела int константе Integer објекту и обратно – сам преводилац зна, зависно од контекста, да приликом превођења у бајт-код уметне наредбе за конверзију (аутоомотавање – autboxing)
- Операције над int су брже од операција над Integer

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

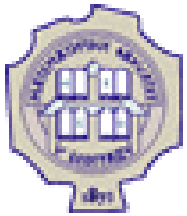


Класа Long

- Класа java.lang.Long омотава вредност примитивног типа long у објекат
- Објекат примерак класе Long у себи садржи поље типа long и у том пољу се чува вредност тог Long објекта
- Примерци класе Long су имутабилни - када се овакав објекат једном креира њихова вредност не може да се мења
- Над примерцима класе Long могу да се примене методи дефинисани у класи Object, са другачијим резултатом
- Допуштена је додела long константе објекту примерку класе Long и обратно - сам преводилац зна, зависно од контекста, да приликом превођења у бајт-код уметне наредбе за конверзију
- Операције над примитивним типом long су брже од операција над имутабилним примерцима класе Long

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

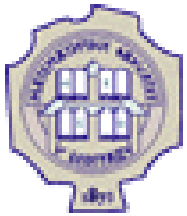


Класа Character

- Класа java.lang.Character омотава вредност примитивног типа char
- Примерак класе Character у себи садржи поље типа char у ком се чува вредност објекта
- Примерци класе Character су имутабилни
- У класи Character су дефинисани методи којима се реализују уобичајене операције над знацима
- Над примерцима класе Character могу да се примене методи дефинисани у класи Object, са другачијим резултатом
- Допуштена је додела char константе објекту примерку класе Character и обратно
- Операције над примитивним типом char су брже од операција над примерцима класе Character

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

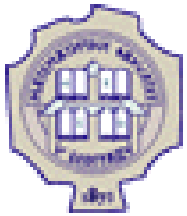


Класе Float и Double

- Класа java.lang.Float и java.lang.Double омотавају вредности примитивних типова float и double
- Примерци ових класа у себи садржи поље одговоарајућег примитивног типа у ком се чува вредност објекта
- Примерци ових класа су имутабилни
- У овим класама су дефинисани методи којима се реализују уобичајене операције над бројевима у покретном зарезу, као и контанте које представљају бројеве у покретном зарезу, по формату дефинисаним стандардом IEEE 754
- Над примерцима ових класа могу да се примене методи дефинисани у класи Object
- Аутоматизована је конверзија између вредности примитивног типа и објекта примерка класе-омотача, и то у оба смера

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

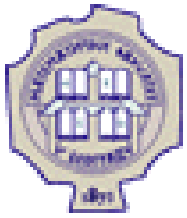


Класа Math

- Класа java.lang.Math садржи:
  - математичке константе e и π, исказане као бројеве у покретном зарезу двоструке тачности (64 бита по формату IEEE 754)
  - методе за реализацију основних математичких операција, као што су експоненцијалне и логаритамске функције, степеновање, тригонометрија и сл.
  - методе за генерисање псеудо-случајних бројева из интервала **[0,1)** по униформној расподели

Математички факултет

{vladaf, kartelj}@matf.bg.ac.rs

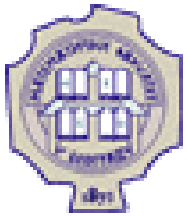


Класе за манипулацију са датумима и временима

- То су класе java.time.LocalDate, java.time.LocalTime и java.time.LocalDateTime које служе за рад са само са датумима, само са временима у дану и са паровима датум-време, респективано
  - класе су уведене релативно скоро, раније се користила класа java.util.Date, са којом је било проблема
  - како се класе не налазе у пакету java.lang, то се приликом реферисања на њих мора користити пуно име тих класа
- Садрже методе за реализацију основних операција над датум-временима као што су поређења датум-времена, одређивање датума-времена на основу постојећег, мерење протеклог времена између два датума-времена и сл.
- За форматирање се користи java.time.format.DateTimeFormatter

Математички факултет

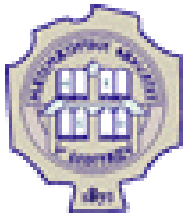
{vladaf, kartelj}@matf.bg.ac.rs



Класа Scanner

- Класа java.util.Scanner представља једноставни скенер текста, који може парсирати примитивне типове и ниске коришћењем регуларних израза
- Како се класа java.util.Scanner не налази у пакету java.lang, то се приликом реферисања на њу мора користити њено пуно име
- Следе важне особине класе java.util.Scanner:
  - За разлику од математичких функција, пре скенирања улаза је неопходно креирати примерак класе Scanner
  - Направљени објекат при скенирању раздваја улаз на токене коришћењем обрасца за раздвајање (енг. delimiter pattern), подразумевано постављеног на белине
  - Операција скенирања може да блокира чекајући на улаз
  - Улаз за скенер се одређује приликом креирања објекта

Математички факултет



{vladaf, kartelj}@matf.bg.ac.rs

Захвалница

Велики део материјала који је укључен у ову презентацију је преузет из презентације коју је раније (у време када је он држао курс Објектно орјентисано програмирање) направио проф. др Душан Тошић.

Хвала проф. Тошићу што се сагласио са укључивањем тог материјала у садашњу презентацији, као и на помоћи коју ми је пружио током конципцирања и реализације курса.