гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Компютърно образователен център
Progress
Палитра от възможности!

Лекция 2.2

Създаване на Java Console приложения.Представяне на данни в паметта.

- **Преговор на предишната лекция**
- **Писане на прости приложения на Java.**
- **Използване на команди за входно изходни операции.**
- **Прости типове данни в Java.**
- **Представяне на данни в паметта**
- **Задачи**

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Преговор

- Как се пишат коментари ?

- Избройте няколко ключови думи

- Как се дефинира class

- Дефиниране на main() метод

- Изпишете един не правилен идентификатори в Java

- Кои метод може да се използва за показване на текст на Стандартен изход

- Избройте няколко правила за добър стил на програмиране

Променяме примерната програма да Fig.2.1 да използва различен начин за извеждане на текст със следните команди команди :

print()

println()

```
1  // Fig. 2.3: Welcome2.java
2  // Printing a line of text with multiple statements.
3
4  public class Welcome2
5  {
6     // main method begins execution of Java application
7     public static void main( String args[] )
8     {
9        System.out.print( "Welcome to " );
10       System.out.println( "Java Programming!" );
11
12    } // end method main
13
14 } // end class Welcome2
```

```
Welcome to Java Programming!
```

System.out.print keeps the cursor on the same line, so System.out.println continues on the same line.

Outline

Welcome2.java

1. Comments

2. Blank line

3. Begin class Welcome2

4.1 Method System.out.println

5. end main, Welcome2

Program Output

Промени :

-Welcome2.java извежда същия текст като Welcome1.java

- Ред номер 9 извежда “Welcome to”, като курсора остава на същия ред.

-Ред номер 10 извежда “Java Programming” на същия ред, като курсора се премества на следващия ред.

Escape sequances :

- Backslash (\) – escape char

- Интерпретира извеждане на специални символи,има специално значение за компилатора

- Символ за нов ред (\n)
  - Интерпретира се като специални знаци от методите.
  - Премества курсора на следващия ред
  - Welcome3.java(Fig 2.4)
  -Курсорът се пренася на следващия ред при извеждане на \n

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Welcome3.java с множество от нови редове:

```
1 // Fig. 2.4: Welcome3.java
2 // Printing multiple lines of text with a single statement.
3
4 public class Welcome3
5 {
6    // main method begins execution of Java application
7    public static void main( String args[] )
8    {
9       System.out.println( "Welcome\nto\nJava\nProgramming!" );
10
11   } // end method main
12
13 } // end class Welcome3
```

Outline

Welcome3.java

1. main

2. System.out.println (uses \n for new line)

Program Output

```
Welcome
to
Java
Programming!
```

Notice how a new line is output for each \n escape sequence.

Специални символи в java :

Escape sequence	Description
\n	Newline. Курсорът се премества на следващия ред.
\t	Horizontal tab. Премества курсора на следващият TAB.
\r	Carriage return. Връща курсора в началото на текущия ред , не се премества на следващия ред.
\\	Backslash. Извежда символа \ без да използва специалното му значение .

\"	Double quote. Извежда ". For example,
	`System.out.println( "\"in quotes\"" );`
	отпечатва
	`"in quotes"`

\' - Добавя единичен знак ' в даден текст.

\u(код на символа) – добавя символ с даден код от ASCII таблицата

\u0042 – **0042 номер за буквата "A"**

Например,ако искате да изведете кавички(" "),трябва да се използва Double quote.

За да принтираме : She said **"Hello!"** to me.

Текстът подаден на функцията println(), трябва да изглежда така :

```
System.out.println("She said \"Hello!\" to me.");
```

- **System.out.printf**

  -Нововъведение от JSE 5.0

  - Форматира изходния текст – пример

```
9         System.out.printf( "%s\n%s\n",
10            "Welcome to", "Java Programming!" );
```

Синтаксис на **System.out.printf**

- printf(formatString, argList)

Методът може да бъде използван за :

- Фиксиран текст

-форматни спецификатори – placeholder(място за вмъкване на стойност от списъка с параметри)

-Пример **%s – placeholder** за низ от символи(String)

- **argList** включва списък от идентификатори,които се съпоставят на форматните спецификатори.

Welcome4 принтира множество редове,чрез printf().

```
1  // Fig. 2.6: Welcome4.java
2  // Printing multiple lines in a dialog box.
3
4  public class Welcome4
5  {
6     // main method begins execution of Java application
7     public static void main( String args[] )
8     {
9        System.out.printf( "%s\n%s\n",
10          "Welcome to", "Java Programming!" );
11
12    } // end method main
13
14 } // end class Welcome4
```

System.out.printf displays formatted data.

```
Welcome to
Java Programming!
```

Outline

Welcome4.java

main

printf

Program output

Правило за добро програмиране :

Оставяйте по един празен символ след запетая(,) в списъка от аргументи за по-добра четаемост на програмния код.

Обичайна грешка при програмиране:

Пропускането на затварящите “ кавички в даден текст,води до **синтактична грешка** и трябва да внимаваме и да прилагаме правилото като при отваряща и затваряща скоба { }

Пример :

System.out.println(“Programming with Java); - ГРЕШНО

System.out.println(Programming with Java); - ГРЕШНО

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Програма за събиране на цели числа

-Използва **class Scanner** за въвеждане на числа от **Стандартен вход**

-Използва **printf()**,за да изведе сумата на тези числа на **Стандартен изход**

-Демонстрира използването на библиотека на Java – **package**

```
1  // Fig. 2.7: Addition.java
2  // Addition program that displays the sum of two numbers.
3  import java.util.Scanner; // program uses class Scanner
4
5  public class Addition
6  {
7     // main method begins execution of Java application
8     public static void main( String args[] )
9     {
10       // create Scanner to obtain input from command window
11       Scanner input = new Scanner( System.in );
12
13       int number1; // first number to add
14       int number2; // second number to add
15       int sum; // sum of number1 and number2
16
17       System.out.print( "Enter first integer: " ); // prompt
18       number1 = input.nextInt(); // read first number from user
19
```

Outline

`import` декларация за импорт на class `Scanner` от библиотека `java.util`.

Декларира и инициализира променлива `input`, която е `Scanner` обект.

Декларира променливи `number1`, `number2` и `sum`.

Прочита цяло число от Стандартен вход и го присвоява на `number1`.

`Addition.java`

(1 of 2)

```
20       System.out.print( "Enter second integer: " ); // prompt
21       number2 = input.nextInt(); // read second number from user
22
23       sum = number1 + number2; // add numbers
24
25       System.out.printf( "Sum is %d\n", sum ); //
26
27    } // end method main
28
29 } // end class Addition
```

```
Enter first integer: 45
Enter second integer: 72
Sum is 117
```

Outline

Прочита цяло число от Стандартен вход и го присвоява на `number2`.

Пресмята сумата на променливите `number1` и `number2`, присвоява резултата на `sum`.

Извежда sum с форматиране.

Двете цели числа въведени от потребителя.

`Addition.java`

(2 of 2)

4. Addition

5. `printf`

Import декларации(3-ти ред)

- Използват се от компилатора, за да разпознае и намери дефинициите за класове в Java програмите
- Указва на компилатора да прочете дефиницията на class **Scanner** от **java.util** package

  Забележка :
  Библиотеката,package java.lang се импортира във всяко приложение по подразбиране – не трябва да се прави импорт на тази библиотека.

Дефиниране на класа

- 5-6 ред са началото на public class **Addition** →името на файла с този клас задължително трябва да се казва **Addition.java**

-Редове 8-9 : началото на **main()**

Обичайна грешка при програмиране:

Всички import декларации трябва да се пишат в група(ред под ред) **ПРЕДИ** декларацията на първия клас във файла.

Поставянето на import декларация във вътрешността на клас или след тялото на класа е синтактична грешка.

-**Променливи**

- Място в паметта, съхраняващо стойност от даден тип
  -Името и типа на променливата се декларират, преди да се използват
- input е от тип Scannner
  -Позволява да се четат данни от Стандартен вход
- Име на променлива може да е всеки идентификатор(започва с малка буква и всяка следваща дума,започва с Главна буква)

-Всяка декларация завършва с **;**

-Променлива може да се инициализира при декларирането й

- Присвояване – използва знак за равенство
- Обект за достъп до Стандартен вход(in е клас обект на class System)
- System.in -Поддържа методи за четене от Стандартен вход

13,14,15 ред – декларация на number1,number2 и sum от тип int.

- **int** означава integer стойности(цели числа) – 5,-3,0,97
- типове **float** и **double** означават числа с плаваща запетая – 5.5,3.1,-4.1,-88,5
- **int**,**float**,**double** и **char** са прости данни

Една декларация позволява да се декларират повече от една променлива

-Използва се списък от променливи разделени със запетая :

```
int number1, // first number to add
    number2, // second number to add
    sum    ; // second number to add
```

-Всяка декларация завършва с ;

Правила за добро програмиране :

- Декларирайте всяка променлива на отделен ред.

- Изборът на смислени имена придава на програмата само – описателен характер(т.е. програмата може да се разбере като се чете кода й, вместо да се използва специално ръководство).
- По подразбиране(и задължително по време на този курс),имената на променливите започват с малка буква, и всяка дума след първата започва с главна буква.

```
17          System.out.print( "Enter first integer: " ); // prompt
```

-Съобщение – текст подканящ потребителя да въведе число

```
18          number1 = input.nextInt(); // read first number from user
```

- Резултатът от метода **nextInt**() се присвоява на number1 с помоща на оператора **=**

- Инструкция за присвояване
- Изразът от дясно се пресмята и присвоява на променливата в ляво
- Чете се като : number1 приема стойността на input.nextInt()

Правило за добро програмиране :

Оставяйте празен символ от двете страни на оператора за присвояване, за да придадете читаем вид на програмата си.

```
System.out.print( "Enter second integer: " ); // prompt
```

-Аналогично на предния ред

```
number2 = input.nextInt(); // read second number from user
```

-Аналогично на предния ред

```
sum = number1 + number2; // add numbers
```

-**Инструкция за присвояване**

- Пресмята сумата на number1 и number2(от дясно на ляво)
- Използва оператор за присвояване = да присвои резултата от събирането на променливата **sum**
- Чете се като : **sum** получава стойността на number1 + number2

```
System.out.printf( "Sum is %d \n: " , sum ); // display sum
```

- Използва **System.out.printf** да изведе резултата

-Използва форматиращ спецификатор **%d**

-Указва място за вмъкване на **int** стойност от списъка с аргументи на **printf**

Пресмятанията могат да се извършват в списъка с аргументи на printf:

```
System.out.printf( "Sum is %d\n: " , ( number1 + number2 ) );
```

Скобите около number1 + number2 не са задължителни – използват се за по-голяма яснота на кода

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Променливи

Променливи

-Всяка променлива има

- **ИМЕ**
- **ТИП**
- **РАЗМЕРНОСТ**
- **СТОЙНОСТ**

-Името символизира адерса в паметта заеман от променливата

-Когато **присвояваме**/записваме/ нова стойност на променливата,тя **замества**/изтрива/ предишната стойност на променливата.

-Четене на променливи от паметта не променя адреса или стойността им

Деклариране на променлива - задава : име,тип,размерност и евентуално стойност.

-При изпълнение на програмата, всяка декларация на променлива от прост тип води до заделяне на област в паметта в съответсвие с **типа** и **размерността** на променливата.

Прости типове променливи:

byte, **short**, **int** , **long**, **float**, **double**, **boolean** или **char**.

Таблица с прости типове данни:

TABLE 1.2 Primitive Data Types

Name	Size in Bits	Range of Values
`boolean`	1	`true` or `false`
`byte`	8	–128 to +127
`char`	16	`'\u0000'` to `'\uFFFF'`
`short`	16	–32,768 to +32,767
`int`	32	–2,147,483,648 to +2,147,483,647
`long`	64	–9,223,372,036,854,775,808 to +9,223,372,036,854,775,807
`float`	32	Approximately 10^{-38} to 10^{+38}; 7 significant digits
`double`	64	Approximately 10^{-308} to 10^{+308}; 15 significant digits

Примери за декларации на променливи от прост тип:

```
{
    int numPeople = 2;      //заделя 2 байта
    double bill   = 32.45; //заделя 4 байта
    double tip    = bill * 0.2;
    double total  = bill + tip;
    double totalPerPerson = total / numPeople ;

    System.out.println(numPeople);
    System.out.println(bill);
    System.out.println(tip);
    System.out.println(total);
    System.out.println(totalPerPerson);
}
```

Типове данни

-Примитивни типове данни

-Променливи от всеки друг тип се наричат референтен тип данни.Това са променливи, които взимат стойност на обект от клас

Референтни типове данни се характеризират с:

- Име – идентификатор на Java
- ТИП – име на клас
-Размерност – размерността на обекта от съответния клас
-Стойност – обект от клас

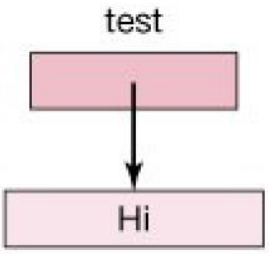
Стойността по подразбиране на променлива от референтен тип данни е **null**

Пример :

```
Scanner input;
// декларация на променлива от референтен тип Scanner
input = new Scanner(System.in);// задаване на стойност
String test;
test = "Hi";
test = new String("Bye");
```

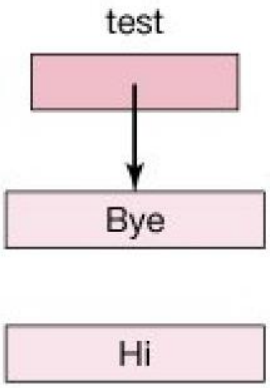
String test;

Reserves space and sets the value to null. Associates "test" with this space.



test = "Hi";

Creates a String object and sets the characters in that object to "Hi". Changes the value in the object reference "test" to refer to the String object.



test = new String("Bye");

Creates a String object and sets the characters in that object to "Bye". Changes the value in the object reference "test" to refer to the String object. The String object with "Hi" in it still exists but can be garbage collected since there is no reference to it

Точност на реалните типове

Разгледахме два реални типа – float и double. Освен с броя на възможните стойности, които могат да заемат, се различават и с точността им (броя десетични цифри, които запазват). Първият тип има точност от 6 до 9 знака, а вторият – от 15 до 17 знака.

Пример:

```
public class ExampleClass {

	public static void main(String[] args){

		// Декларация на 2 променливи от тип float и double
		float floatPI = 3.141592653589793238f;
		double doublePI = 3.141592653589793238;

		// Изпечатване на екрана 2-те променливи
		System.out.println(floatPI);
		System.out.println(doublePI);

	}
}
```

Резултат:

```
3.1415927
3.141592653589793
```

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Задачи :

Декларирайте следните променливи използвайки най-подходящия тип за променливата (**byte, short , int или long**) :

-115, 30 212, 98, - 10 000, 489 312 421

Кои от следните стойности могат да се присвоят на променлива от тип **long** и кои на променлива от тип **double**: **9.192934941 , 2.412 , 29.19293, 9491.2939192** ?

Запазете стойността “**C:\windows\system32\notepad.exe**” в променлива String и я изпечатайте с метода **println()** и **printf()**.

1.Напишете приложение, което извежда числата от 1 до 4 на един ред,разделени със запетая и шпация след нея.Използвайте следните техники:

а) една **System.out.println** команда

б) четири **System.out.print** команди

в) една **System.out.printf** команда

2. Напишете Java приложение, което да изпечатва вашето име, но използвайки само \u(код).

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

3.Напишете Java приложение,което чете цяло число, дели го по модул от 2 и извежда на стандартния изход съобщение на получения резултат.

3.Напишете приложение, което имплементира следния алгоритъм :

1. Прочетете цяло число **x** от клавиатурата

2.Отпечатайте **x**

3.Разделете **x** на две

4.Отпечатайте **x**

5.Умножете **x** по 3 и прибавете 1

6.Отпечатайте **x**