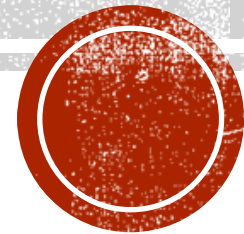


ОБЈЕКТНО ОРИЈЕНТИСАНО ПРОГРАМИРАЊЕ ПРОГРАМСКИ ЈЕЗИК ЈАВА – 1

Карактеристике програмског језика Јава



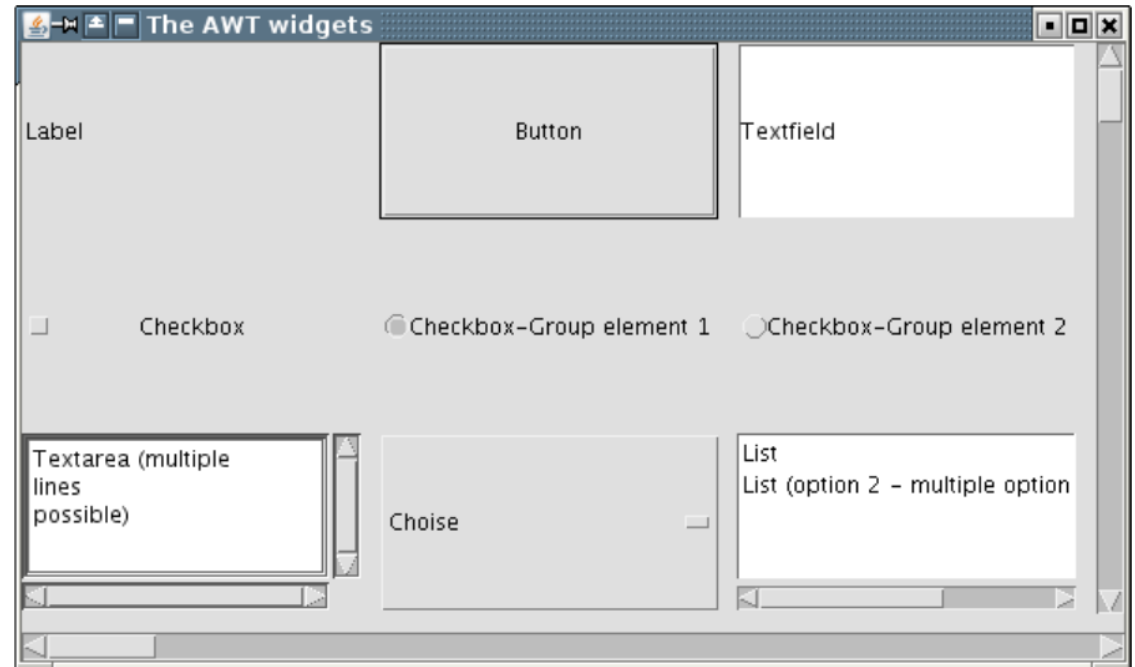
ИСТОРИЈАТ И РАЗВОЈ ЈАВЕ

- Јава обухвата програмски језик и платформу за извршавање и развој.
- Производ компаније Sun Microsystems, сада у власништву компаније Oracle.
- Значајни делови на располагању као софтвер отвореног кода.
- Претеча Јаве настала је 1991. године:
 - намењена мрежном кућном окружењу;
 - пројекат Оак, под руководством Џејмса Гослинга.
- Језик Јава је лансиран 1995. године, на SunWorld конференцији.
 - Моментални успех.
 - Netscape прегледач, IBM, Microsoft...



ЈАВА ДИСТРИБУЦИЈЕ 0-1

- 1997. Sun развија JDK 1.0 (Java Development Kit).
- 1997. појавио се JDK 1.1:
 - JavaBeans;
 - унутрашње класе;
 - JDBC;
 - RMI;
 - рефлексију;
 - AWT — Abstract Window Toolkit;
 - подршку за интернационаизацију, тј. за **Unicode** знакове.



ЈАВА ДИСТРИБУЦИЈЕ 2-3

- 1998. J2SE 1.2:
 - нова номенклатура именовања (раздвајање SE и EE, standard edition, enterprise edition);
 - ME – mobile edition;
 - Swing графичка библиотека;
 - JIT преводац за JVM;
 - Јава IDL за CORBA;
 - колекције (лист, мапе, скупови).
- 2000. J2SE 1.3:
 - садржи подршку за рад у дистрибуисаном окружењу (више рачунара комуницирају) RMI;
 - рад са звуком – JavaSound;
 - JNDI;
 - JPDA,;
 - синтетичке класе-заступници.

ЈАВА ДИСТРИБУЦИЈА 4

- 2002. J2SE 1.4:
 - `assert`;
 - уланчавање изузетака;
 - подршка за IPv6;
 - NIO;
 - `logging API`;
 - рад са сликама преко `Image I/O API`;
 - XML и XSLT процесор;
 - JAXP;
 - JCE криптографију;
 - и `Java Web Start`.

ЈАВА ДИСТРИБУЦИЈА 5

- 2004. J2SE 5.0 (промена у начину бројања):
 - подршка за генеричке типове;
 - сигурност типова за колекције;
 - анотације;
 - аутоматско паковање/распакивање типа;
 - набројиви (енумерисани) типови;
 - променљиви аргументи;
 - колекцијски for циклус;
 - статички увоз;
 - аутоматска подршка за удаљено позивање метода;
 - подршка за паралелно програмирање;
 - класе за парсирање улаза.
- Први пут настале значајније промене у самом језику.
- Изостанак подршке неким застарелим концептима.

ЈАВА ДИСТРИБУЦИЈА 6

- 2006. Java SE 6:
 - убрзање перформанси језгра и **Swing** графичке библиотеке;
 - побољшане веб сервисе коришћењем **JAX-WS**;
 - побољшан рад при повезивању са базама података — подршка за **JDBC 4.0**;
 - укључен **API** за Јава превођење;
 - архитектура за **XML** повезивање (**JAXB**);
 - усавршен **GUI**;
 - побољшана **JVM**.
- 2006. велики део Јаве постаје слободан, отворен и доступан је под **GPL** лиценцом.

ЈАВА ДИСТРИБУЦИЈА 7

- Компанија Oracle 2010. године купује Јаву од од Sun-а.
- 2011. Java SE 7:
 - JVM подршка за динамичке језике;
 - компресовани 64-битни показивачи;
 - нова I/O библиотека са подршком за мета-податке;
 - XRender ток за Java 2D који убрзава цртање модерним GPU;
 - знаковне ниске као обележја у **switch** наредби;
 - аутоматско управљање ресурсима код **try-catch**;
 - простије декларисање метода са променљивим параметрима;
 - литерали који представљају бинарне бројеве;
 - подвлаке у нумеричким литералима;
 - симултано хватање више врста изузетака
 - и избацавање изузетака уз побољшану контролу типа.

ЈАВА ДИСТРИБУЦИЈЕ 8-9

- 2014. Java SE 8:
 - подршка за рад са ламбда изразима (аспекти функционалног програмирања);
 - побољшани рад са временом и календарима;
 - ефикаснији Nashorn JavaScript модул;
 - побољшања на пољу сигурности.
- 2017. Java SE 9:
 - платформа система модула (позната и као пројекат Jigsaw);
 - дата конфигурација може садржати један мањи скуп модула, а не мора садржавати целокупну Јава платформу као монолитни систем, као раније;
 - боље перформансе,
 - лакше одржавање
 - и сигурност Јава апликације.

ЈАВА ДИСТРИБУЦИЈА 10

- Почев од верзије 9, компанија Oracle дистрибуира OpenJDK.
 - Софтвер отвореног кода (слично као Линукс).
 - Идеја је да оба производа: Oracle JDK и OpenJDK могу да замене један другог.
 - Раздвајање комерцијалне и некомерцијалне употребе.
- Март 2018. Јава 10 (мало новина због нове политика брзих дистрибуција):
 - подршка за одређивање типа локалне променљиве;
 - унапређење скупљача отпадака;
 - Јава EE почиње да се развија независно под називом Jakarta EE (Eclipse Foundation).

ЈАВА ДИСТРИБУЦИЈА 11

- Септембар 2018. Јава 11 LTS.
 - Прва верзија Јаве после верзије 8 са дурогочном подршком (енг. **long term support**).
 - Нови скупљач отпадака, тзв. **Epsilon garbage collector**.
 - Промењен формат датотека са бајт кодом.
 - Уклоњени су **Java EE** и **CORBA** модули из **Java SE** платформе и из **JDK**.
 - **JavaFX** је постао софтвер отвореног кода, који се испоручује као самостална библиотека.
- Мења се начин појаве нових верзија Јаве — одлучено је да ће:
 - појава нових верзија бити чешћа,
 - а само неке међу њима ће бити дугорочно подржане (**LTS**),
 - док ће остале верзије бити подржаване само до изласка нове верзије.

ЈАВА ДИСТРИБУЦИЈЕ 12-14

- Март 2019. Јава 12:
 - експериментални скупљач отпадака са кратком паузом;
 - **switch** наредба добија могућност да чини део израза;
 - унапређење процеса компилације **JDK**, итд.
- Септембар 2019. Јава 13:
 - веће измене заправо биле поправке претходне верзије;
 - уведени су текстуални блокови који омогућавају лакши рад са вишелинијским текстом при писању кода.
- Март 2020. Јава 14 је донела доста унапређења и поправки:
 - интеграција регуларних израза у **instanceof** наредбу;
 - нови алат за паковање (**Incubator**);
 - информативнији изузеци у случају реферисања **null** референце;
 - прилагођавање **ZGC** скупљача отпадака за рад у **macOS** и **Windows** оперативним системима.

ЈАВА ДИСТРИБУЦИЈЕ 15-17

- Септембар 2020. Јава 15:
 - увођење запечаћених (**sealed**) класа и интерфејса, које постављају ограничења на то које друге класе или интерфејси их могу наслеђивати.
- Март 2021. Јава 16:
 - **Vector API** за унапређена векторска израчунавања;
 - миграција **OpenJDK** пројекта на **Git** систем.
- Септембар 2021. Јава 17 **LTS**:
 - друга по реду **LTS** верзија после верзије 11 из септембра 2018. године;
 - побољшани рад са генераторима псеудослучјаних бројева;
 - ојачавање енкапсулације интерних **JDK** библиотеке;
 - запечаћене класе (енг. **sealed classes**);
 - подршка за рад са регуларним изразима у оквиру **switch** наредбе итд.

ЦИЉЕВИ ПРИ РАЗВОЈУ ЈАВЕ

- **Једноставност, објектна оријентисаност, фамилијарност**
 - Садржи готове библиотеке за најразличитије намене.
 - Објектно оријентисан од самог почетка.
 - По синтакси сличан C/C++.
- **Робусност и сигурност**
 - Омогућава креирање веома поузданог софтвера:
 - интензивна провера током компилације,
 - провера током извршавања програма.
 - Модел управљања меморијом једноставан:
 - нема показивача,
 - нити показивачке аритметике.



Џејмс Гослинг, вођа тима који је креирао Јаву

ЦИЉЕВИ ПРИ РАЗВОЈУ ЈАВЕ (2)

- **Архитектонска неутралност и преносивост**
 - Садржи компајлер који преводи до нивоа бајт-кѡда:
 - бајт-кѡд није исто што и машински кѡд,
 - међуформат који је архитектонски неутралан,
 - машински кѡд није архитектонски неутралан, зависи од процесора,
 - преносив на различите врсте процесора и оперативних система.
 - Стриктно се дефинише основни језик:
 - величине простих типова увек исте без обзира на оперативни систем и архитектуру процесора — на пример, тип за запис означеног целог броја је увек 4 бајта (у С-у ово не важи).
 - Има исто извршавање на свакој платформи:
 - за дате улазне податке даје исте излазне податке, што не важи за програмски језик С.

ЦИЉЕВИ ПРИ РАЗВОЈУ ЈАВЕ (3)

- **Перформансе**

- Програм се компајлира до бајт-кода, а потом интерпретира.
- Интерпретер ради пуном брзином, јер су сигурносне провере обављене раније.
- Постоји аутоматски скупљач отпадака, па програмер не мора да ослобађа меморију.
- Секције са интензивним рачунањем могу да се извезу директно у машински код.

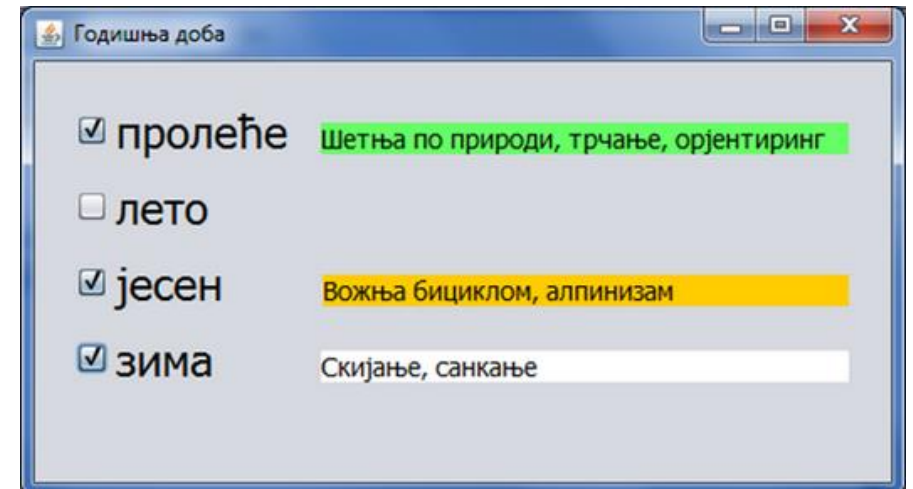
- **Интерпретираност, вишенитност и динамичност**

- Интерпретатор извршава бајт-код на било ком рачунару где постоји систем за извршавање.
- Подржава вишенитно извршавање:
 - нпр. веб прегледач, мора „истовремено“ да:
освежава графичке компоненте, учитава страницу и преузима датотеку.
- Динамички учитава класе у току извршавања
— класе се повезују (линкују) само када је то потребно.

ТИПОВИ ЈАВА АПЛИКАЦИЈА

ДЕСКТОП АПЛИКАЦИЈЕ

- Развој преносивих апликација са графичким корисничким интерфејсом (енг. **Graphical User Interface**) – ГУИ.
- Најпопуларније библиотеке су:
 - AWT,
 - Swing,
 - SWT
 - и Java FX.
- Поједини произвођачи софтвера су креирали сопствене Јава ГУИ библиотеке
 - Нпр. IBM Јава програмерима понудио **swt** библиотеку.



ТИПОВИ ЈАВА АПЛИКАЦИЈА АПЛИКАЦИЈЕ ИЗ КОМАНДНЕ ЛИНИЈЕ

- Апликације из командне линије не користе графичке компоненте.
- То, међутим, не нарушава изражајност саме апликације.
- Унос и испис се врше путем командне линије уместо путем текстуалних поља, лабела итд.



```
Administrator: Command Prompt
P:\Temp>java Duzina
Unesite rastojanja a i b
12 56
d = 44.0
P:\Temp>_
```

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window content displays the execution of a Java application named "Duzina". The user enters the command "java Duzina" at the prompt "P:\Temp>". The application then prompts the user to enter two numbers: "Unesite rastojanja a i b". The user enters "12 56". The application outputs "d = 44.0". The prompt returns to "P:\Temp>_". A red oval highlights the entire execution sequence from the command to the final prompt.

ТИПОВИ ЈАВА АПЛИКАЦИЈА

АПЛИКАЦИЈЕ ЗА МОБИЛНЕ УРЕЂАЈЕ

- Може се рећи да је Јава званичан језик за развој **Android** софтвера.
 - Јава има највећу подршку од стране произвођача **Android** система (компаније **Google**)
 - и највећи број апликација, које се налазе на **Google Play Store**, креиран коришћењем Јаве.
- Наравно, за писање апликација за **Android** уређаје је, поред језика Јаве, потребно:
 - познавати и друге елементе — развојно окружење (нпр. **Android Studio**),
 - библиотеке, тј. алат за развој **Android SDK**,
 - алат **Gradle**,
 - структуру датотека **Android Manifest** и језика за означавање **XML**.

ТИПОВИ ЈАВА АПЛИКАЦИЈА

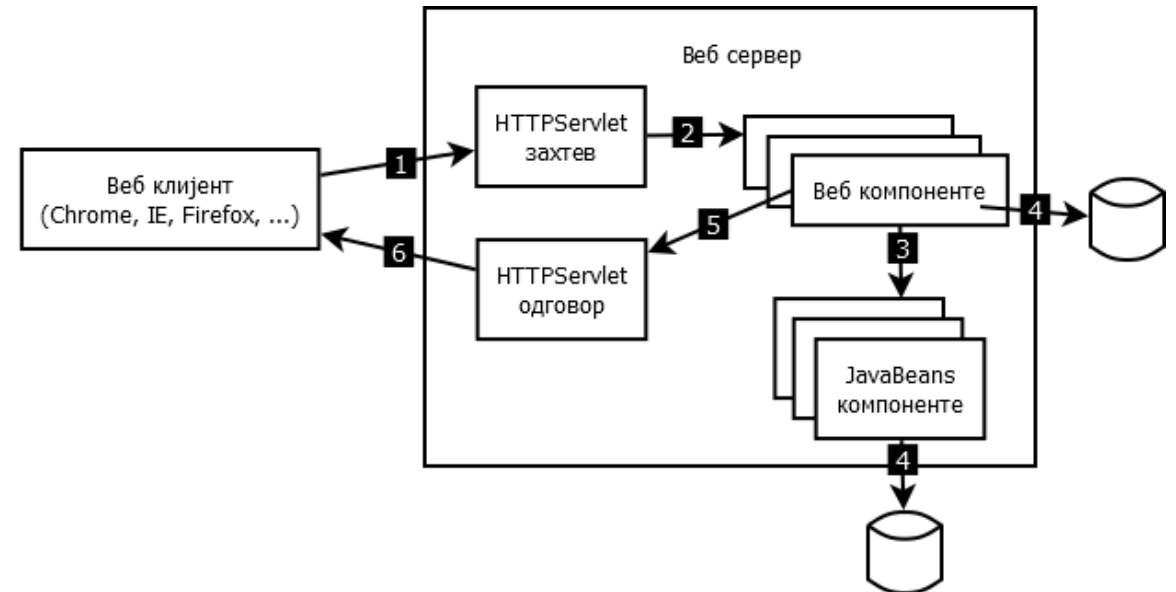
АПЛЕТИ

- Јава аплети представљају пример тзв. веб програмирања на клијентској стран.
- У овом моменту застарела технологија и ретко се користи.
- Програм, аplet се, дакле, преузме са сервера, а потом се извршава на клијенту (прегледачу).

ТИПОВИ ЈАВА АПЛИКАЦИЈА

СЕРВЛЕТИ И ЈАВА СЕРВЕРСКЕ СТРАНИЦЕ

- За разлику од аплета овде је у питању Јава извршавање на страни сервера.
- Сценарио је следећи:
 - захтев стиже од клијента,
 - на веб серверу се извршавају наредбе,
 - потом веб сервер генерише одговор и шаље га клијенту.



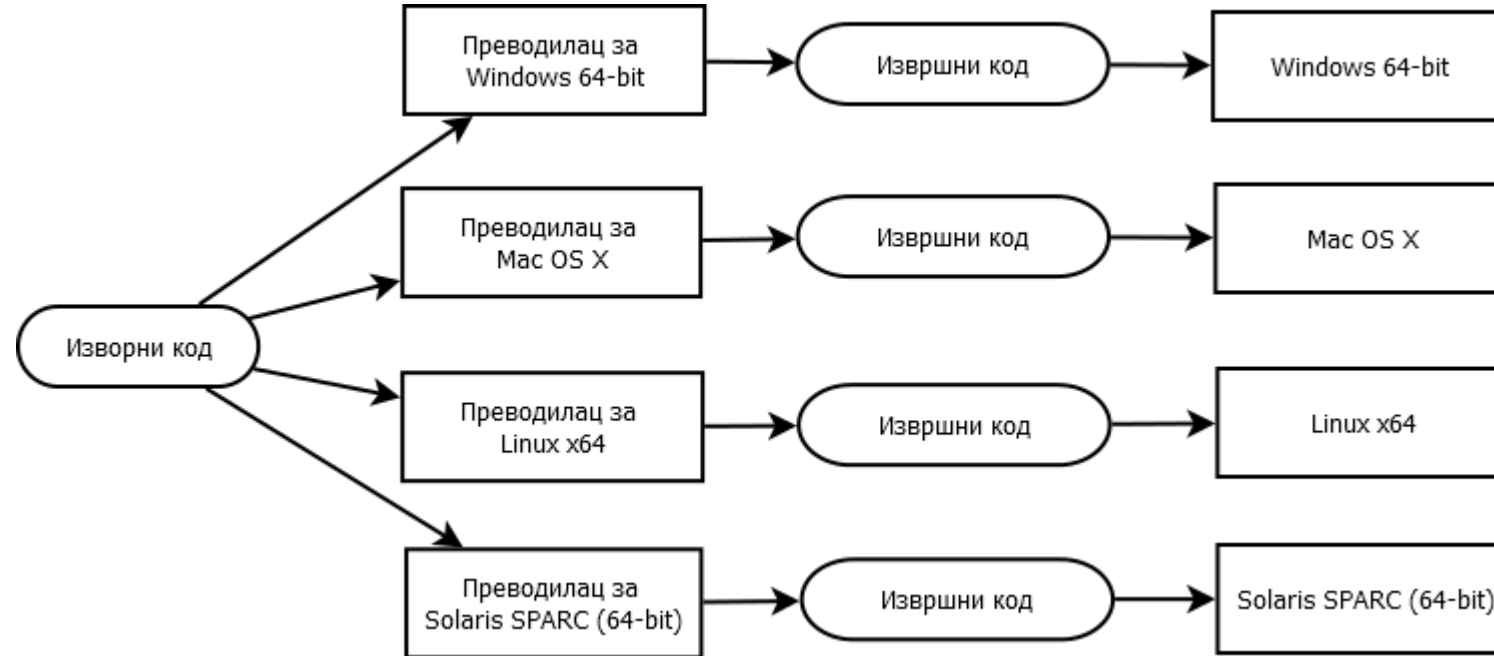
ТИПОВИ ЈАВА АПЛИКАЦИЈА ВЕБ СЕРВИСИ

- Веб сервиси омогућавају комуникацију између апликација које се извршавају на разноврсним платформама (помоћу **HTTP(S)** протокола).
- Описи веб сервиса су задати најчешће као **XML** датотека.
 - Ово омогућава њихову проширивост и динамичност.
- Обично се граде хијерархије веб сервиса.
 - Они софистициранији користе услуге једноставнијих.
- Основна предност веб сервиса је интероперабилност.
 - Тј. повезивање разноврсних (хетерогених) софтверских система на елегантан начин.

ТИПОВИ ЈАВА АПЛИКАЦИЈА БИБЛИОТЕКЕ КЛАСА

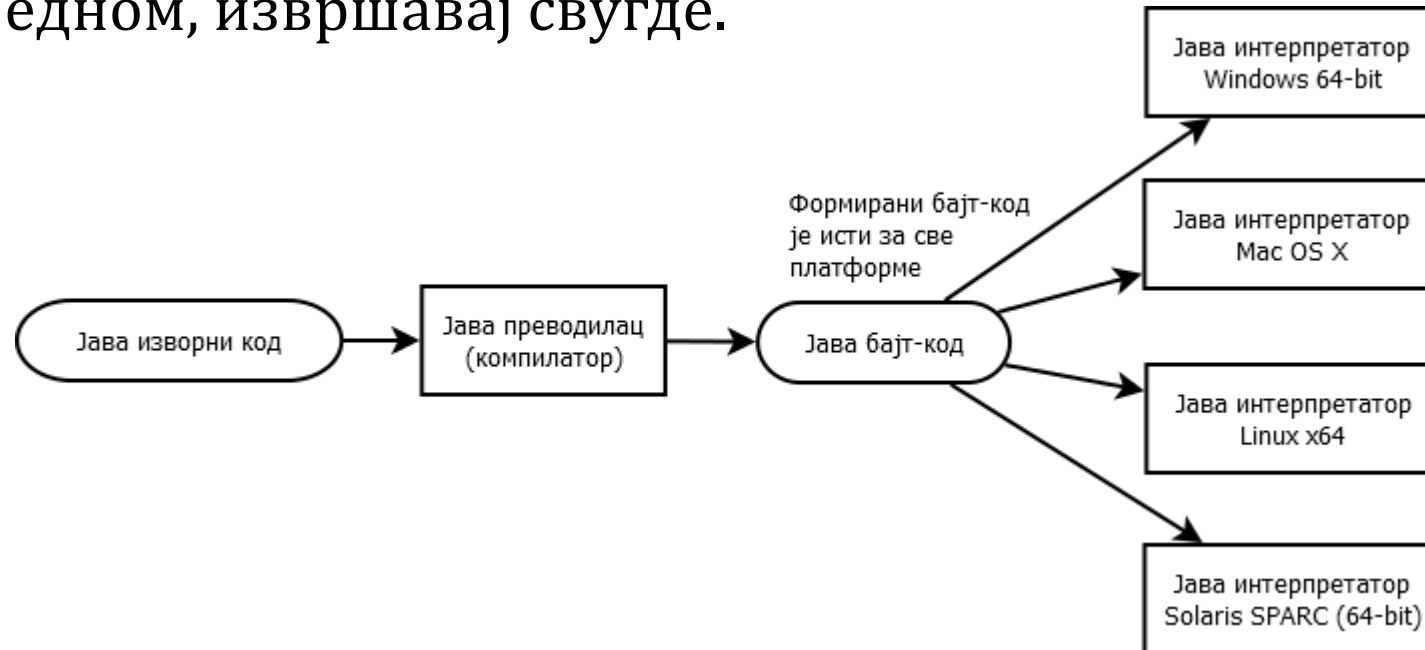
- Има смисла правити библиотеку од компоненти:
 - које ће се више пута користити у различитим програмима,
 - а које одређују неку функционалност.
- Програмер те компоненте може:
 - спаковати у своју библиотеку
 - и касније их користити у новим пројектима.
- Библиотека обично садржи већи број сродних функционалности, нпр.:
 - библиотека за рад са текстом,
 - библиотека за везу према базама података и слично.

ПРОЦЕС ПРЕВОЂЕЊА И ИЗВРШАВАЊА (ПРАВИ ПРЕВОДИЛАЦ)

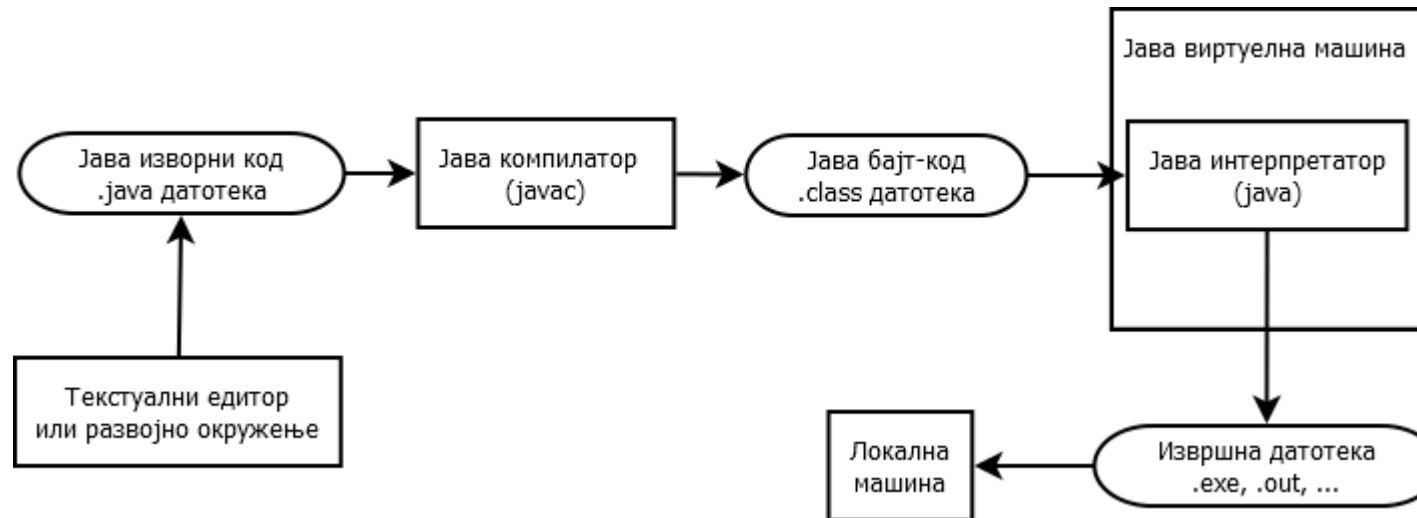


ПРОЦЕС ПРЕВОЂЕЊА И ИЗВРШАВАЊА (JAVA)

„Напиши једном, извршавај свугде.“

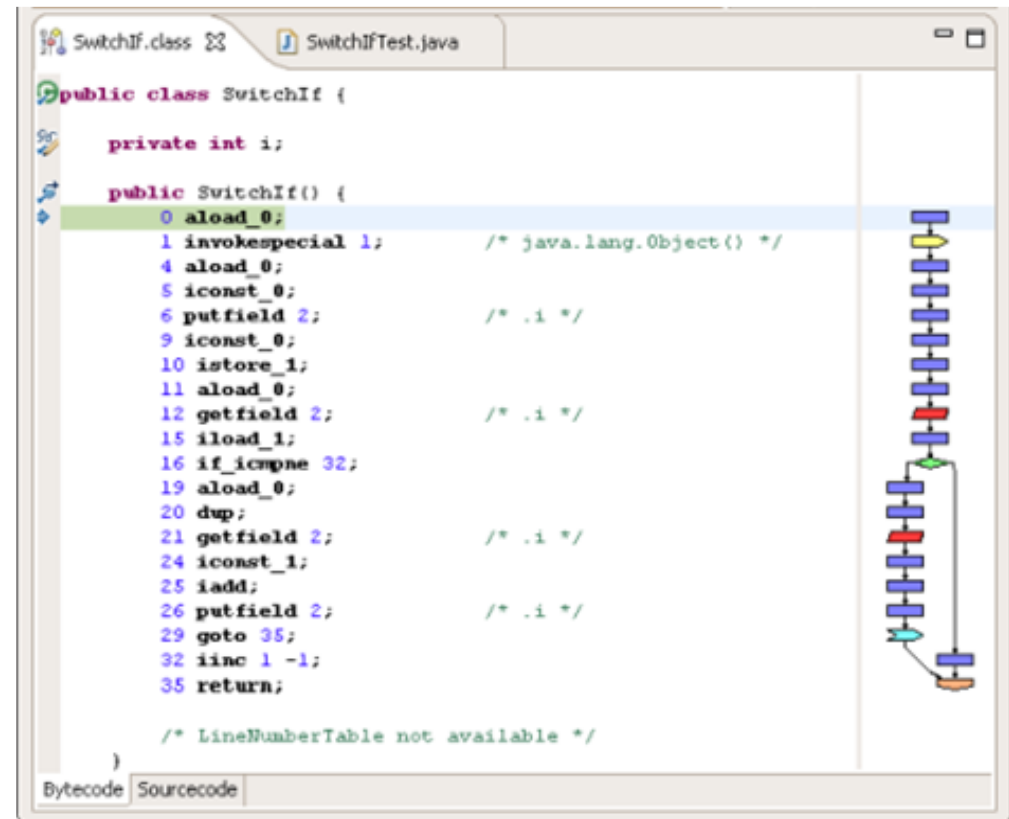


ПРОЦЕС ПРЕВОЂЕЊА И ИЗВРШАВАЊА ДЕТАЉИ (JAVA)



ЈАВА БАЈТ (МЕЋУ) КОД

- По нивоу апстракције близак асемблерским језицима.
- Јава је „више“ компајлира него што је интерпретиран.
- Већи део пута од изворног језика (Јаве) до циљног (машинског) се пређе у компилацији.
 - Боља ефикасност.

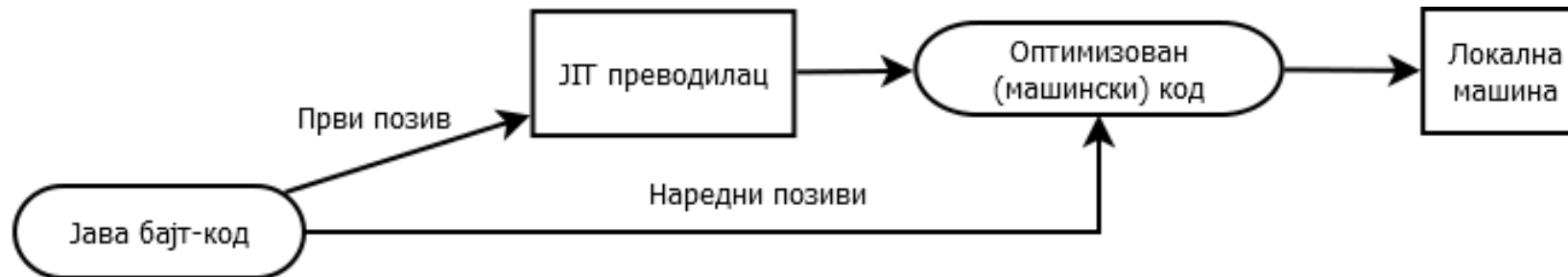


```
public class SwitchIf {  
    private int i;  
    public SwitchIf() {  
0      aload_0;  
1      invokespecial 1;      /* java.lang.Object() */  
4      aload_0;  
5      iconst_0;  
6      putfield 2;          /* .i */  
9      iconst_0;  
10     istore_1;  
11     aload_0;  
12     getfield 2;          /* .i */  
15     iload_1;  
16     if_icmpne 32;  
19     aload_0;  
20     dup;  
21     getfield 2;          /* .i */  
24     iconst_1;  
25     iadd;  
26     putfield 2;          /* .i */  
29     goto 35;  
32     inc 1 -1;  
35     return;  
    }  
    /* LineNumberTable not available */  
}
```

JIT ЈАВА ПРЕВОДИЛАЦ

- И поред тога што је ближа компајлираним језицима, Јавина преносивост ипак изазива одређени губитак перформанси.
 - У процесу интерпретирања бајт-кода превођење у машинске инструкције за конкретну платформу врши се више пута.
- Минимална јединица превођења је метод, односно његов придружени бајт-код.
 - Позивање метода више пута (нарочито код рекурзије) захтева поновну интерпретацију.
- JIT је попут других компилатора који производе машински језик на излазу.
 - Међутим, превођење се одвија тек по потреби, а не унапред.
 - Друга разлика је томе што је овде на улазу бајт-код, а не полазни изворни код.
- Приликом превођења бајт-кода до машинског језика за конкретну платформу, JIT преводац може да врши разне компилаторске оптимизације над методама.
 - Ниво оптимизације треба да буде усаглашен са “значајем” метода, динамички ради JIT.
 - Када се неки метод преведе помоћу JIT, машински код се се памти, тј. уписује у својеврсни кеш.

JIT ЈАВА ПРЕВОДИЛАЦ (2)



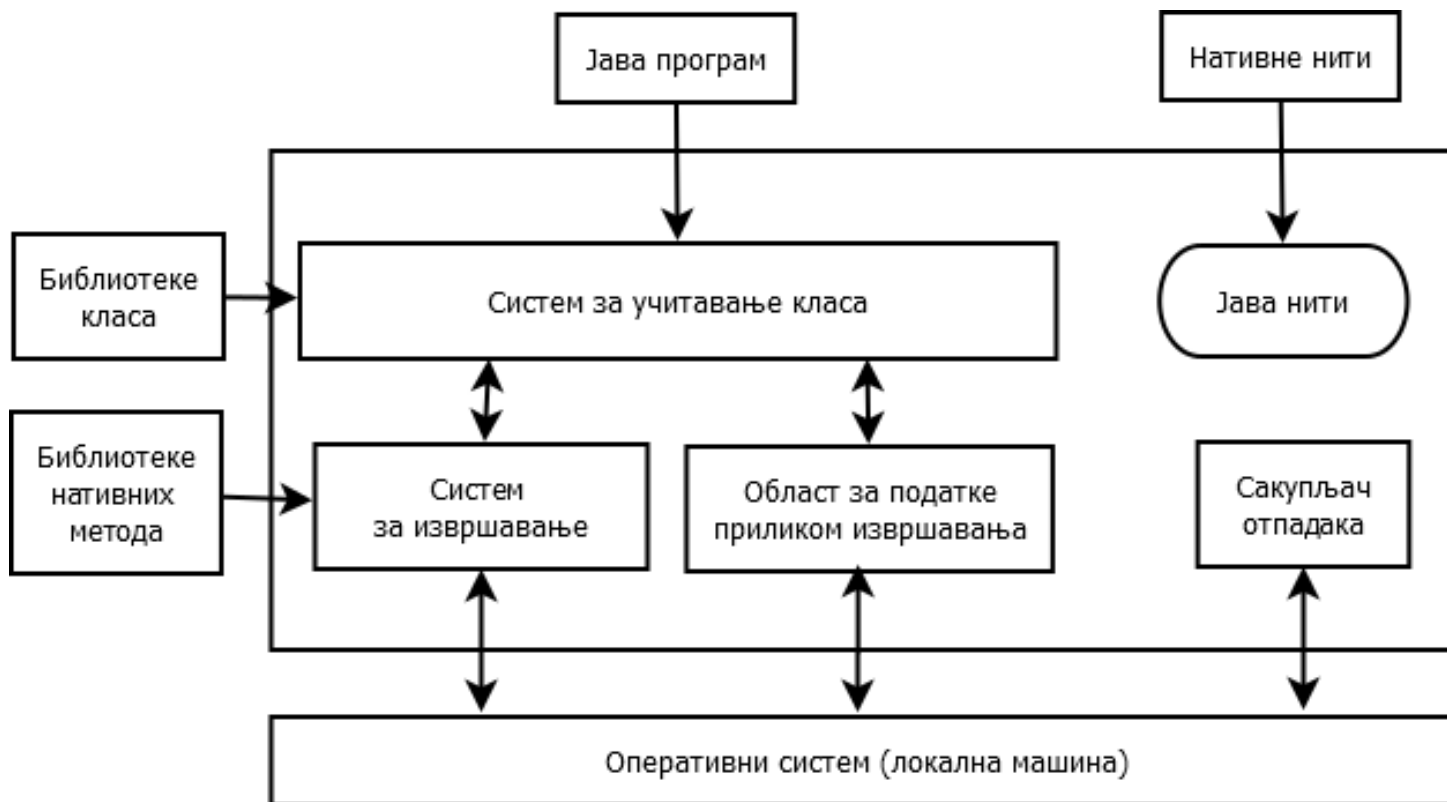
ЈАВА ВИРТУАЛНА МАШИНА

- Језгро Јаве је Јава виртуелна машина (eng. Java Virtual Machine — JVM).
- Виртуални рачунар који постоји само у меморији.
- JVM допушта да програми буду извршавани на разним платформама (портабилност).
 - Потребно је да JVM 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 уређајима.
 - Цео језик Јава је оригинално развијан тако да се на уму има и кућна електроника.
 - Увођењем платформе система модула, почев од Јава 9, величина окружења може додатно да се смањи, тако да буду обухваћени само они модули који ће стварно бити коришћени.
- JVM чита ток бајт-кóдова из .class датотеке као секвенцу машинских инструкција.
- Извршавање бајт-кóда унутар JVM опонаша извршавање машинских инструкција.
 - Код процесора на улазу машинске инструкције, а на излазу је микрокод (контролни сигнали).
 - Код JVM на улазу бајт-кóд, а на излазу машински језик за конкретну платформу.
 - JVM чисто софтверски док је процесор, наравно, хардвер.

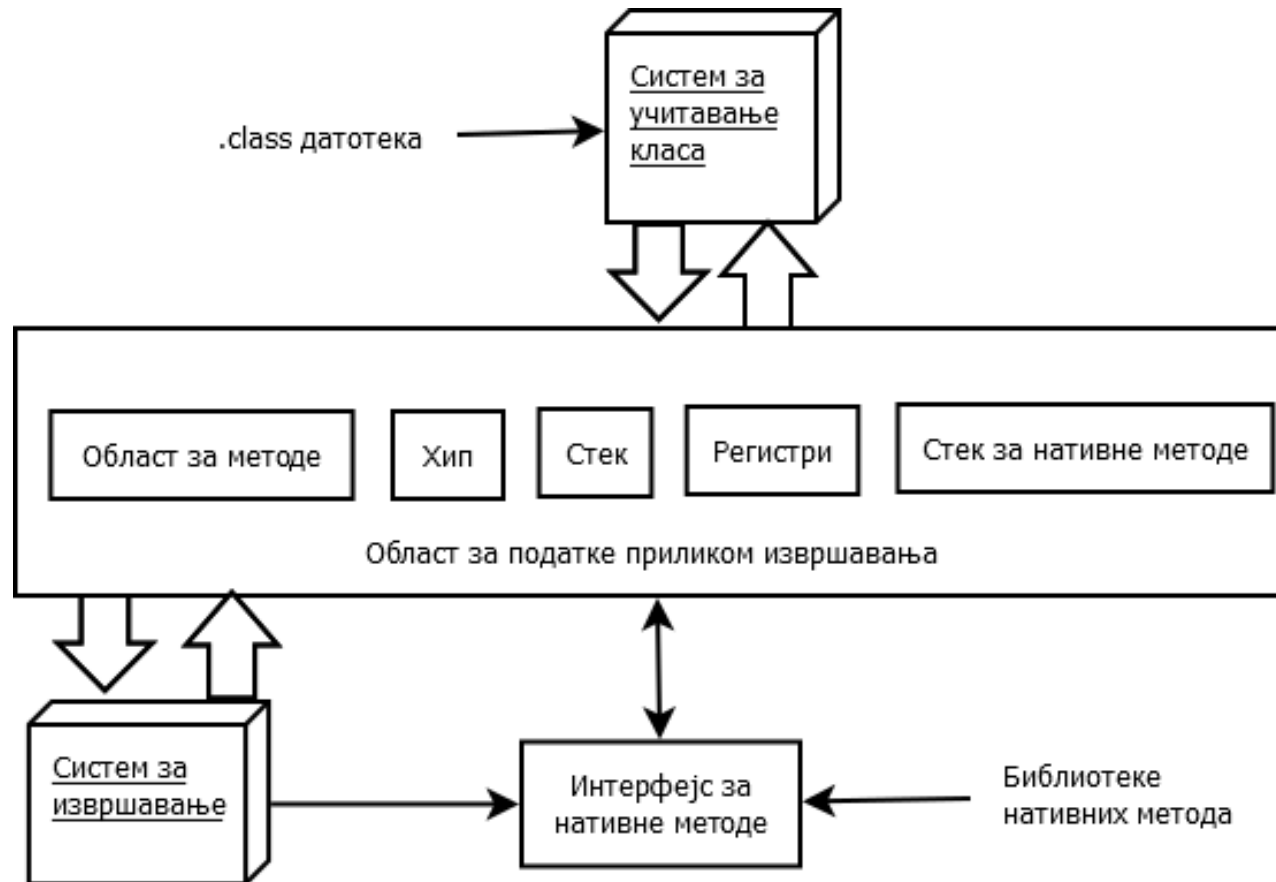
АРХИТЕКТУРА JVM

- Архитектура JVM одсликава архитектуру конкретног рачунарског система.
- Свака од инструкција JVM је слична асемблерској инструкцији.
 - Једнобајтни операциони код.
 - И нула, један или више операнада.
- JVM садржи:
 - систем за читавање класа,
 - систем за извршавање,
 - област за податке приликом извршавања,
 - скупљач отпадака
 - и Јава нити.

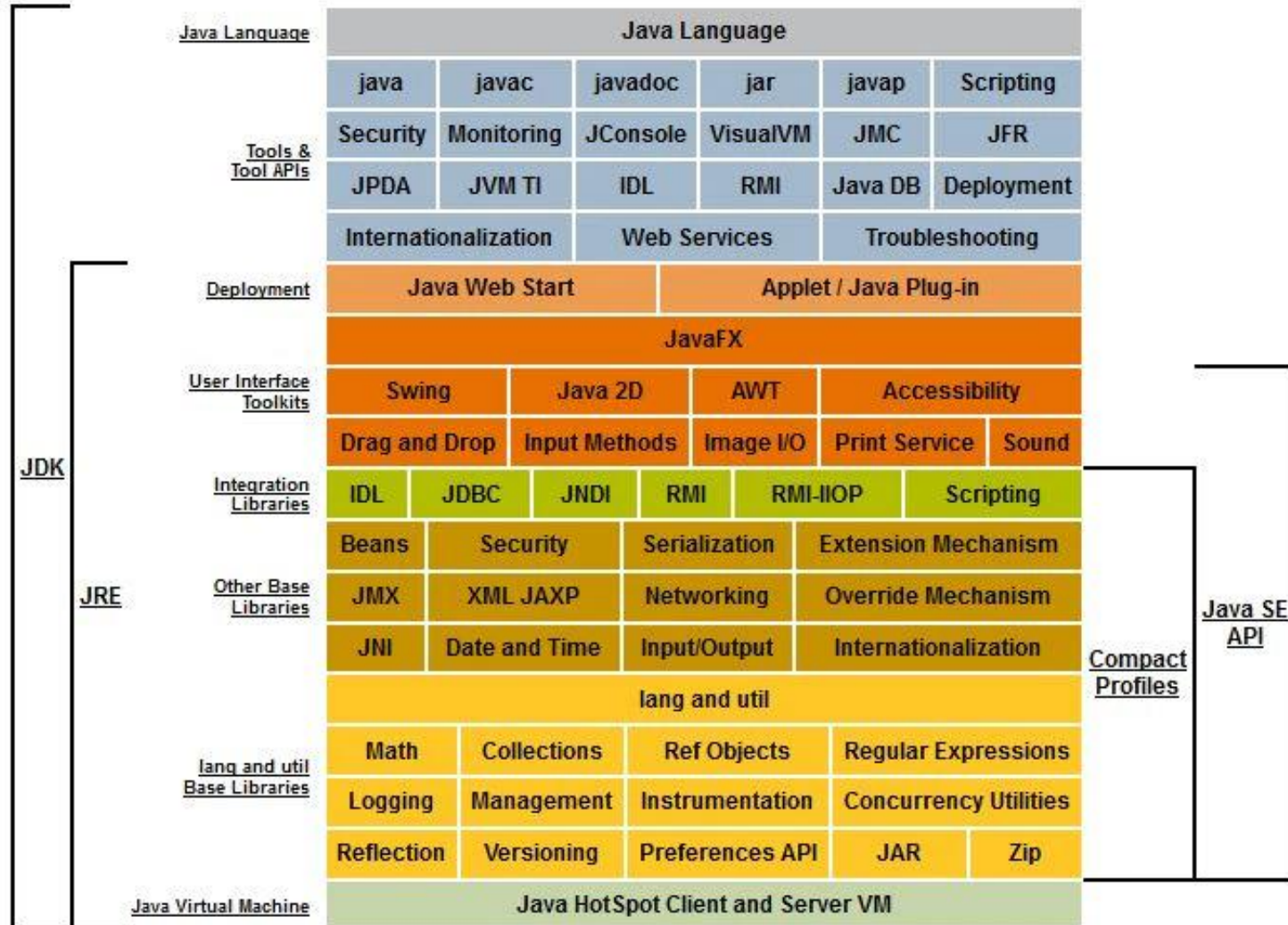
АРХИТЕКТУРА JVM



МЕМОРИЈА JVM



JAVA АЛАТИ ЗА РАЗВОЈ - JDK

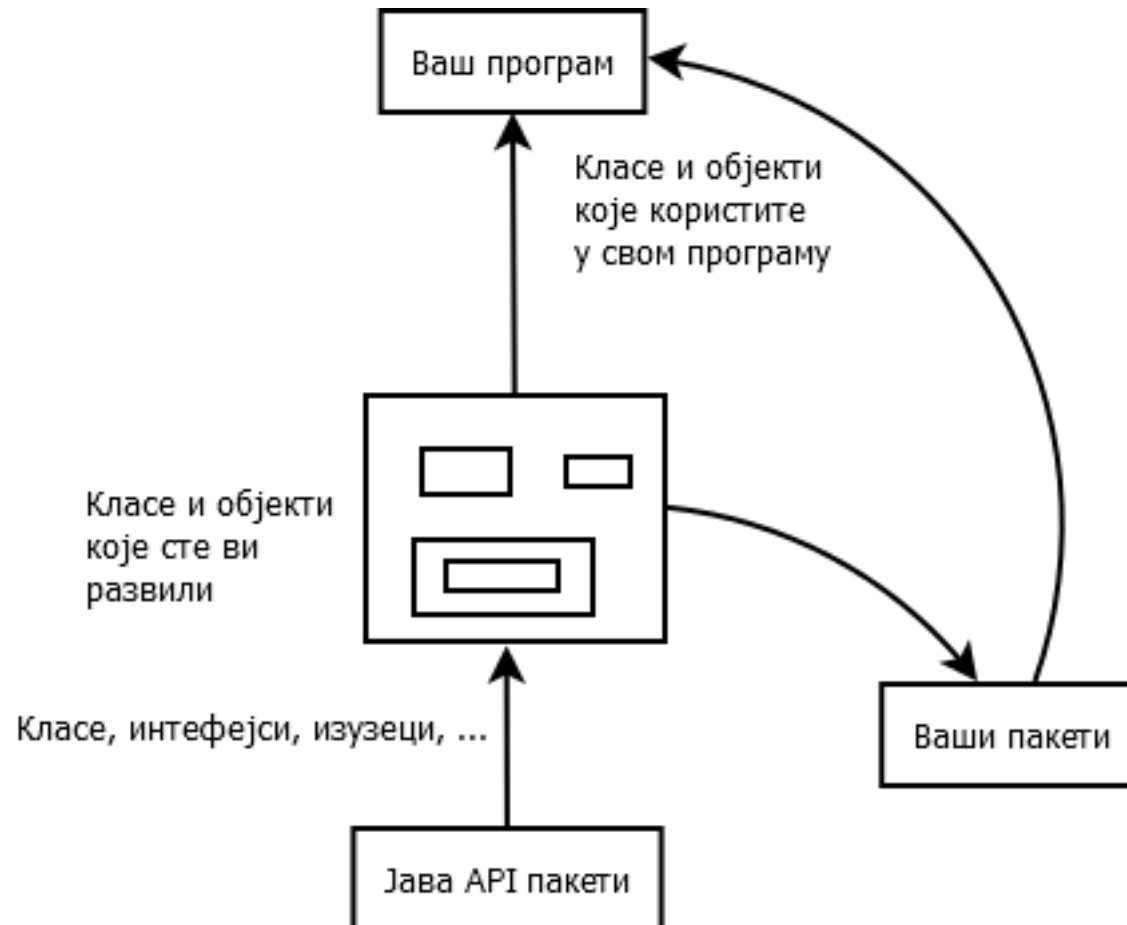


JAVA API

- Јава интерфејс за програмирање апликација (енг. Java Application Programming Interface) тј. Јава API.
 - Скуп класа које је развио Sun (а надоградио Oracle), за коришћење у језику Јава.
- Класе унутар Јава API су груписане у пакете (као директоријуми).
- Саме класе одговарају датотекама унутар тих директоријума.



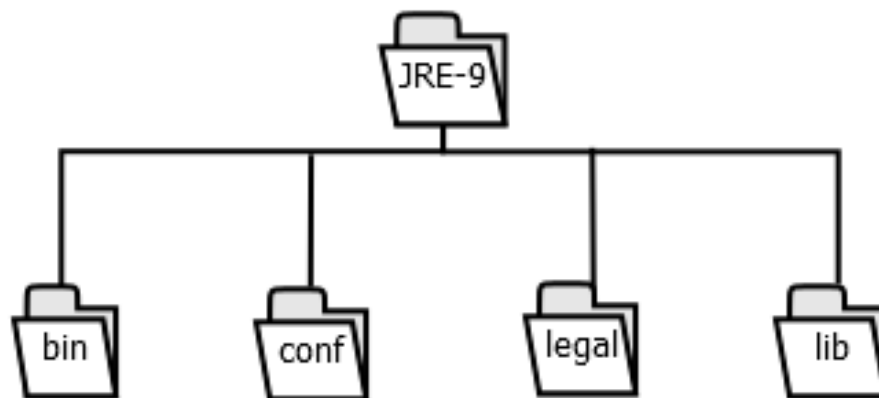
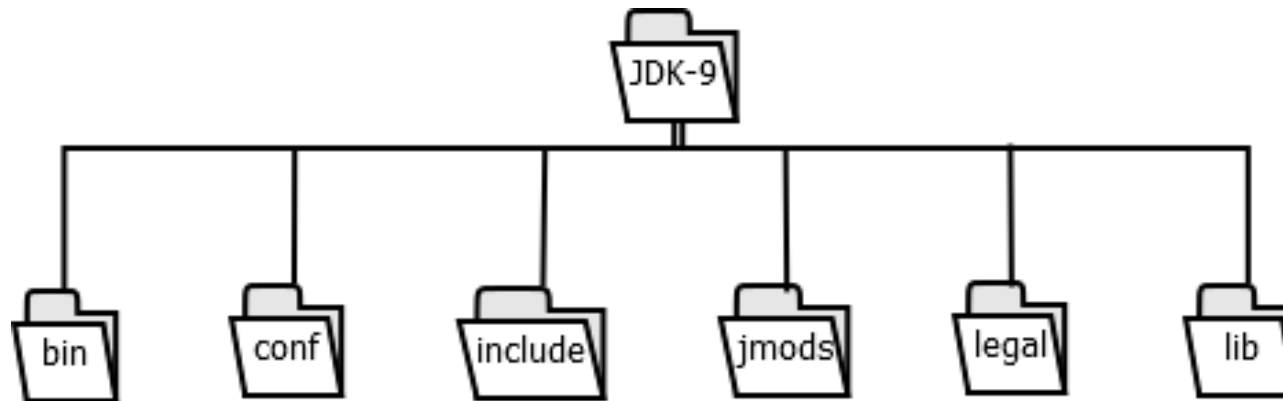
JAVA API (2)



ЈАВА МОДУЛИ

- Омогућавају елегантнију организацију Јава апликација.
- У оквиру Јава модула се може дефинисати:
 - који од Јава пакета, од којих се састоји апликација, могу бити виљиви другим модулима;
 - и који Јава модули су за извршавање Јава апликације.
- Предности:
 - величина испорученог бајт-кода;
 - учауривање интерних пакета (сакривени пакети, недоступни изван припадајућег модула);
 - детекција недостајућих модула при покретању.
 - (Пре Јаве 9, недостајући програмски код детектован тек при покушају употребе у току извршавања.)

СТРУКТУРА JDK



НЕКИ ЈДК МОДУЛИ

Модул	Кратак опис
<code>jdk.charsets</code>	Подршка за карактерске скупове који нису у <code>java.base</code> (најчешће карактери дужине два бајта и IBM карактерски скупови).
<code>jdk.dynalink</code>	Дефинише API за динамичко везивање операција високог нивоа над објектима.
<code>jdk.javadoc</code>	Дефинише имплементацију алата за генерисање документације и његовог конзолног еквивалента, наредба <i>javadoc</i> .
<code>jdk.jcmd</code>	Дефинише алате за дијагностику и решавање проблема у JVM, као што су <i>jcmd</i> , <i>jps</i> , и <i>jstat</i> алати.
<code>jdk.jdi</code>	Дефинише интерфејс за дебаговање.
<code>jdk.net</code>	Дефинише API за мрежно програмирање.
<code>jdk.security.auth</code>	Имплементације сигурносних интерфејса и разноврсних модула за аутентификацију.

ЦЕНТРАЛНИ ЈАВА АРІ

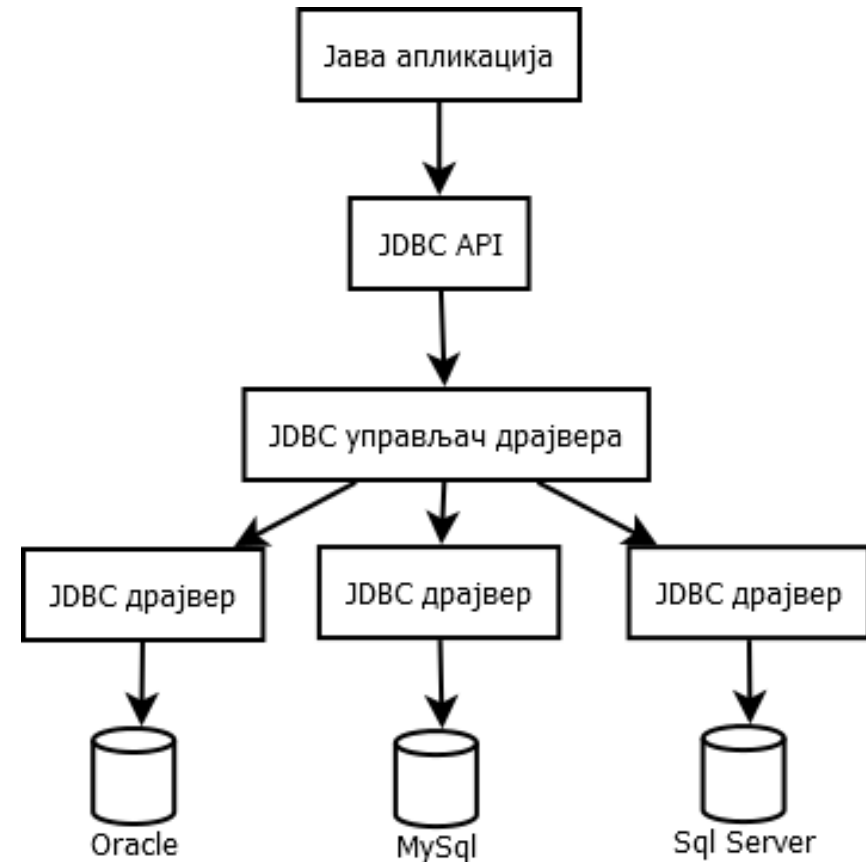
- `java.lang` — састоји се од класа које су централне за језик Јава.
- `java.io` — стандардна улазно/излазна Јава библиотека.
- `java.util` — садржи већи број корисних:
 - за рад са датумима,
 - за структурисање података, као што су `Stack` и `Vector`,
 - као и класе које омогућавају парсирање улазног тока података.
- `java.net` — овај пакет чини језик Јава мрежно заснованим језиком.
- `java.awt` — назив означава скраћеницу за **Abstract Window Toolkit**.
 - Стандардна библиотека за прављење ГУИ-ја, са графичким компонентама попут `Button`, `TextField` итд.
- `javax.swing` — напреднија ГУИ библиотека, садржи класе као што су `JButton`, `JTextField` итд.
- `java.applet` — овај пакет је најмањи пакет у Јава АРІ, и данас се ретко користи.
 - У њему је дефинисана класа `Applet`, која омогућује рад са Јава аплетима.

ДОДАТНИ ЈАВА АРІ

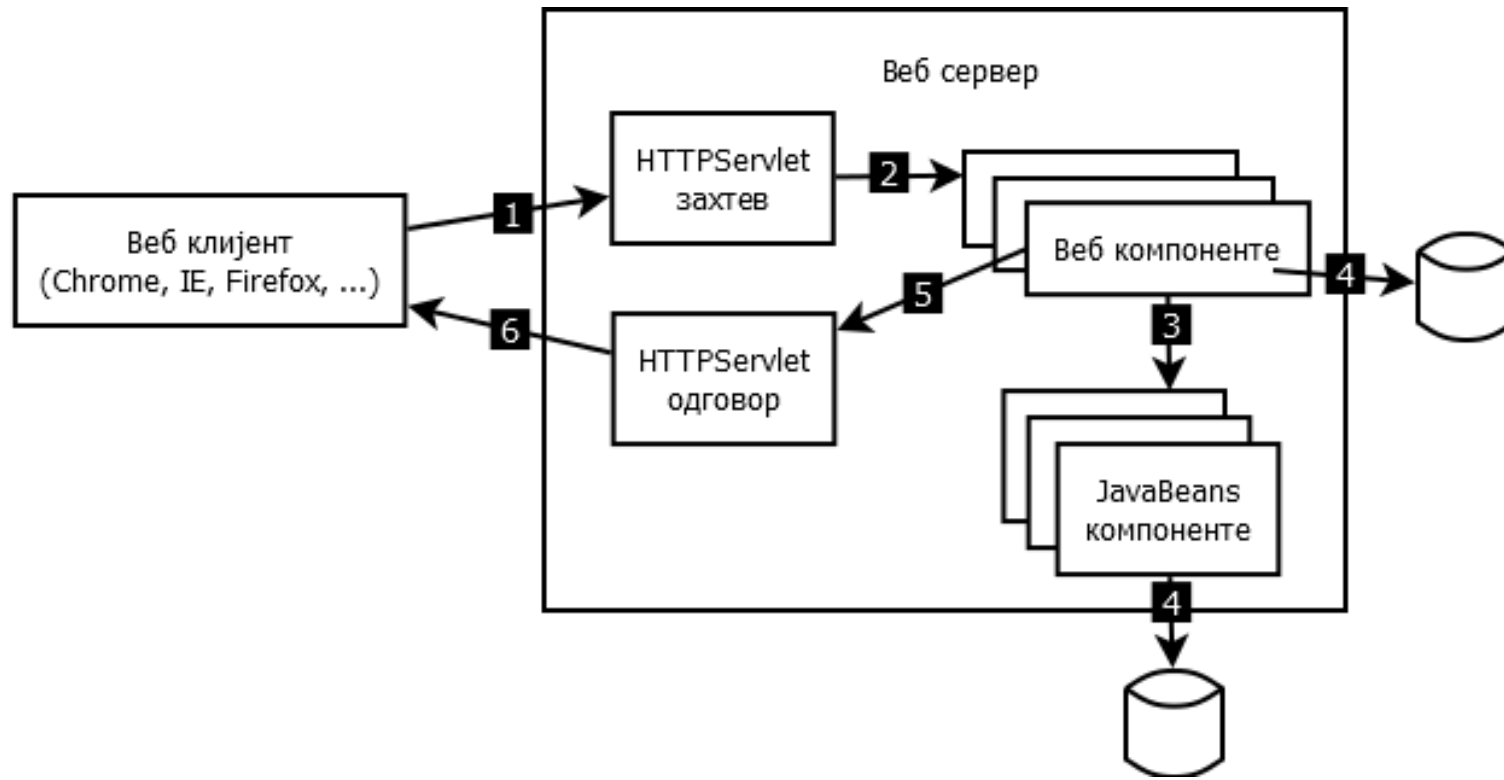
- Овде ће бити побројано неколико АРІ који се налазе ван централног:
 - Enterprise АРІ (укључује JDBC, Java IDL и Java RMI),
 - Server АРІ,
 - Security АРІ итд.
- Ови АРІ-ји су се до верзије Јава 8, налазили у оквиру пословног издања Јаве, тј. J2EE.
- Почев од верзије Јава 9, J2EE се више не испоручује у оквиру Јаве.
 - Већ је, под именом Jakarta EE, као софтвер отвореног кода.
 - Доступна од стране Eclipse Foundation организације.

JAVA ENTERPRISE API

- Коришћењем овог API развијају се сложене дистрибуиране, клијент/сервер и друге апликације у Јави.
- Најважнији делови Java Enterprise API су:
 - Java Database Connectivity, или JDBC.
 - Јава RMI омогућава удаљену комуникацију између Јава програма.
 - Библиотека Jini је напреднија верзија RMI:
 - она ради слично као и RMI, али уз побољшану сигурност, могућност проналажења удаљених објеката итд.



JAVA SERVER API



ПИТАЊА И ЗАДАЦИ

- У којој верзија Јаве (по њеним творцима) су направљене значајније промене? Које су то промене?
- Шта се подразумева под платформом система модула која је уведена у Java SE 9?
- Које су захтеве за програмски језик Јава су поставили креатори овог језика на почетку развоја?
- Који типови апликација се могу креирати програмским језиком Јава?
- Упоредити процес превођења изворног кода који је написан у програмском језику Јава у извршни код и процес превођења изворног кода написаног у програмском језику С у извршни код.
- Предности и недостаци употребе ЈИТ Јава преводиоца.
- Објаснити организацију и архитектуру Јава виртуелне машине.
- Шта је Јава API и за шта се користи?
- Шта су Јава модули и која је предност употребе система модула?
- Упоредити структуру JDK директоријума за Јава 8 и Јава 9.
- Истражити и упоредити различита издања Јава окружења.