Објектно орјентисано програмирање

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

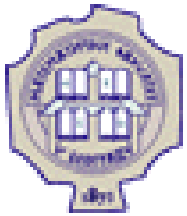
Математички факултет

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

Дизајн програмског језика Јава

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

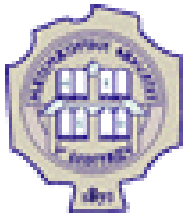
Математички факултет
{vladaf,kartelj}@matf.bg.ac.rs



Увод

- Пре него што се креира апликација, аплет или библиотека у Јави, важно је да се разуме како Јава ради.
- У презентацији која следи упознајемо се са:
  - језиком Јава,
  - ограничењима језика Јава
  - и Јава окружењем за извршавање.
- Проучавамо и како се може постићи да Јава програмски код буде вишеструко коришћен.

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

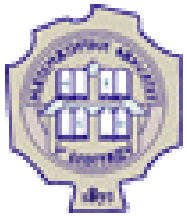


Особине језика Јава

- Јава је и компајлирана и интерпретирана
  - проучава се начин и превођења и интерпретирања Јаве,
    те како тако дефинисани процеси превођења и интерпретације
    утичу на брзину Јаве и на њену независност од платформи.
- Јава се извршава коришћењем Јава виртуалне машине
  - проучава се дизајн Јава виртуалне машине (JVM),
    како она ради током извршавања Јава програма
    и како то утиче на наше одлуке.
    Такође ће бити речи и о сигурности коју обезбеђује JVM.
- Јава користи Java API
  - Јава садржи скуп класа које су на располагању програмеру за коришћење -
    њихов назив је Java API.
    Проучава се начин коришћења Java API-ја у програмирању.

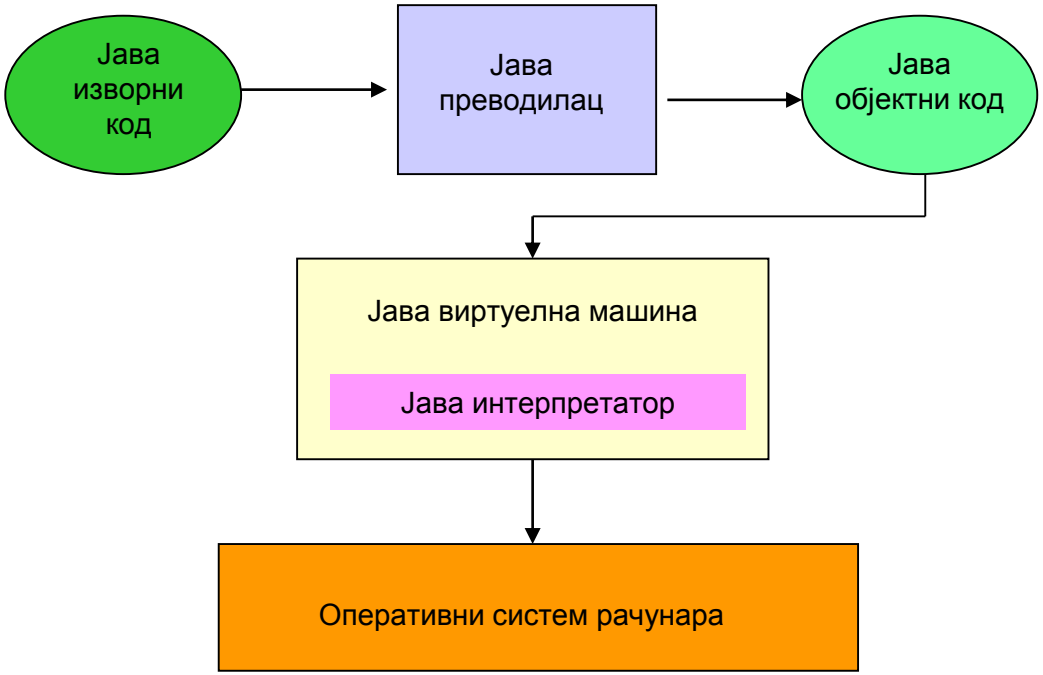
Математички факултет

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



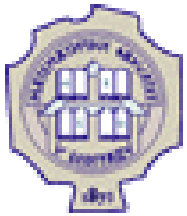
Извршење Јава програма

Јава је језик који се преводи и интерпретира.



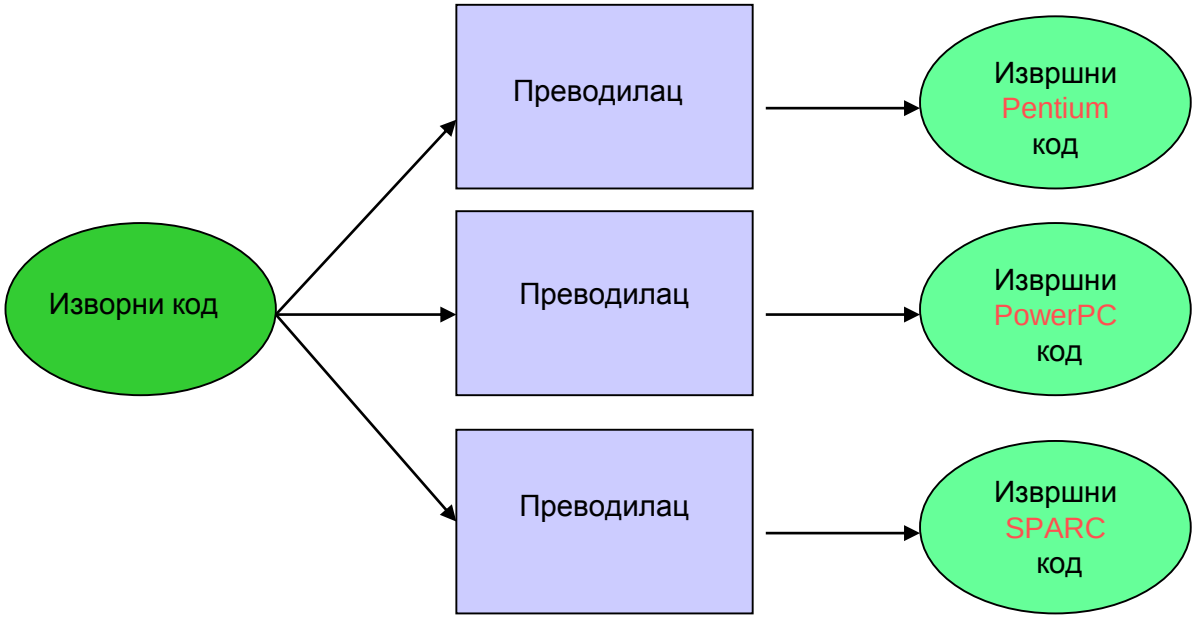
Математички факултет

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



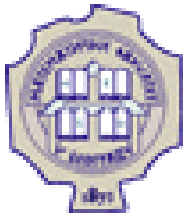
Извршење Јава програма (2)

Традиционални начин креирања извршног кода превођењем изворног програма



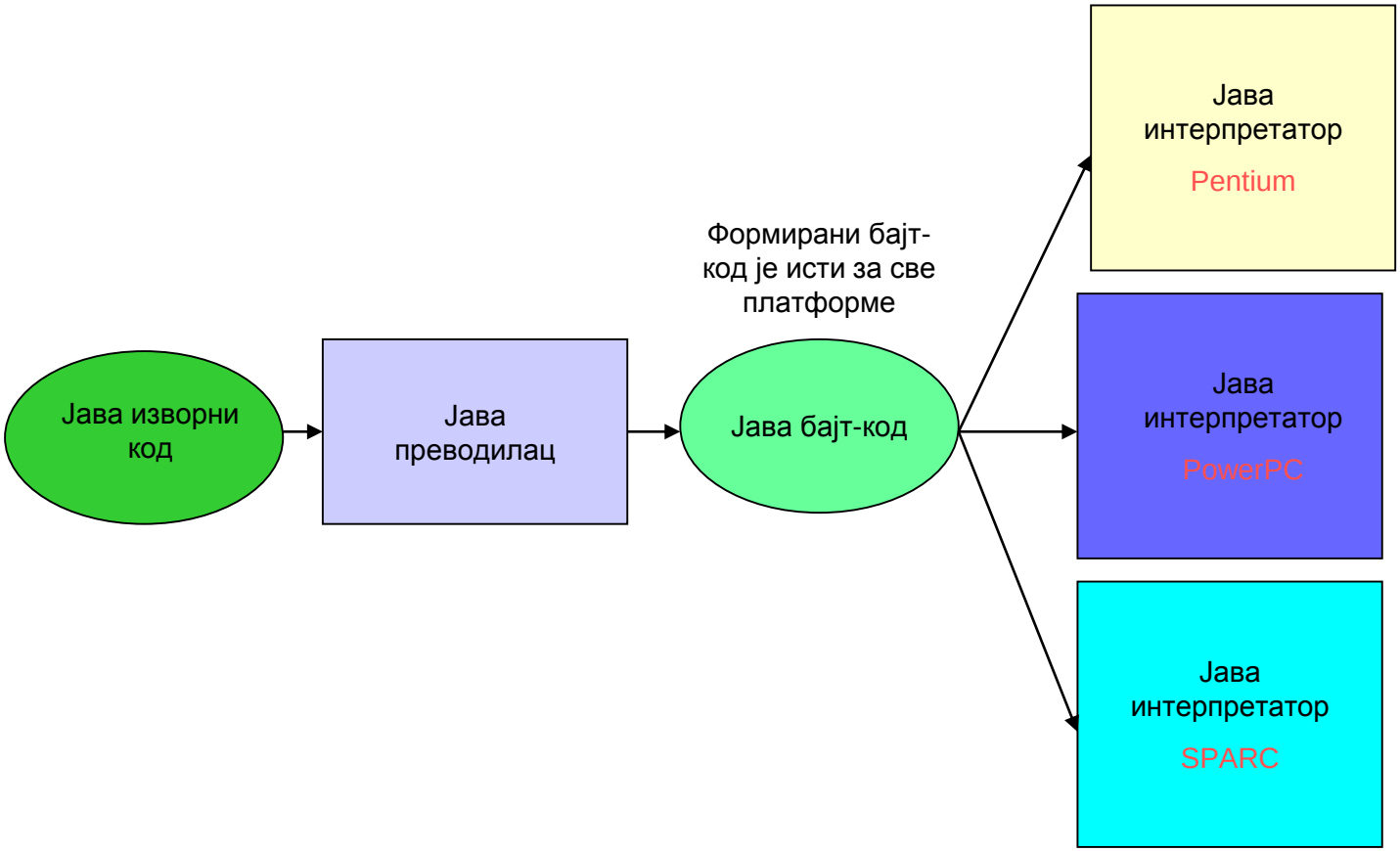
Математички факултет

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



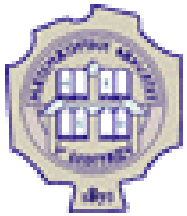
Извршење Јава програма (3)

Креирање Јава извршног кода од изворног програма – превођење и интерпретација



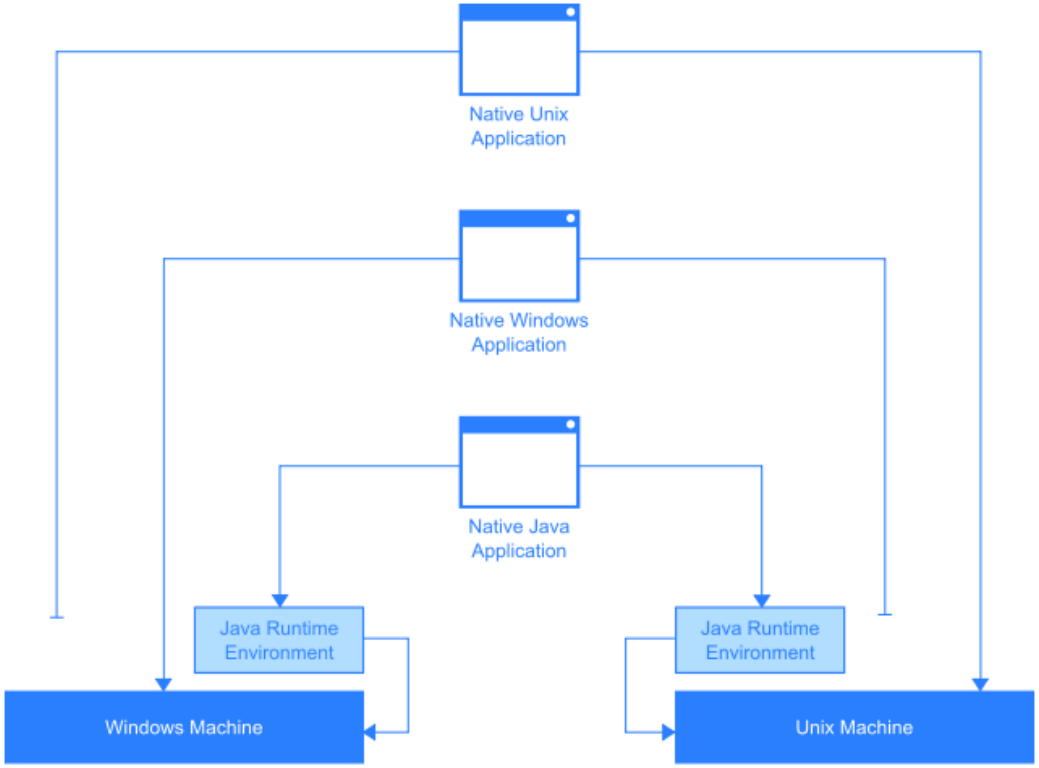
Математички факултет

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



Извршење Јава програма (4)

Дијаграм показује разлику између начина извршења код традиционалних и код Јава апликација.

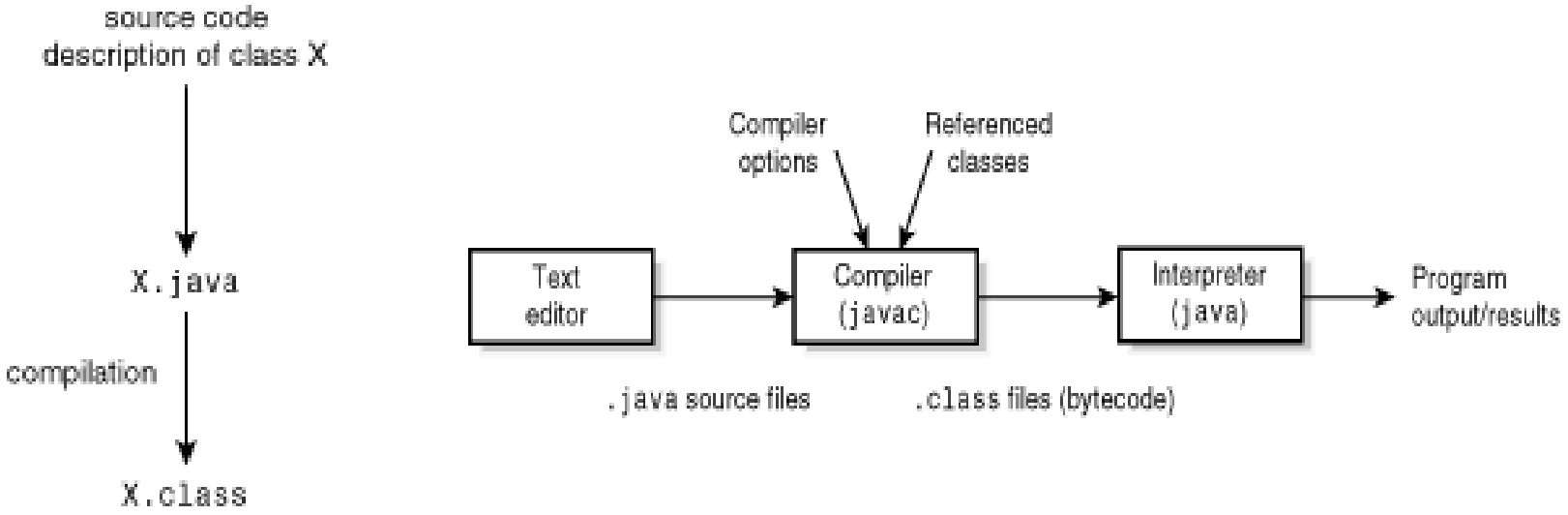


Математички факултет

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

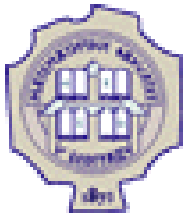
Извршење Јава програма (5)

- Дакле, написани изворни Јава програм се прво преведе коришћењем Јава компајлера **javac** у тзв. бајт-код.
- Потом се преведени бајт-код извршава уз помоћ Јава интерпретатора **java.**

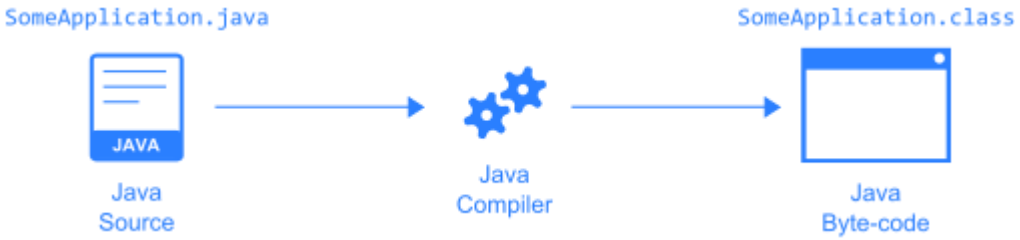


Математички факултет

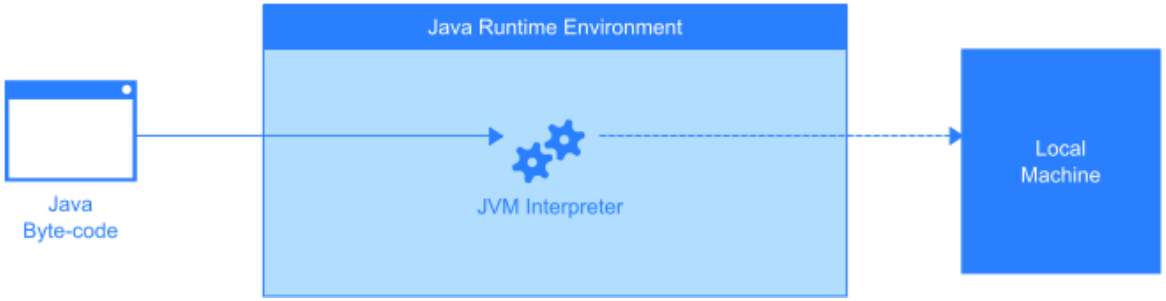
{vladaf,kartelj}@matf.bg.ac.rs



Извршење Јава програма (6)



Превођење Јава кода



Интерпретирање бајт-кода

Математички факултет

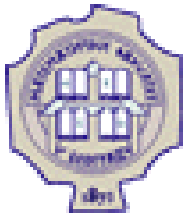
{vladaf,kartelj}@matf.bg.ac.rs

Извршење Јава програма (7)

- Креирани бајт-код је бинаран и архитектонски неутралан (платформски неутралан).
- Јава окружење за извршавање постоји посебно за сваку конкретну платформу и оно је надлежно за превођење бајт-кода до извршног кода.
- Јава изворни код и Јава бајт-код остаје исти без обзира на којој се платформи извршава:
  - Тако се, на пример, Јава аплет може написати и компајлирати на UNIX систему и потом убацити тај аплет у веб страну.
- Коришћењем Јаве се постиже да постоји јединствени изворни Јава код, а да програм ради на различитим платформама – “Write once, run everywhere”.

Математички факултет

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



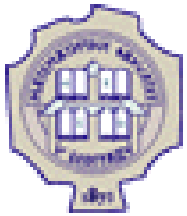
Извршење Јава програма (8)

Зашто је комбинација компилације и интерпретације позитивна особина?

- Обезбеђује сигурност и стабилност:
  - Јава окружење садржи елеменат назван повезивач (eng. linker), који проверава податке који долазе на Јава виртуелну машину како би се уверила да они не садрже делове који би могли оштетити датотеке (сигурност) или на други начин онеспособити рад система (робусност).
- Ова комбинација компилације и интерпретирања разрешава проблеме неуклапања верзија.

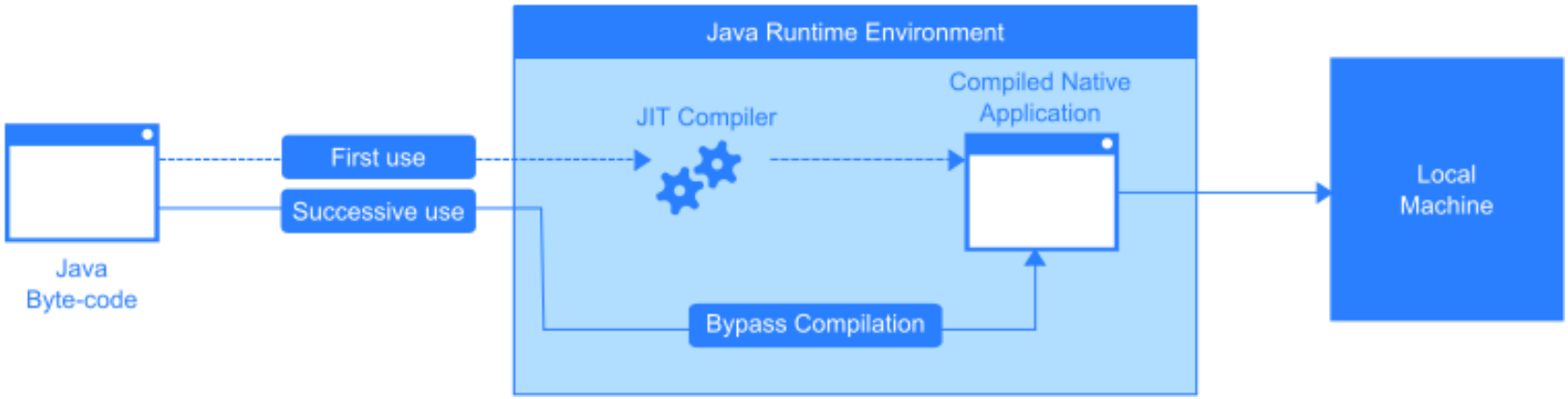
Математички факултет

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



JIT Јава компајлер

- Веб стране са аплетима се обично учитавају и приказују дуже од класичних (статичних).
- То је зато што се:
  - заједно са страном се на клијентски систем довлачи бајт-код;
  - потом серија процедура проверава сигурност аплета, тј. његову робусност.
- Јавина преносивост, дакле, изазива губитак перформанси.
- Губитак перформанси је смањен коришћењем Just-in-time (срп. «у право време») или JIT компајлера.
- JIT компајлер преводи Јава методе у машински код за конкретну платформу на којој се користи.



Математички факултет

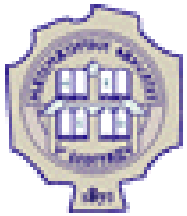
{vladaf,kartelj}@matf.bg.ac.rs

Јава виртуелна машина

- Језгро Јаве је JVM (eng. Java Virtual Machine).
- JVM је виртуални рачунар који постоји само у меморији.
- JVM допушта да Јава програми буду извршавани на разноврсним платформама (портабилност).
- Да би Јава програми могли да раде на одређеној платформи, JVM мора да буде имплементирана на тој платформи.
- JVM је врло мала када се имплементира у RAM-у:
  - Таква мала величина JVM омогућава да се Јава користи у разноврсним електронским уређајима.
  - Цео језик Јава је оригинално развијан тако да се на уму има и кућна електроника.

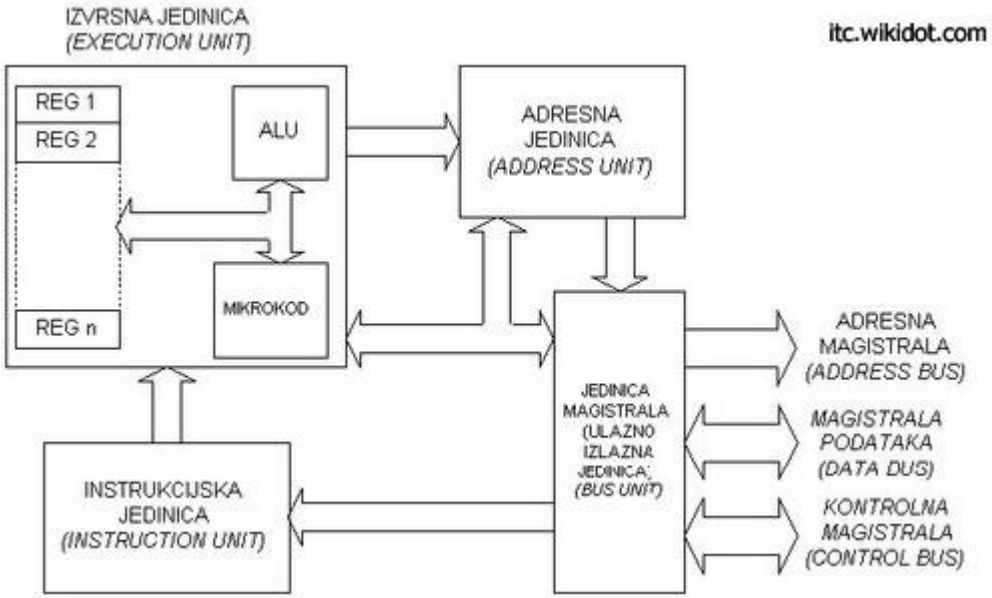
Математички факултет

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

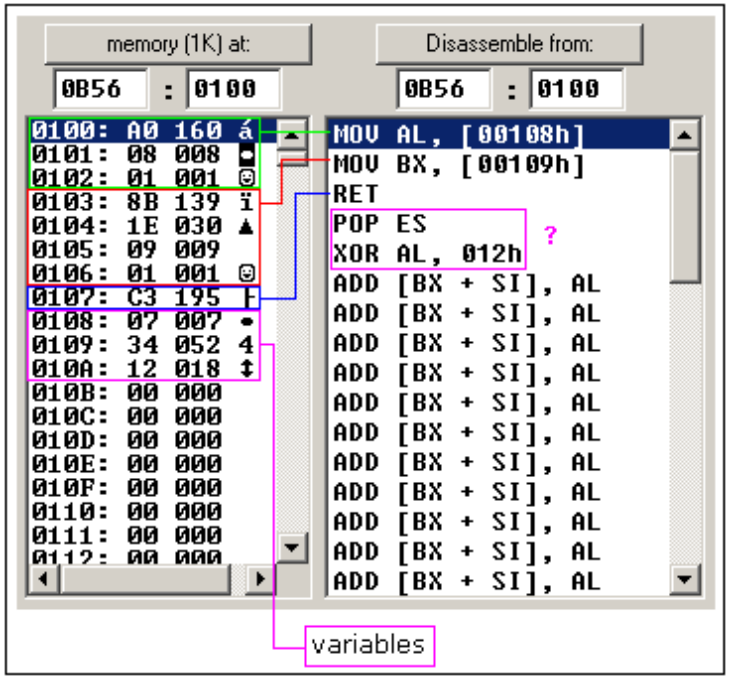


Јава виртуелна машина (2)

Архитектура JVM одсликава архитектуру конкретног рачунарског система.



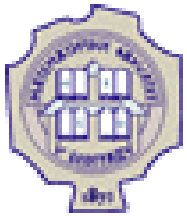
Архитектура рачунарског система



Пример асемблерског кода

Математички факултет

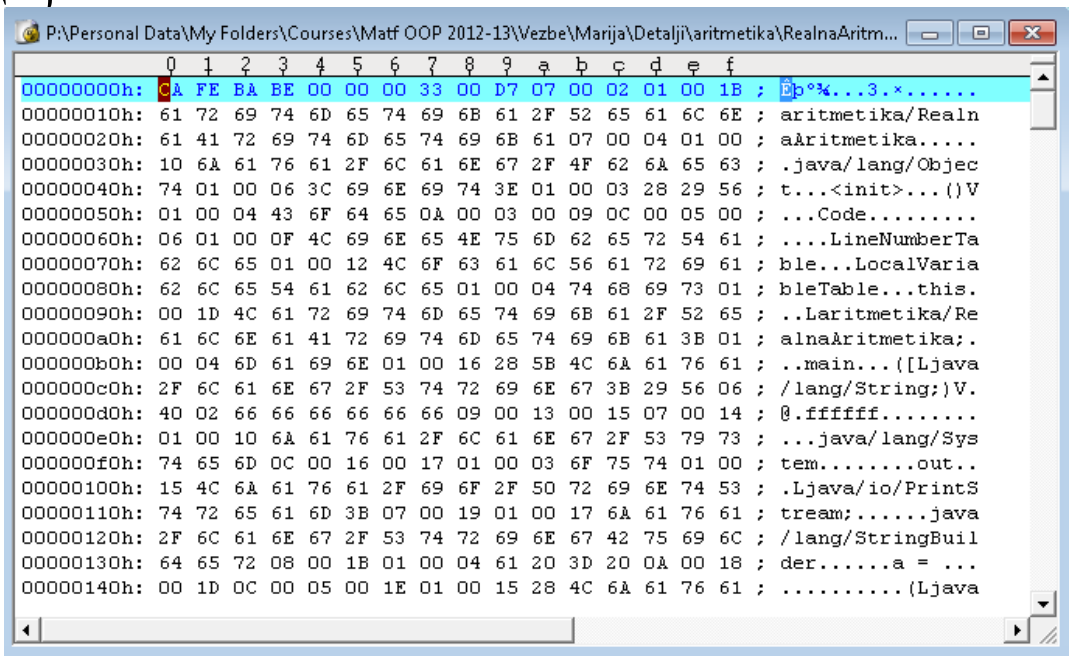
{vladaf,kartelj}@matf.bg.ac.rs



Јава виртуелна машина (3)

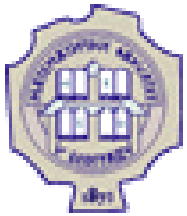
- JVM извршава бајт-код. Пре тога, програм **javac**, тј. Јава преводилац, процесира .java датотеке и резултујући бајт-код чува у датотеци са екстензијом .class.
- JVM чита ток бајт-кодова из .class датотеке као секвенцу машинских инструкција.
- Извршавање инструкција бајт-кода опонаша извршавање машинских инструкција.

Пример бајт-кода



Математички факултет

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



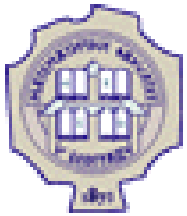
Јава виртуелна машина (3)

- Свака од инструкција JVM је слична асемблерској инструкцији:
  - Састоји се од једнобајтног операционог кода (опкода), који представља специфичну и препознатљиву команду;
  - И од нула, једног или више операнада (података потребних за комплетирање инструкције).

Пример бајт-кода

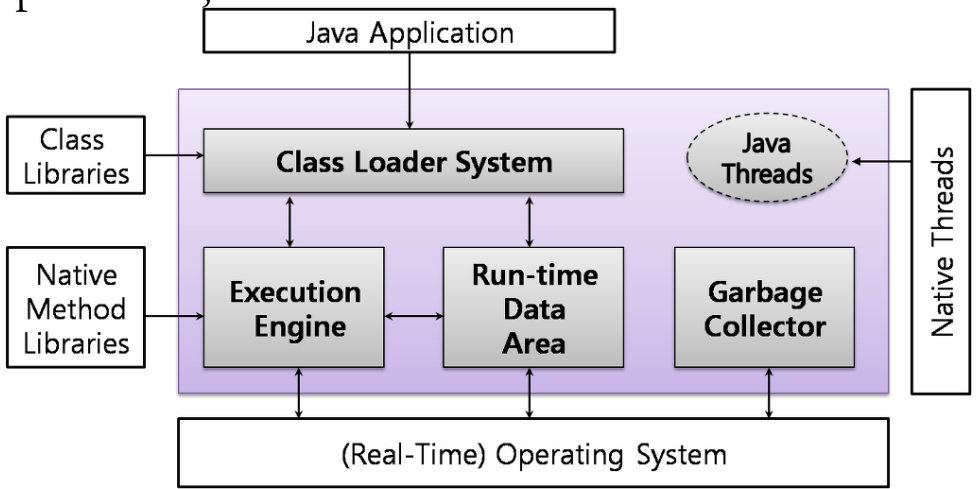
Математички факултет

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



Јава виртуелна машина (4)

- Приликом свог рада, тј. приликом извршавања бајт-кода Јава апликације, JVM има интеракцију са:
  - библиотекама Јава класа,
  - библиотекама нативних (eng. native) метода,
  - са нативним нитима
  - и са реалним оперативним системом рачунара на ком се извршава JVM.
- JVM садржи:
  - систем за учитавање класа,
  - подсистем за извршавање,
  - област за податке приликом извршавања,
  - сакупљач отпадака и Јава нити.



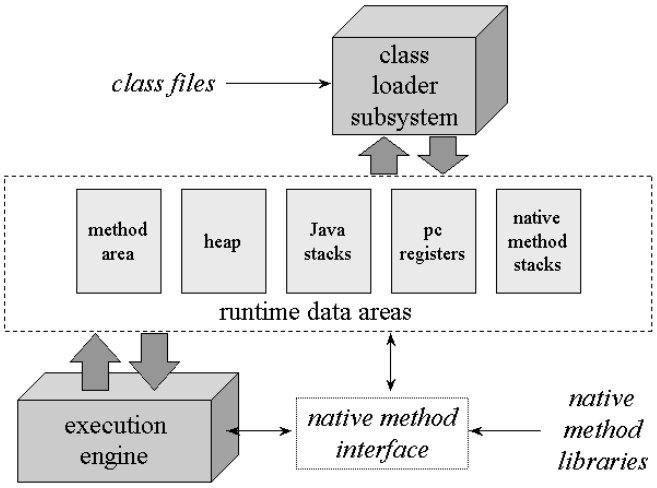
Структура Јава виртуалне машине

Математички факултет

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

Меморија Јава виртуелне машине

- Област за податке приликом извршавања се састоји од:
  - области за Јава методе,
  - простора - хип меморије (eng. heap),
  - стек меморије,
  - регистара
  - и стека за нативне методе.
- Меморија код JVM:
  - Адресна дужина кода JVM је 32 бита, па JVM може адресирати до 4 Gb.
  - Стек, простор који се рециклира и област за методе су смештени у оквиру тих 4 Gb адресибилне меморије.
  - Јава метод не може бити дужи од 32 Kb.



Математички факултет

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

Позиви метода

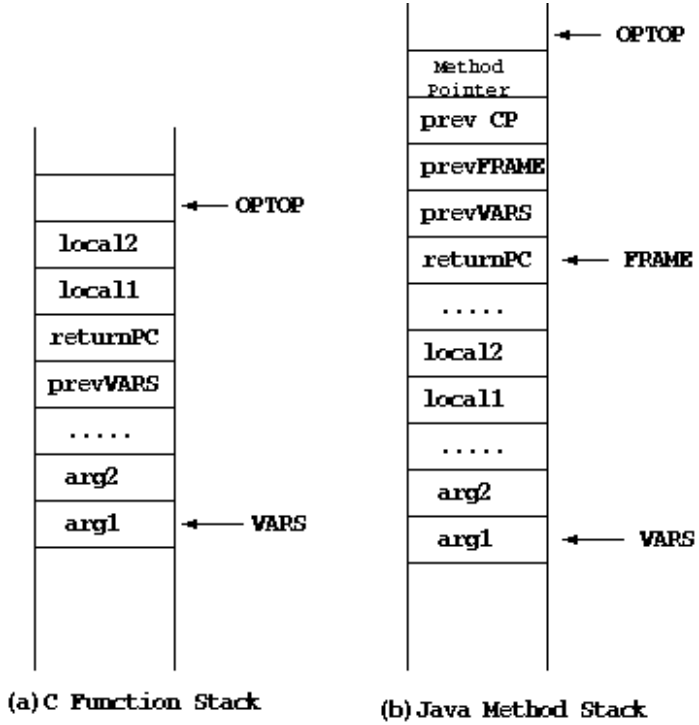
- Сваки регистар код JVM је дужине 32 бита, па може чувати једну 32-битну адресу.
- JVM користи следеће регистре за управљање системским стеком (*):
  - бројач (counter)
  - врх (optop)
  - оквир (frame)
  - променљиве (vars)
- Тим који је развијао Јаву одлучио је да користи само четири регистра због перформантности (**).

Математички факултет

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

Позиви метода (2)

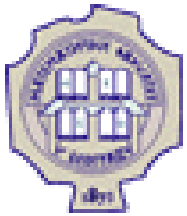
- Стек садржи податке релевантне за извршавање метода.
- Стек ради по принципу «last-in, first-out», или LIFO што је природно за извршавање метода.
- На врх стека показује регистар оптоп.
- Регистар оквир показује где се метод извршава у меморији.
- Регистар променљиве указује на прву променљиву на стеку.



Пример стека: C и Јава

Математички факултет

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

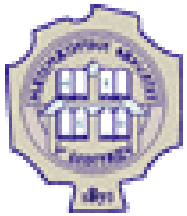


Простор и скупљач отпадака

- Простор (енг. heap) је део меморије из кога се врши инстанцирање и алокација објекта – примерка дате класе.
- Кад год се алоцира меморија са оператором new, та меморија долази из простора.
- Окружење у ком се Јава програм извршава чува информације о референцама на сваки објекат из простора и аутоматски ослобађа објекте које више нико не реферише.
- Ова операција ослобађања простора зове се скупљање отпадака (енг. garbage collection).
- Скупљач отпадака ради као позадинска нит и врши рашчишћавање током неактивности процесора.

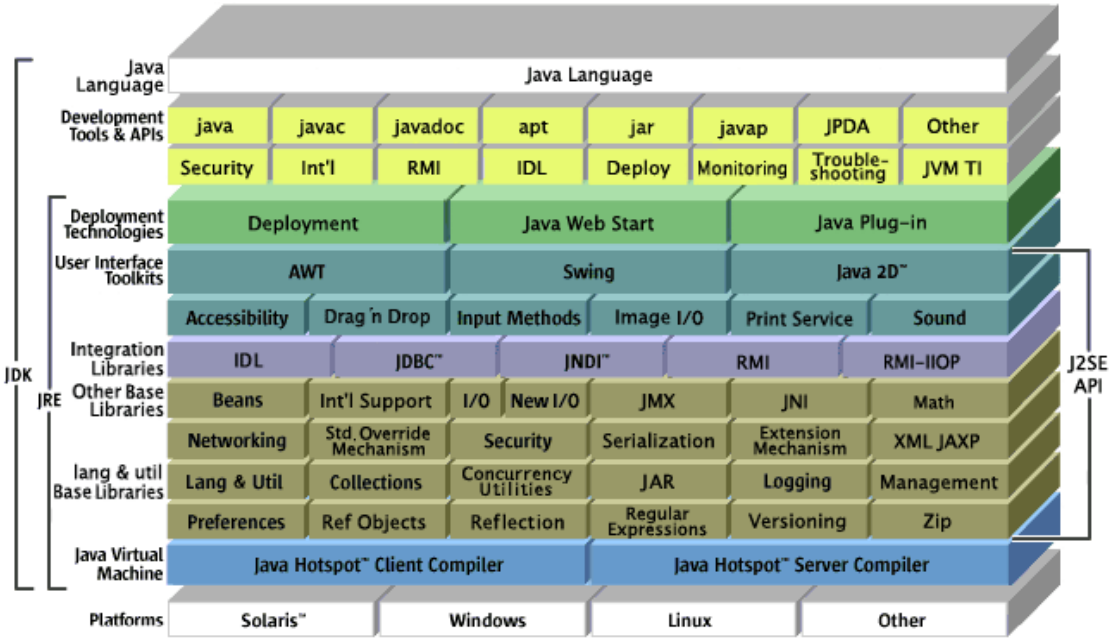
Математички факултет

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



Алати за Јава развој (JDK)

- Разлог велике популарности језика Јава је, поред квалитета њеног дизајна, постојање богатог скуп алата, као и скуп библиотека који је организован у пакете.
- Ови већ оформљени објекти омогућавају брз старт при раду са Јавом, из два разлога:
  - програмер не мора поново да развија њихову функционалност
  - изворни код је свима доступан



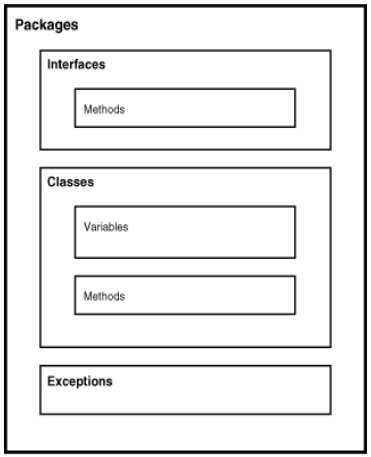
Елементи Јаве и JDK

Математички факултет

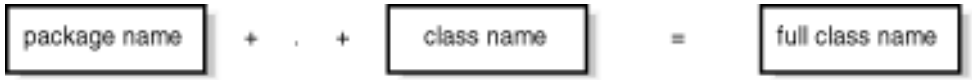
{vladaf,kartelj}@matf.bg.ac.rs

Java API

- Java Application Programming Interface, или Java API, је скуп класа које је развио Sun, за коришћење у језику Јава.
- Java API је дизајниран да би се помогло програмеру у развоју сопствених класа, аплета и апликација.
- Класе унутар Java API -ја су груписане у пакете, при чему сваки пакет може садржавати више класа и интерфејса.
- Надаље, свака од класа може имати више особина (енг. properties), више поља (енг. fields) и/или метода.



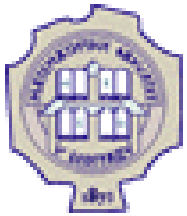
Пример Јава пакета



Пун назив Јава класе представља називе пакета (раздвојених тачком) иза кога следи тачка, па име Јаве класе

Математички факултет

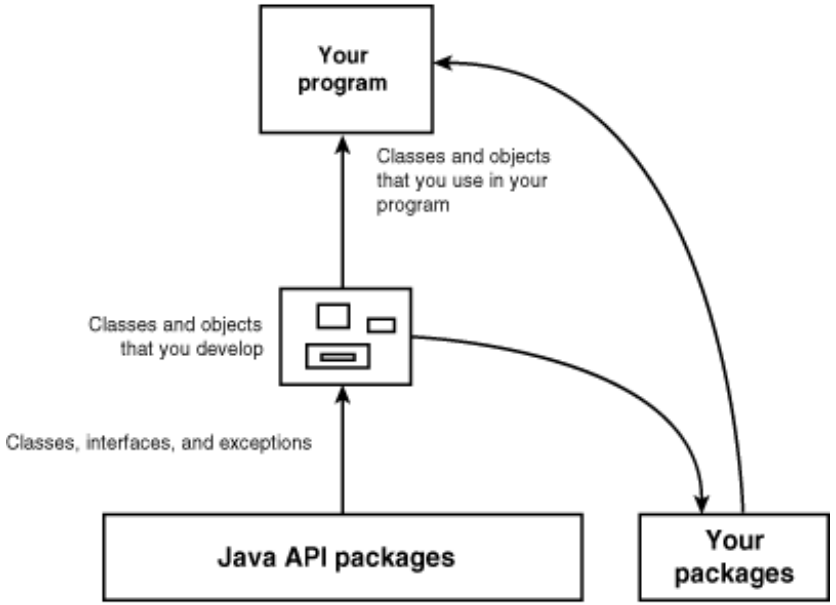
{vladaf,kartelj}@matf.bg.ac.rs



Java API (2)

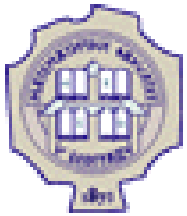
- Иако је могуће програмирати у Јави без великог знања API-ја, треба нагласити да свака новоразвијена класа зависи од бар једне класе из API-ја.
- Надаље, при развоју све сложенијих програма, који раде са нискама, сокетима и графичким интерфејсом, постаје веома значајно познавање објеката које је Sun обезбедио.

Пример програмирања коришћењем Java API



Математички факултет

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



Java API (3)

Кратак преглед најзначајнијих пакета који долазе са Јавом:

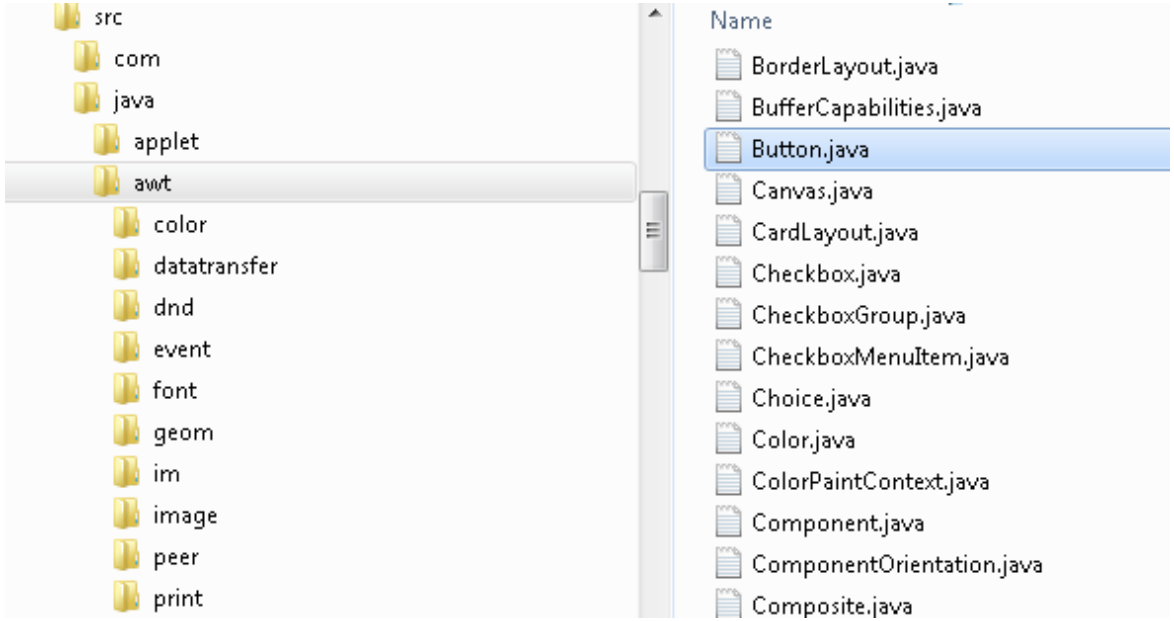
- Библиотеке за GUI као што су прозори, дијалози, дугмићи, текстуална поља, итд.:
  - Abstract Window Toolkit (AWT),
  - Swing и
  - Java FX класе
- Класе за мреже, URL-ове, клијент-серверске сокете.
- Класе за различите типове улаза и излаза.
- Класе за различите типове података, процесе и нити који се извршавају, стрингове, итд.
- Помоћне класе за датуме, колекције итд.
- Класе за развој аплета.
- Класе за манипулацију сликама.

Математички факултет

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

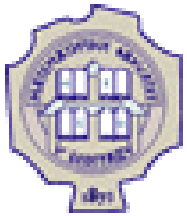
Java API (4)

- Пакети представљају начин за организовање класа.
- Пакет може садржавати класе и друге пакете, на сличан начин као што директоријум садржи датотеке и друге директоријуме.
- На пример, класе које припадају пакету AWT, јава.awt, су смештене у поддиректоријуму AWT директоријума JAVA.



Математички факултет

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



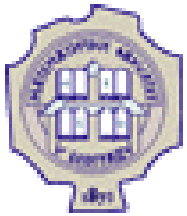
Java Core API

Централни (енг. core) API садржи пакете са објектима за које се гарантује да су доступни без обзира на Јава имплементацију:

- **java.lang**
  - Састоји се од класа које су централне за језик Јава.
  - Он обезбеђује не само класе-омотаче за просте типове података, као што су Character и Integer, већ и обраду грешака уз коришћење класа Throwable и Error
  - Надаље, класе System и SecurityManager омогућују програмеру контролу (до извесног нивоа) над Јава системом за извршавање.
- **java.io**
  - Стандардна улазно/излазна Јава библиотека.
  - Овај пакет обезбеђује програмеру могућност креирања и рада са токовима (eng. streams) података.
  - Подржан је рад са једноставним типовима као што је String и са сложеним типовима као што је StreamTokenizer.

Математички факултет

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



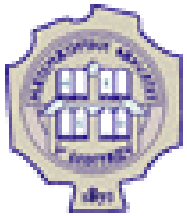
Java Core API (2)

- **java.util**
  - Садржи већи број корисних класа које нису могле бити уклопљене у друге пакете.
  - Класе које омогућавају рад са датумима,
  - Класе које омогућавају структурисање података, као што су Stack и Vector.
  - Класе које омогућавају парсирање улазног тока података

- **java.net**
  - Пакет java.net чини језик Јава мрежно заснованим језиком.
  - Он обезбеђује способност комуникације са удаљеним чворовима, било преко сокета, било коришћењем URL-ова.
  - На пример, коришћењем овог пакета програмер може креирати своје сопствене Telnet, Chat или FTP клијенте и/или сервере.

Математички факултет

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

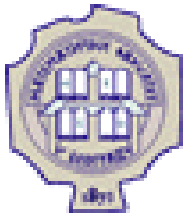


Java Core API (3)

- **java.awt**
  - Назив пакета java.awt је означава скраћеницу за Abstract Window Toolkit (AWT).
  - AWT садржи не само класе, као што је GridBagLayout, већ и неколико конкретних интерактивних алата, као што су Button и TextField.
  - Класа Graphics обезбеђује богатство графичких могућности, укључујући и могућност цртања разних облика и могућност приказа слика.
- **java.awt.image**
  - Пакет java.awt.image је блиско повезан са пакетом java.awt.
  - Овај пакет се састоји од алата који су дизајнирани за руковање и манипулацију сликама које долазе кроз мрежу.
- **java.awt.peer**
  - Пакет java.awt.peer је пакет са интерфејсима који служе као посредници између Јава кода и рачунара на коме се тај код извршава.

Математички факултет

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



Java Core API (4)

- **javax.swing**
  - Swing је уведен како би се превазишли проблеми на које су наилазили програмери који су користили AWT за креирање GUI апликација.
  - Наиме, AWT је сувише рано поверавао исцртавање прозора и контрола, тако да је иста арликација изгледала битно другачије на различитим платформама.
  - Swing се ослања на AWT и садржи класе као што су JButton и JTextField.

- **java.applet**
  - Пакет java.applet је најмањи пакет у API-ју,
    али је често коришћен због класе Applet која је дефинисана у њему.
  - Класа Applet садржи много корисних метода
  - Информације о аплетовом окружењу се могу добити кроз интерфејс AppletContext.

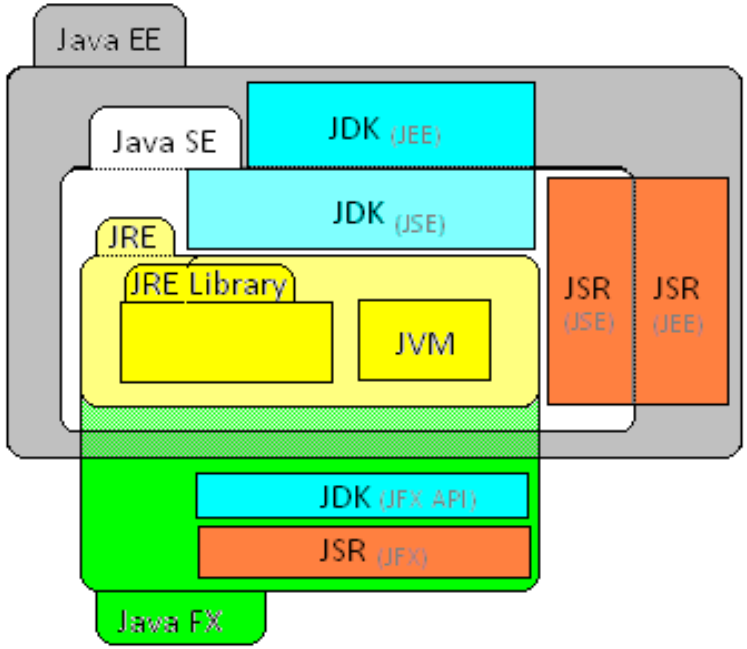
Математички факултет

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

Non-core Java API-ји

Набројаћемо и неколико API-ја који се налазе ван централног API -ја, а који се такође често користе:

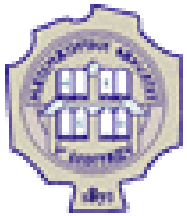
- Enterprise API (укључује JDBC, Java IDL и Java RMI)
- Commerce API (Java Wallet)
- Server API
- Media API
  - Java 2D
  - Java Media Framework
  - Java 3D
- Security API
- Beans API
- Embedded API



Однос разних Java API-ја

Математички факултет

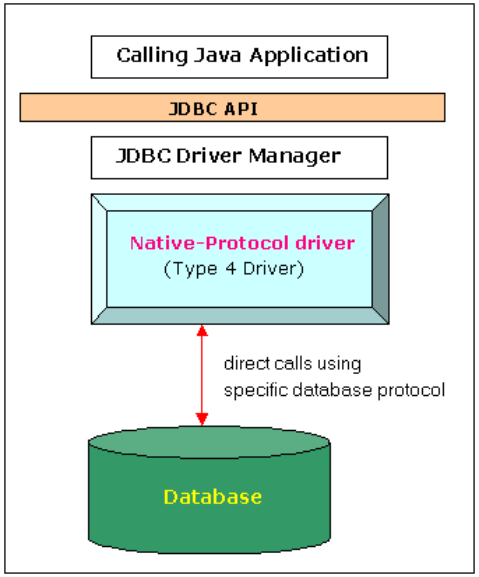
{vladaf,kartelj}@matf.bg.ac.rs

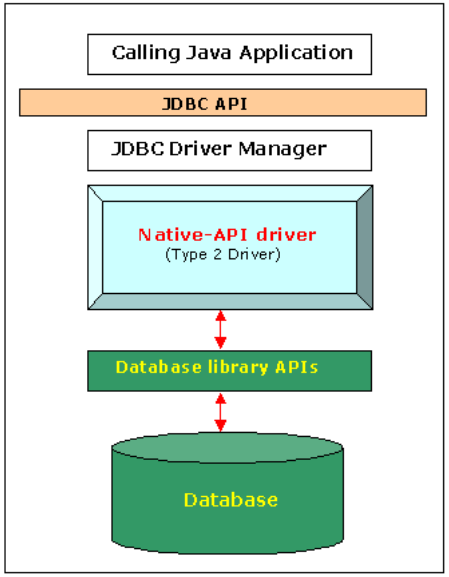


Java Enterprise API

- Java Enterprise API
  - Одржава везу према великим базама података и према наслеђеним апликацијама (eng. legacy applications).
  - Коришћењем овог API-ја развијају се сложени дистрибуисани клијент/сервер аплети и апликације у Јави.
- Java Database Connectivity, или JDBC
  - Стандардни интерфејс за приступ SQL базама података, који обезбеђује унифоман приступ за широк опсег релационих база података.

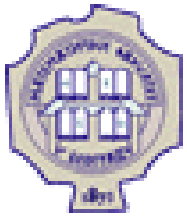
Приступ бази података преко JDBC-а





Математички факултет

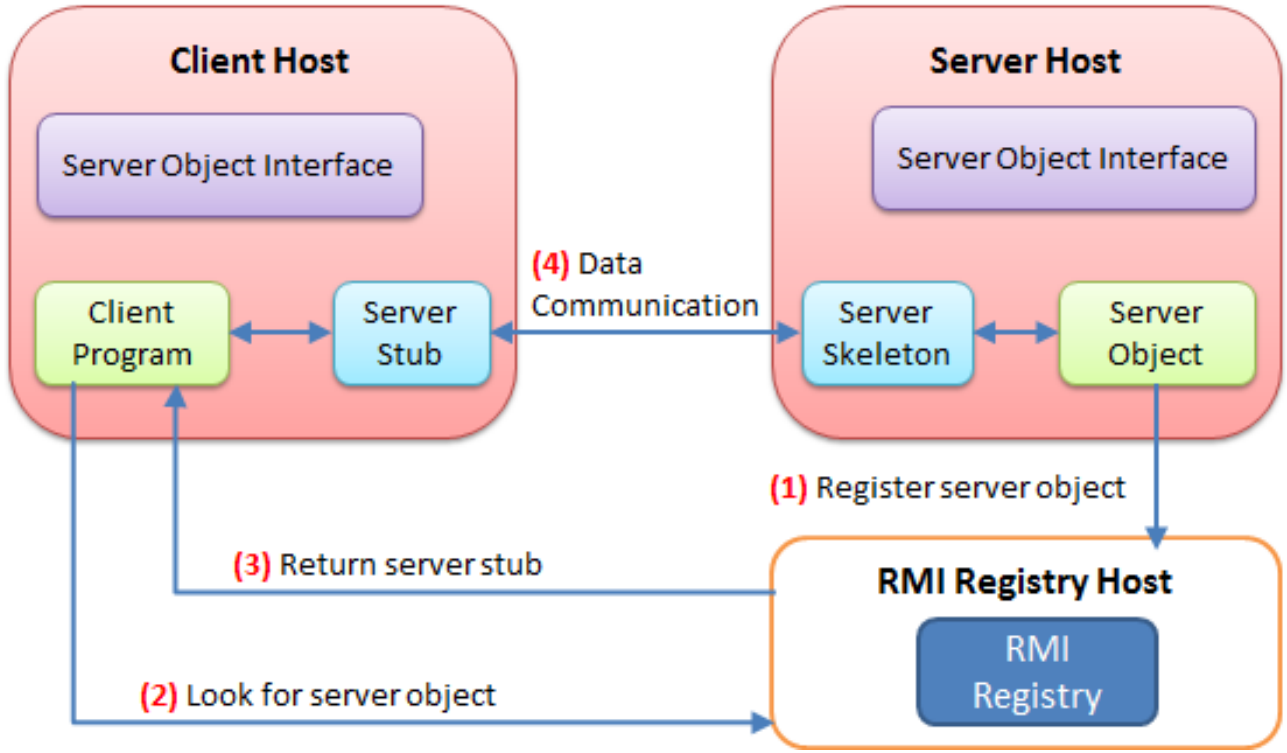
{vladaf,kartelj}@matf.bg.ac.rs



Java Enterprise API (2)

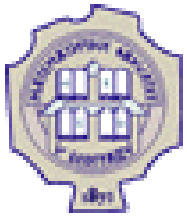
- Java RMI омогућује позивање удаљених метода између чворова или клијента и сервера у случајевима када се на оба краја позива налазе апликације писане у језику Јава.

Илустрација Java RMI



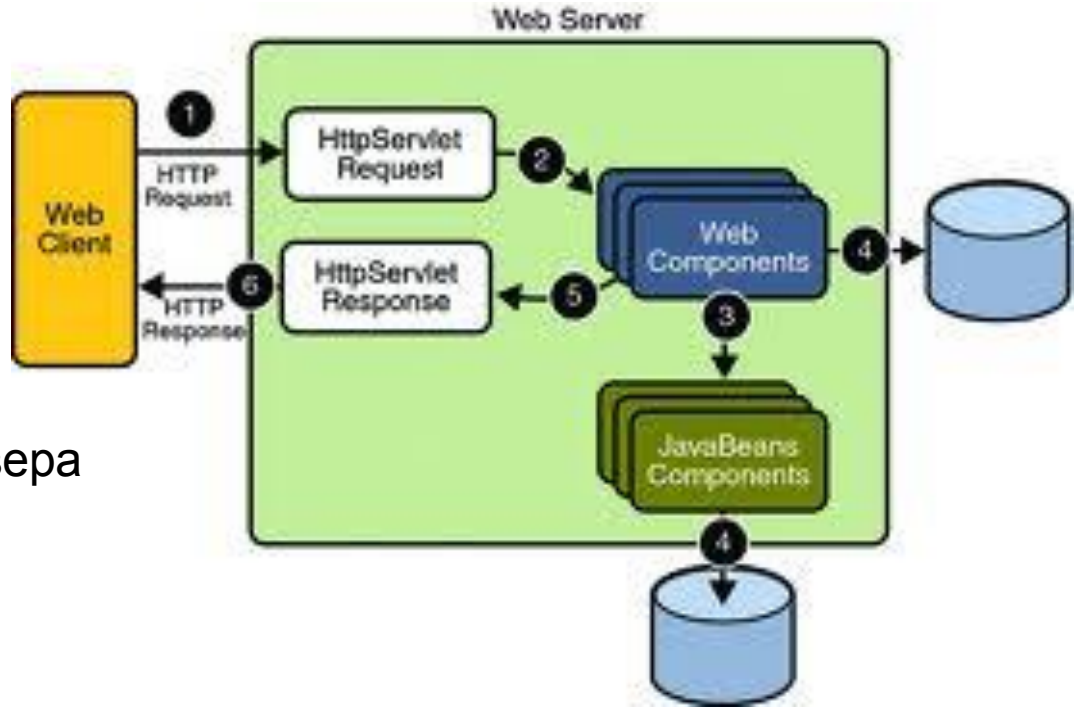
Математички факултет

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

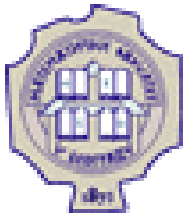


Java Server API

- Java Server API је проширив оквир (енг. framework)
  - Омогућује и олакшава развој Јава-заснованих Интернет и Интранет сервера.
  - Овај API је неопходан програмерима који развијају Јава сервлете.



Илустрација рада веб сервера



Java Media API

- Java Media API лако и флексибилно омогућује програмерима и корисницима да користе мултимедију.
- Media Framework садржи сатове за синхронизацију и медија плејере за приказ аудио, видео и MIDI-ја.

Java Security API

- Java Security API је оквир за програмере који треба да лако и поуздано укључе сигурносну функционалност у аплете и апликације.
- Ово укључује криптографију са дигиталним потписима, енкрипцију и аутентификацију.

Java Beans API

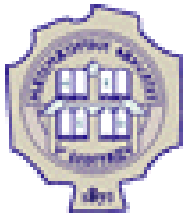
- Java Beans API дефинише преносив скуп API-ја за софтверске компоненте који је независтан од платформе.
- Java Beans је скуп конвенција који омогућава униформан пренос објеката и података између различитих платформи.

Java Embedded API

- Java Embedded API описује како Java API може бити коришћена за уметнуте уређаје (енг. embeded devices).
- Ови уређаји су неспособни за подршку целог централног Java API.
- API укључује минималан ументнути API заснован на java.lang, java.util, и деловима java.io.

Математички факултет

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

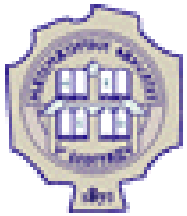


Java Commerce API

- Java Commerce API обезбеђује сигурну куповину и финансијске трансакције на Интернет-у.
- Иницијална компонента је JavaWallet – она дефинише и имплементира функционалност на клијентској страни за кредитне картице, дебитне картице, и трансакције са новцем.

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.