



CODE@FEIT

ЧАС 1: ВОБЕД



ПРЕГЛЕД НА КУРСОТ

- Цел - што сакаме да постигнеме?
- Опис – која содржина ќе биде покриена?
- Ресурси – кои книги и софтвер ќе бидат користени?
- Организација – траење на курсот, главни активности, дневен распоред



ЦЕЛ НА КУРСОТ

- Вовед во општите концепти на програмирањето и дизајнирањето на алгоритми.
- Научете доволно Java за да направите нешто корисно 😊
- Размислувајте како програмери!
- Започнете да програмирате во Java!



ОПИС НА КУРСОТ

- Преглед на Јава програмскиот јазик и околината за развој
 - Запознавање со Јава
 - Основни податочни типови и променливи
 - Оператори(аритметички, релациони, логички) и операции
 - Структури за избор и повторување (if...else, while, for)
 - Функции
 - Низи
 - Алгоритми за сортирање



ЗОШТО ЈАВА?

- Моќен програмски јазик, развиен од Sun
- Објектно-ориентиран
- Развој на десктоп, мобилни и веб апликации
- Java е програмскиот јазик на интернетот
- Напиши еднаш, изврши секаде!



РЕСУРСИ

- Книги:
 - <http://docs.oracle.com/javase/tutorial/>
 - Introduction to Java Programming, Y. Daniel Liang, 8th Edition
- Софтвер
 - JDK 7 со NetBeans
 - <http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html>
 - Eclipse



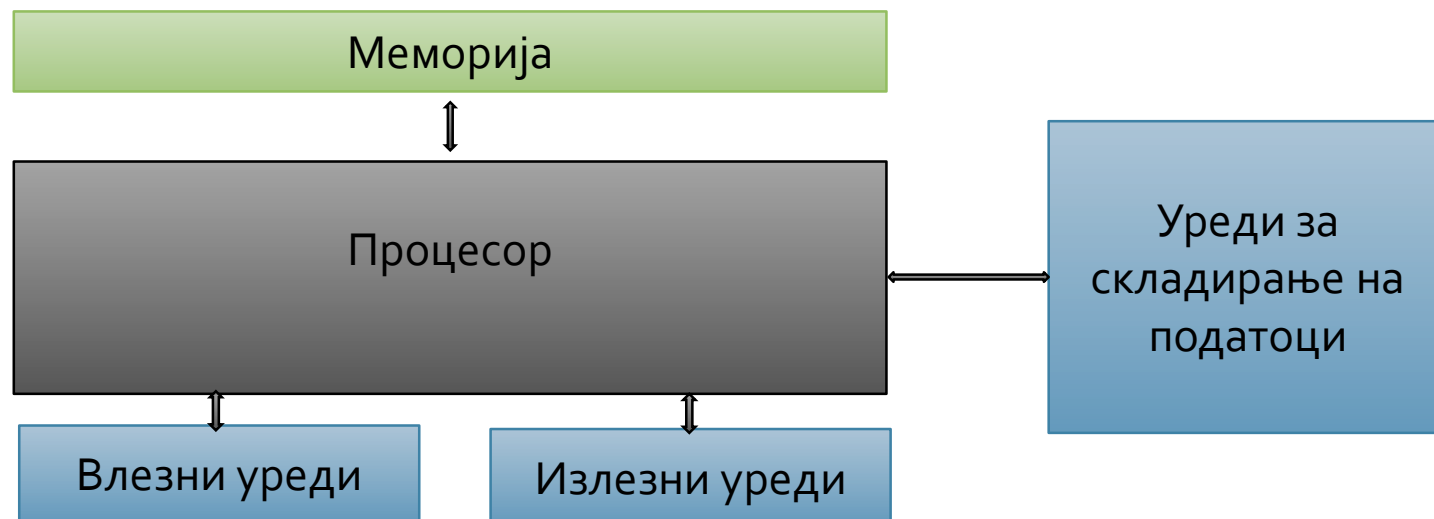
ОРГАНИЗАЦИЈА

- Трање
 - 5 дена (12.09.2016 – 16.09.2016)
- АКТИВНОСТИ
 - Предавања и вежби
- Место
 - ФЕИТ, 110
- Потребни предзнаења
 - Нема!



ШТО Е КОМПЈУТЕР?

- Електронски уред кој чува и процесира податоци
- Комбинација од хардвер и софтвер
- Организација:
 - Процесор (CPU)
 - Меморија
 - Уреди за складирање на податоци
 - Влезно - излезни уреди





МЕМОРИЈА

- Податоците во меморија се чуваат во бинарна форма (битови, 0 и 1)
- Група од 8 бита сочинува еден бајт ($8b = 1B$)
- Најчесто еден бајт е минимална мемориска единица
- Пред да почне да се извршува, програмата, заедно со податоците мора да се сместат во меморија
- Секој бајт, односно податок има своја мемориска адреса



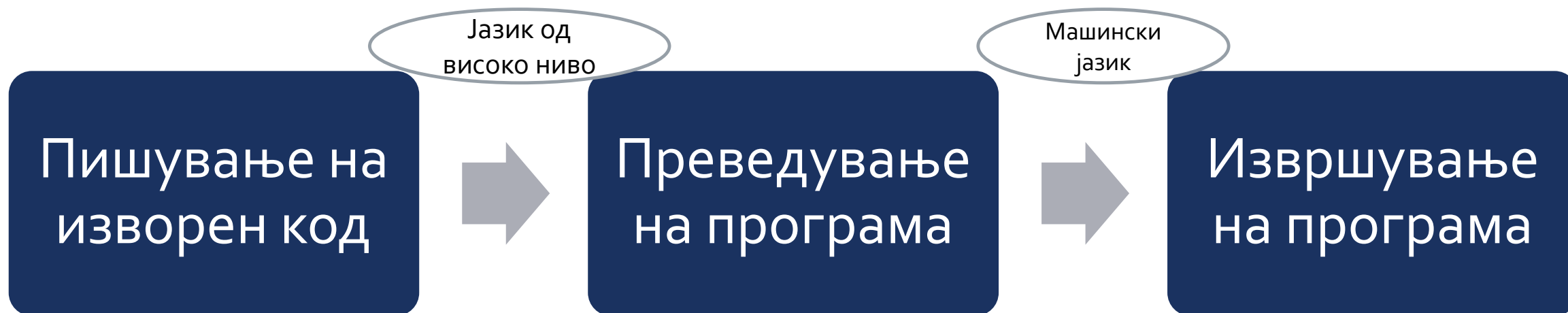


ШТО Е ПРОГРАМИРАЊЕ?

- Како ги користиме компјутерите:
 - Програмери: пишуваат програми за решавање на определени задачи
 - Крајни корисници: Користат програми напишани од програмерите
- Што е програма?
 - Множество од инструкции
- Што е програмски јазик?
 - Средство за пишување на програми
- Процесорот ги извршува програмите напишани во определен програмски јазик
- ...но, пред да ги изврши мора да ги добие во бинарна форма, наречена машински јазик



ПРОГРАМСКИ ЦИКЛУС





ПИШУВАЊЕ, КОМПАЈЛИРАЊЕ И ИНТЕРПРЕТИРАЊЕ

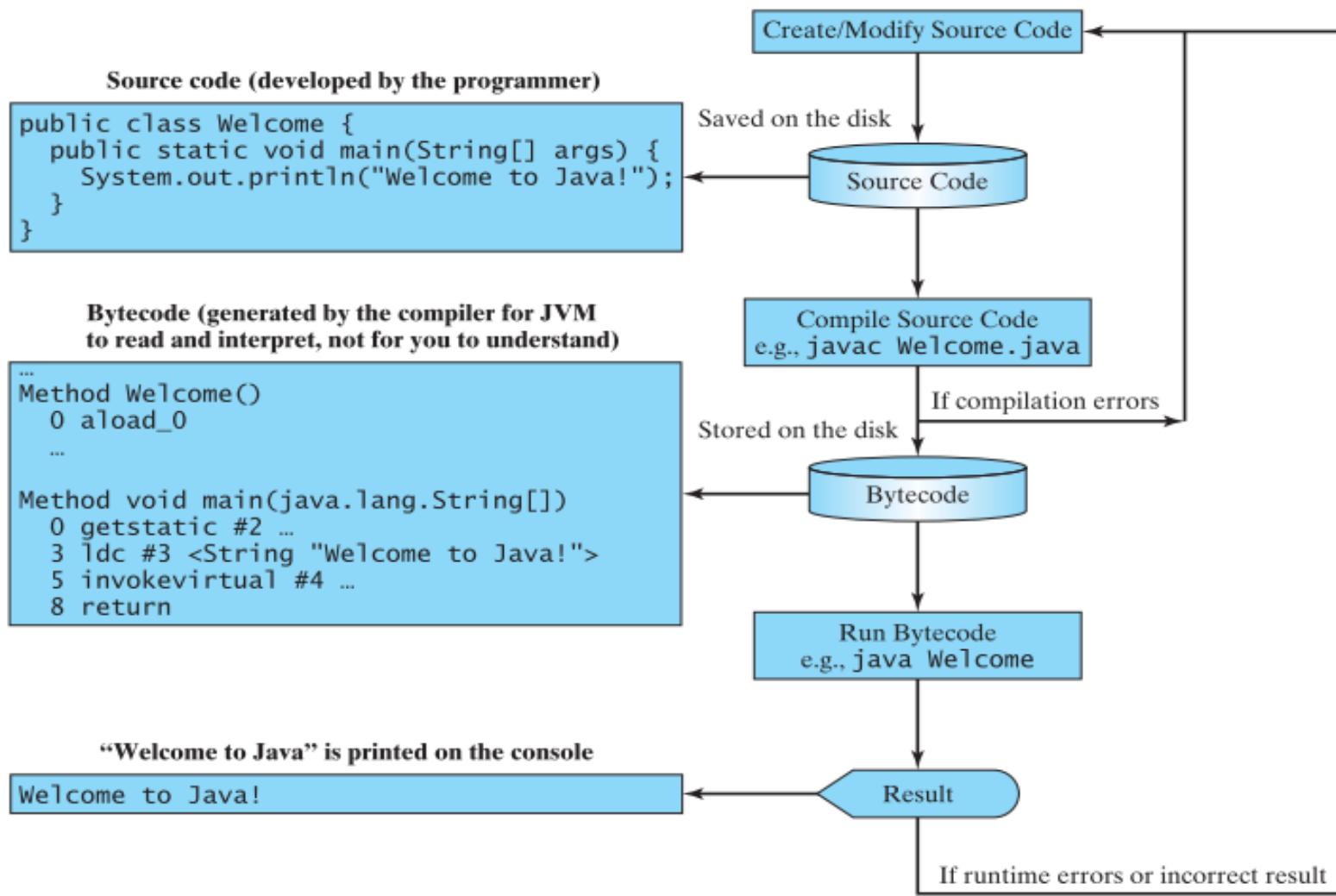
■ Во текстуален едитор:

- Изворниот код го пишуваме во текстуална датотека
- Изворниот код треба да биде именуван според правила кои ги налага јазикот
 - Програмите напишани во Java програмскиот јазик имаат **.java** наставка
- Команда за компајлирање: **javac**
 - > **javac** EdnostavenPrimer.java
- Компајлерот ги пријавува сите грешки кои би можеле да настанат при компајлирање
 - Грешките мора да се поправат и програмата треба да се искомпајлира одново
- Ако помине компајлирањето, на излез се добива Java бајткод (со **.class** наставка)
- Команда за извршување: **java**
 - > **java** EdnostavenPrimer





ПИШУВАЊЕ, КОМПАЈЛИРАЊЕ И ИНТЕРПРЕТИРАЊЕ

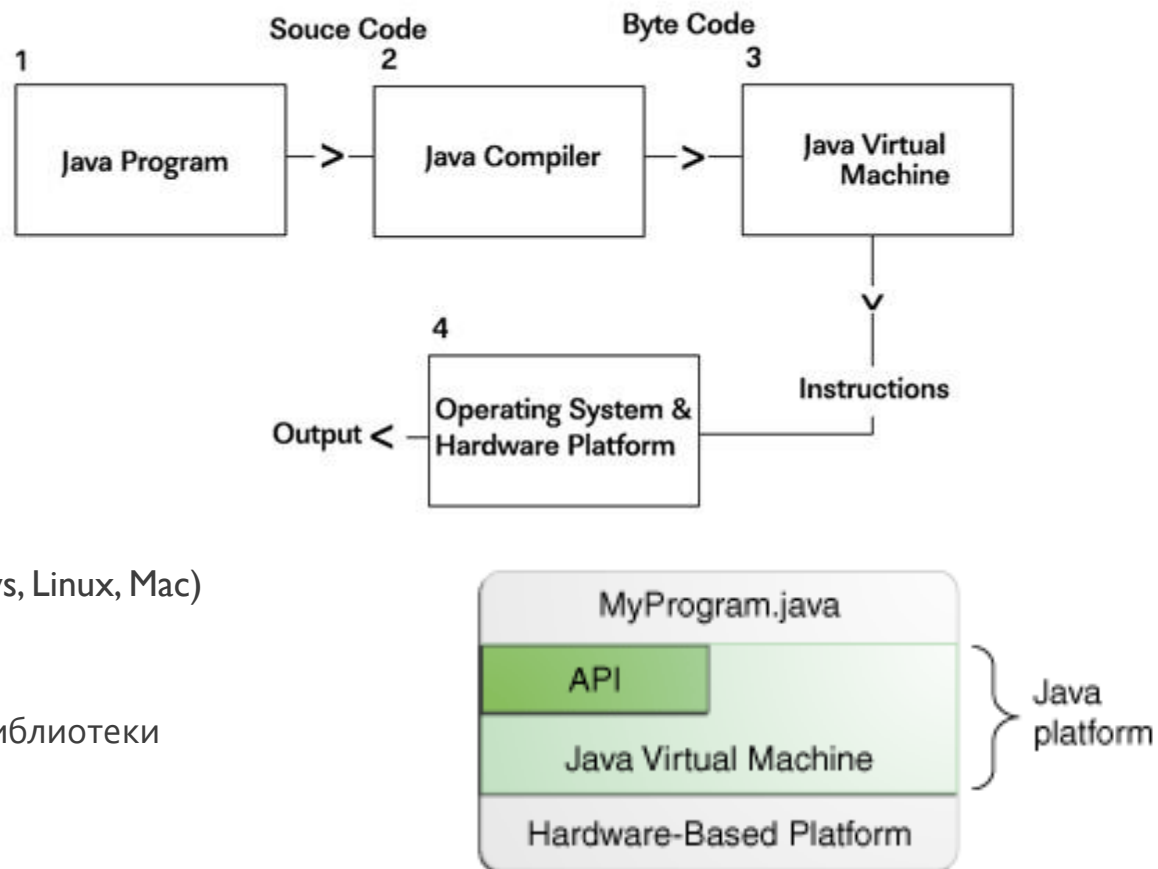




JAVA ТЕХНОЛОГИЈА

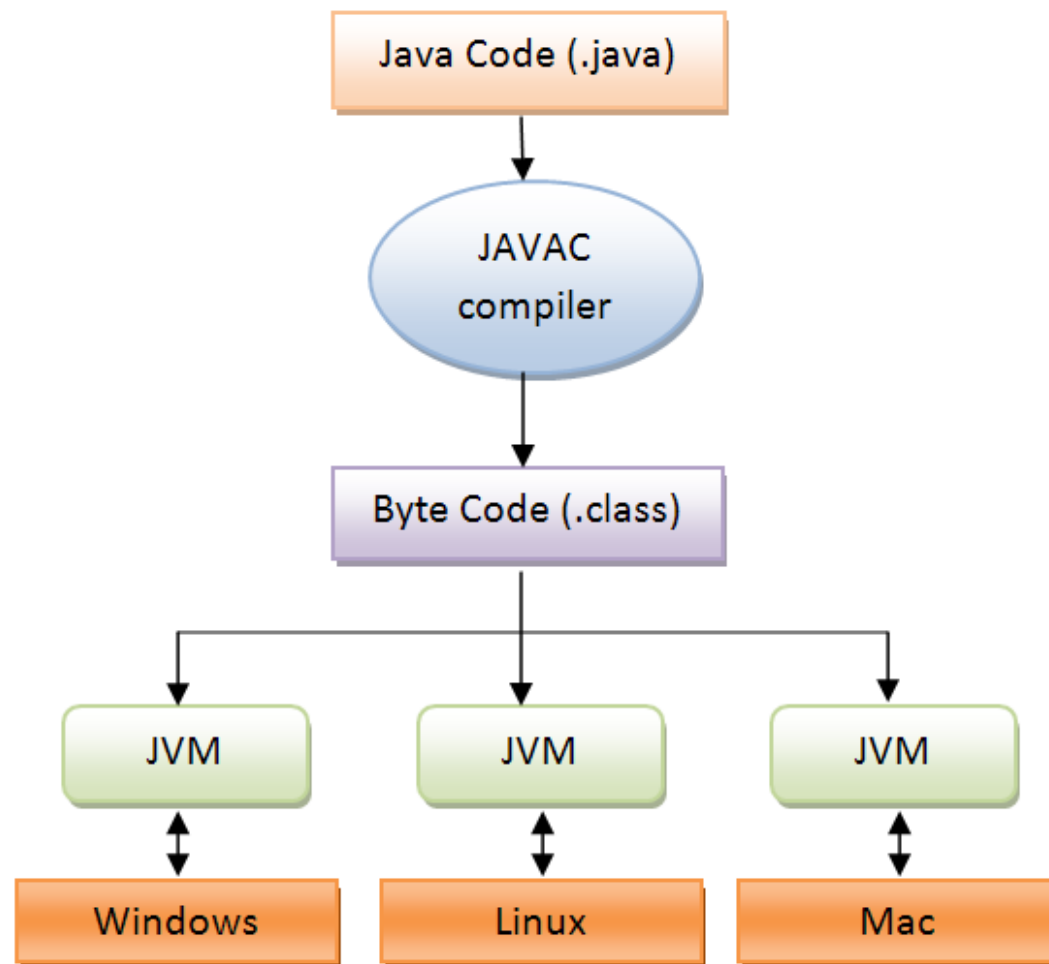
- Java програмски јазик
 - Јазик од високо ниво
 - Разбирлив за човекот
- Java бајткод
 - Јазик од ниско ниво
 - Множество од основни инструкции
 - Независен од платформа
- Java виртуелна машина (JVM)
 - Програма која интерпретира Java бајткод
 - Имплементации постојат за повеќе платформи (Windows, Linux, Mac)
- Java API
 - Збир од готови софтверски компоненти групирани во библиотеки

Java Program Execution





JAVA ПЛАТФОРМСКА НЕЗАВИСНОСТ





JAVA ПРОГРАМСКИ КОД НА НЕКОЛКУ НИВОА





ИЗВОРЕН КОД ВО JAVA (ПРИМЕР)

// Овој изворен код е именуван како EdnostavenPrimer.java

```
public class EdnostavenPrimer
```

```
{
```

// Кодот печати на екран едноставна реченица

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

```
{
```

```
    System.out.println("Prva Java programa");
```

```
}
```

```
}
```



ЕДНОСТАВНА ПРОГРАМА (ОБЈАСНУВАЊЕ 1)

- Типична Java програма изгледа вака:

```
public class EdnostavenPrimer{  
    public static void main(String[] args)  
    { // ...}  
}
```

- Java кодот е сместен во класи
- Секоја Java програма мора да има барем една public класа
- Името на класата мора да соодветствува со името на програмата
 - EdnostavenPrimer.java



ЕДНОСТАВНА ПРОГРАМА (ОБЈАСНУВАЊЕ 2)

- За да биде извршлива, програмата мора да дефинира точно една `main()` функција

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

```
{
```

```
//тука се пишуваат наредби кои ќе бидат извршени секвенцијално
```

```
}
```

- Телото на програмата се пишува во рамки на големи загради { ... }
 - Секоја логичка целина во програмата, наречена блок, исто така се пишува во рамки на големи загради
- Секоја наредба завршува со точка-запирка (;)
 - `System.out.println("Prva Java programa");`



ЕДНОСТАВНА ПРОГРАМА (ОБЈАСНУВАЊЕ 3)

```
public class EdnostavenPrimer {  
    public static void main(String[] args) {  
        System.out.println("Prva Java programa");  
    }  
}
```

Class блок

main() блок



ИЗВОРЕН КОД ВО JAVA (ПРОШИРУВАЊЕ)

```
public class EdnostavenPrimer2
{
    public static void main(String[] args)
    {
        System.out.println("Dobredojdovte na Code@FEIT");
        System.out.println("Nauchete da programirate vo Java.");
        System.out.println("Programiranjeto e zabavno! 😊");
    }
}
```



ИЗВОРЕН КОД ВО JAVA (ПРОШИРУВАЊЕ)

- Како ќе го пресметате изразот: $\frac{10.5+2*3}{45-3.5}$?

```
public class PresmetajIzraz {  
    public static void main(String[] args) {  
        System.out.println((10.5 + 2 * 3) / (45 - 3.5));  
    }  
}
```



КОМЕНТАРИ И ПОРАМНУВАЊЕ

- `//` Ова е коментар во еден ред и компајлерот го игнорира
- `/*` Ова е коментар
во повеќе
редови `*/`
- Ова технички ќе работи:

```
public class EdnostavenPrimer{ public static void main(String[] args)
{System.out.println("Programiranjeto e zabavno!"); System.out.println("Nauchete da programirate
vo Java."); System.out.println("Rabotete naporno 😊");}}
```

- Но не го практикувајте!
- Пишувајте уредно и разбирливо!
- Java конвенција: користете четири празни места за секое ново вовлекување на текстот



ТИПОВИ НА ПОДАТОЦИ

- Податочниот тип определува:
 - како вредностите за даден податок се сместуваат во меморијата и колку меморија зафаќаат (зависно од платформата),
 - множеството вредности за податокот, и
 - операциите што може да се извршат со или над неговите вредности.



ТИПОВИ НА ПОДАТОЦИ

- Цели броеви



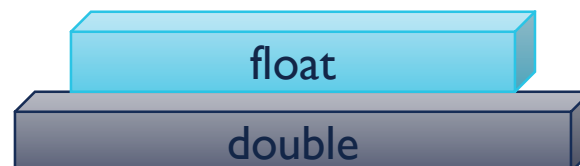
-1, 0, 134, -765432

- Знаци



'a', 'C', '\n', 'f'

- Реални броеви



1.23, 1.23e3, -23.3546

- Стрингови



"niza", "feit"



ПРОМЕНЛИВИ

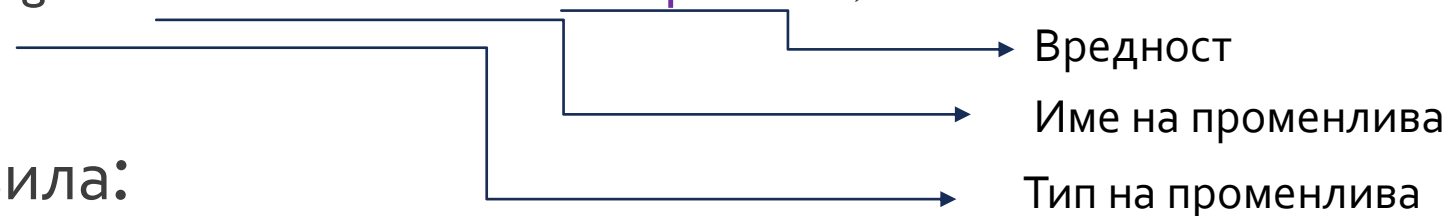
- Именувани мемориски локации
- Чуваат вредности за време на извршување на програмата
- Секоја променлива има свој тип, име и вредност
- Декларирање на променлива:
`<tip_promenliva> <ime_promenliva> [=<vrednost_promenliva>];`
- Секогаш кога нова вредност ќе биде сместена во променливата, претходната вредност се брише
- Читањето вредност од променливата не ја менува истата



ПРОМЕНЛИВИ

■ Пример:

- `String rechenica= "Practice makes perfect!";`



■ Правила:

- Декларирањето секогаш се прави на почетокот на функцијата или
- на почеток на кој и да е блок од изрази или
- и тоа секогаш пред да биде употребена

■ Употреба:

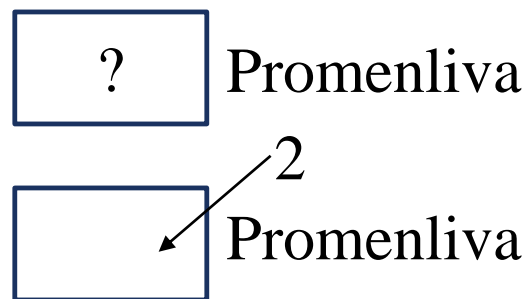
- `System.out.println(rechenica);`



ПРОМЕНЛИВИ - ЗАДАВАЊЕ ВРЕДНОСТ НА ПРОМЕНЛИВА

```
int Promenliva;
```

```
Promenliva = 2;
```



- формат

```
Promenliva = vrednost;
```



ПРОМЕНЛИВИ

- `int a, b, c; //целобројни променливи`
- `int d = 3, e, f = 5;`
- `double x = 22.2; //реална променлива`
- `char bukva = 'x'; //знаковна променлива (во единечни наводници)`
- `String niza1, niza2="Niza"; //текстуални променливи (во двојни наводници)`



ПРОМЕНЛИВИ

```
public class Hello3 {  
    public static void main(String[] arguments) {  
        String foo = "FOO";  
        System.out.println(foo);  
        foo = "BOOFOO"; /* претходната вредност на променливата се заменува  
                        со нова */  
        System.out.println(foo);  
    }  
}
```



ФУНКЦИИ ЗА ВЛЕЗ И ИЗЛЕЗ

- Функции за излез(System.out)
 - println – го поместува курсорот во нов ред
 - print – курсорот останува во истиот ред
- Функции за влез(System.in)

`Scanner input = new Scanner(System.in);`

- Методи за читање:

`nextByte()`

`nextShort()`

`nextInt()`

`nextLong()`

`nextFloat()`

`nextDouble()`

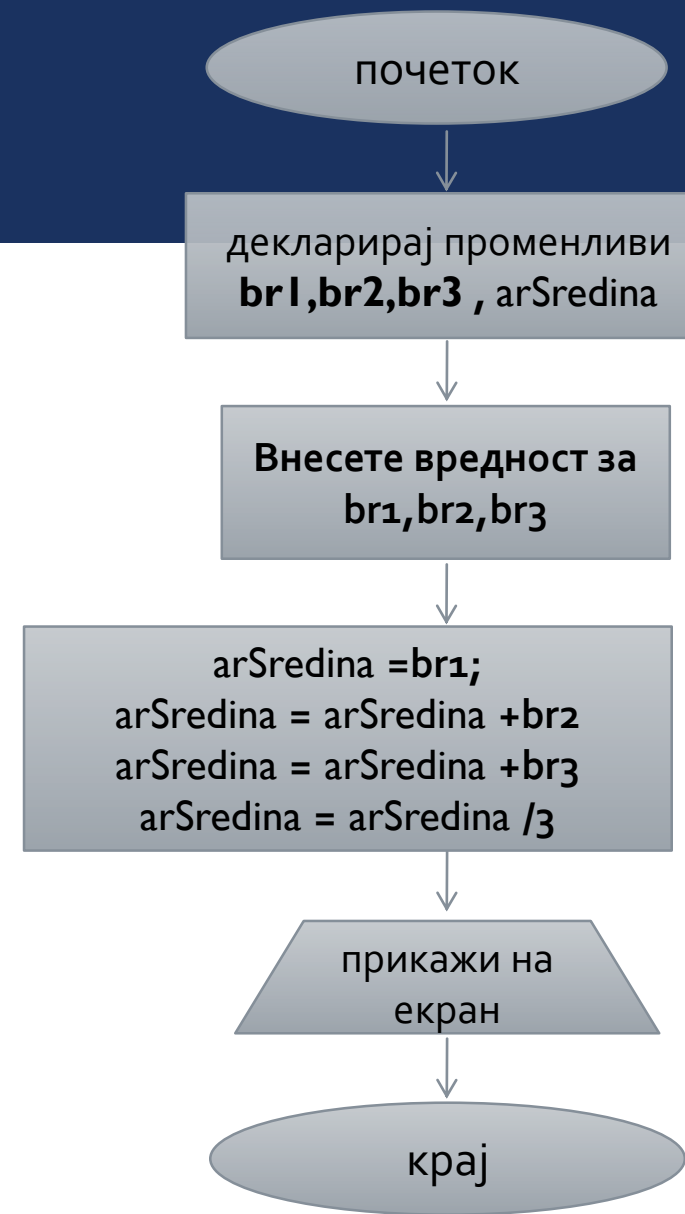
`next()` – чита стринг до првото празно место

`nextLine()` – чита цела линија стринг (до првиот ентер)



ФУНКЦИИ ЗА ВЛЕЗ И ИЗЛЕЗ

- Да се пресмета аритметичка средина од три броја внесени од тастатура





ФУНКЦИИ ЗА ВЛЕЗ И ИЗЛЕЗ

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    System.out.println("Vnesete gi broevite ");  
    double br1 = input.nextDouble();  
    double br2 = input.nextDouble() ;  
    double br3 = input.nextDouble() ;  
    double arSredina = (br1 + br2 + br3) / 3;  
  
    System.out.println("Aritmetichkata sredina na " + br1 + " " + br2  
        + " " + br3 + " e " + arSredina);  
}  
}
```



ФУНКЦИИ ЗА ВЛЕЗ И ИЗЛЕЗ

- Како ќе напишете програма која на екран го дава следниов излез:

```
public static void main(String[] args) {  
    System.out.println("  J ");  
    System.out.println("  J ");  
    System.out.println(" J  J ");  
    System.out.println(" JJ ");  
}
```

```
  J      A      V      V      A  
  J      A A      V      V      A A  
J  J      AAAAA      V V      AAAAA  
J J      A      A      V      A      A
```



ОПЕРАТОРИ

- Аритметички
- Релациони
- Логички



АРИТМЕТИЧКИ ОПЕРАТОРИ

- Доделување: =
- Собирање: +
- Одземање: -
- Множење: *
- Делење: /
- Остаток од целобројно делење: %
- Редослед на извршување на операции (стандардни математички правила):
 - Загради
 - Множење и делење
 - Собирање и одземање
- Ако еден операнд е реален број, а другиот цел број, резултатот е реален број.
- $5/2=2$
- $5\%2=1$ (важи само за цели броеви!)

$$5.0/2.0=2.5$$

$$5/2.0=2.5$$

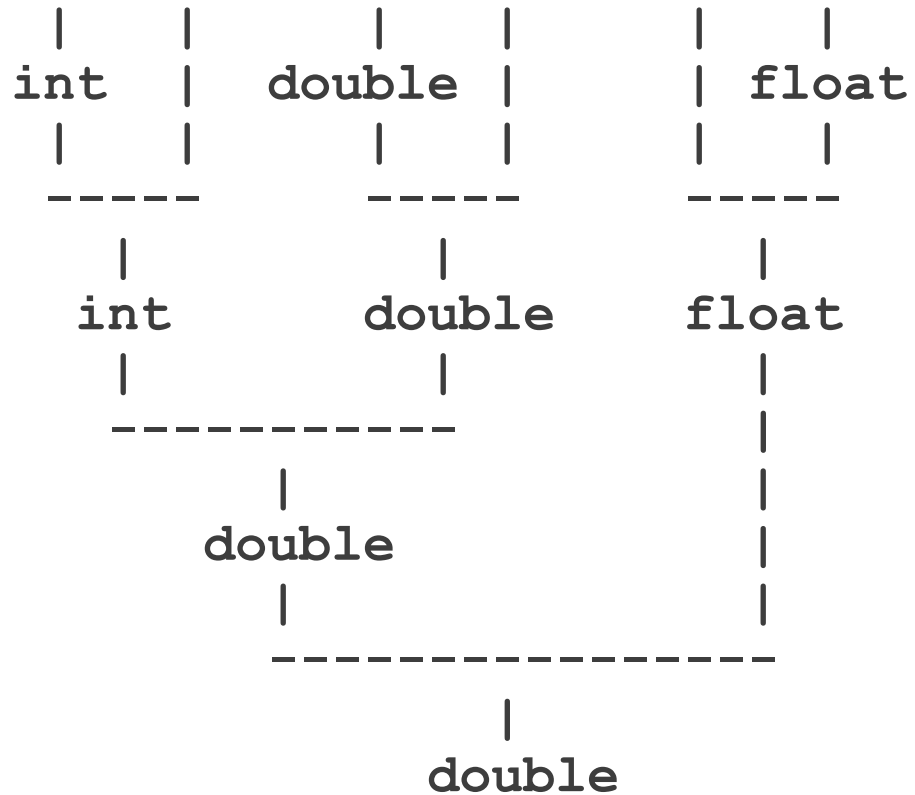
$$5.0/2=2.5$$



АРИТМЕТИЧКИ ОПЕРАТОРИ - ПРИОРИТЕТ

```
char ch; int i; float f; double d, rezultat;
```

```
rezultat = (ch / i) + (f * d) - (f + i);
```





АРИТМЕТИЧКИ ОПЕРАТОРИ

```
public class Matematika {  
    public static void main(String[] args) {  
        double zbir = 2.0+3.1+4.2;  
        System.out.println(zbir);  
        double starZbir = zbir ;  
        zbir = zbir/2;  
        System.out.println(starZbir);  
        System.out.println(zbir);  
    }  
}
```



СКРАТЕНИ АРИТМЕТИЧКИ ОПЕРАТОРИ

- $a = a + 1$
- $a += 1$
- $a = 10$
- $a += 1 // a = a + 1 // 11$
- $a \% = 3 // a = a \% 3 // 1$

$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$
$a \% = b$	$a = a \% b$



СКРАТЕНИ АРИТМЕТИЧКИ ОПЕРАТОРИ

- Кога се користат оператори за доделување, изразот се евалуира од десно кон лево

```
public static void main(String[] args)
{
    int a=5, i=7, j=9, k;
    float f=10.5;

    a += 27;
    f /= 9.2;
    k = i *= j+2;

    System.out.println(a+f+i+k);
}
```

32
1.141404
77

32 1.141404 77 77



ПРЕФИКС/ПОСТФИКС НОТАЦИЈА (++/--)

- Постфикс нотација (prom++/--) – употреби, а потоа инкрементирај/декрементирај
- Префикс нотација (++/--prom) – инкрементирај/декрементирај, а потоа употреби

```
public static void main(String[] args)
{
    int i = 1;
    System.out.println("i =" + i);
    System.out.println(" i =" + i++);
    System.out.println(" i =" + i);
    return 0;
} //main
```

i=1

i=1

i=2

```
public static void main(String[] args)
{
    int i = 1;
    System.out.println(" i = " + i);
    System.out.println(" i =" + ++i);
    System.out.println(" i =" + i);
    return 0;
} //main
```

i=1

i=2

i=2



КОЈ ЌЕ БИДЕ ИЗЛЕЗОТ НА СЛЕДНАТА ПРОГРАМА?

```
public static void main(String[] args)
{
    int i=0, j, k=7, m=5;
    j = m += 2;
    System.out.println("j = " + j); /* j = m = m+2; */

    j = k++ > 7;
    System.out.println("j = " + j); /* j = k>7, k++ */

    j = i == 0 || --k;
    System.out.println("j = " + j + " k = " + k); // j = ? k = ?
}
```



АРИТМЕТИЧКИ ОПЕРАТОРИ

- Конкатенација на низи (стрингови)

- Оператор: +

Спојува повеќе помали стрингови во еден голем стринг

```
String tekst = "Zdravo" + "svetu";
```

```
tekst = tekst + "!";
```



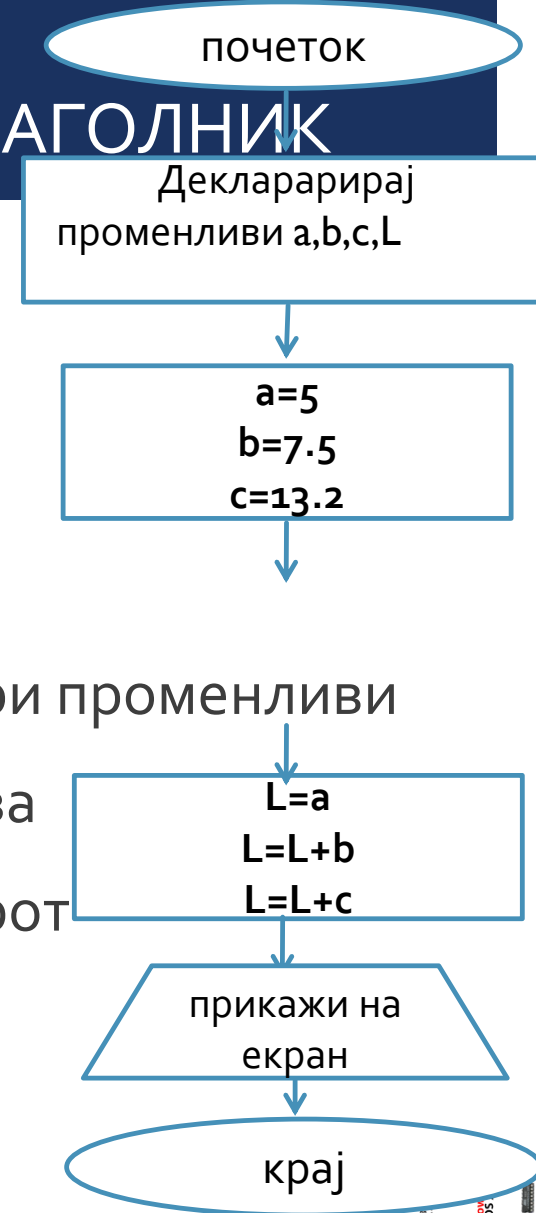
НЕСООДВЕТНОСТ НА ТИПОВИ НА ПРОМЕНЛИВИ И КАСТИРАЊЕ

- `String pet = 5; //грешка`
test.java.2: incompatible types
found: int
required: java.lang.String
- `int a = 2; //a = 2`
`double b = 2; //b = 2.0`
`int a = 15.5 //грешка`
`int a = (int) 15.5; //a = 15, кастирање`
`double b = 2/5; //b = 0.0`
`double b = (double)2/5; //b = 0.4`



ПРОГРАМА 1: ПРЕСМЕТУВАЊЕ НА ПЕРИМЕТАР НА ТРИАГОЛНИК

- Чекори:
 - Дизајнирање на алгоритам (псевдокод или дијаграм)
 - Преведување на алгоритмот во Java код
- Чекор 1. Вчитајте вредност за трите страни на триаголникот во три променливи
- Чекор 2. Соберете ги и резултатот сместете го во нова променлива
- Чекор 3. Прикажете ја на екран добиената вредност за периметарот





ПРОГРАМА 1: ПРЕСМЕТУВАЊЕ НА ПЕРИМЕТАР НА ТРИАГОЛНИК

```
public class Triagolnik
{
    public static void main(String[] args) {
        //Чекор 1. Вчитајте вредност за трите страни на триаголникот во три променливи
        //Чекор 2. Соберете ги и резултатот сместете го во нова променлива
        //Чекор 3. Прикажете ја на екран добиената вредност за периметарот
    }
}
```



ПРОГРАМА 1: ПРЕСМЕТУВАЊЕ НА ПЕРИМЕТАР НА ТРИАГОЛНИК

```
class Triagolnik
{
public static void main(String[] args) {
    /* програма која ќе пресметува периметар на триаголник
    со познати вредности за страни */
    int a=6, b=4, c=5; //декларирање и иницијализирање на три целобројни променливи
    int L; //декларирање на променлива во која ќе се смести вредноста за периметарот
    L=a+b+c; //пресметување на периметарот
    System.out.print("Perimetarot na triagolnikot so strani "+a+", "+b+", "+c); //печатење
    System.out.println(" iznesuva "+L);
}
}
```



ПРОГРАМА 2: ПРЕСМЕТУВАЊЕ НА ПЛОШТИНА НА ТРИАГОЛНИК

- Херонова формула:

- $P = \sqrt{S(S-a)(S-b)(S-c)}, S = \frac{a+b+c}{2}$



ПРОГРАМА 2: ПРЕСМЕТУВАЊЕ НА ПЛОШТИНА НА ТРИАГОЛНИК

```
public class EdnostavenPrimer {  
    public static void main(String[] args) {  
        int a ,b ,c, S;  
        double P;  
        a = 8;  
        b = 5;  
        c = 11;  
        S = (a+b+c)/2;  
        P = Math.sqrt(S*(S-a)*(S-b)*(S-c)); //се вклучува библиотеката import java.lang.*  
        System.out.print("Ploshtinata na triagolnik");  
        System.out.print(" so strani "+a+", "+b+", "+c+ " e "+P);  
    }  
}
```



ПРОГРАМА₃: ПРЕСМЕТУВАЊЕ НА ПЛОШТИНА НА КРУГ

- Задача: Да се пресмета плоштина на круг со даден радиус. Радиусот го внесува корисникот.

```
import java.util.Scanner;

public class JavaApplication2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Vnesete vrednost za radiusot: ");
        double radius = input.nextDouble() ;

        double area = radius * radius * 3.14159;
        System.out.println("Ploshtinata na krugot so radius " +
            radius + " e " + area);
    }
}
```



ПРОГРАМА 4: ЗАМЕНА НА ВРЕДНОСТ НА ДВЕ ПРОМЕНЛИВИ

```
import java.util.Scanner;
public class JavaApplication2 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int a, b, pom;
        System.out.print("Vnesete vrednost za dve promenlivi: ");
        a= input.nextInt() ;
        b= input.nextInt() ;
        pom=a;
        a=b;
        b=pom;
        System.out.println("Novite vrednosti se: " + a+ " i " + b);
    }
}
```