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 2.1

Въведение в ООП,Принципи на ОО дизайн.Първа програма на Java.

Основни теми :

- **Въведение**
- **Въведение в обектно ориентираното програмиране.**
- **Представяне на обекти и класове.**
- **Атрибути и стойности**
- **Действия и съобщения**
- **Въведение в синтаксиса на Java**
- **Въпроси и задачи**

Java е обектно ориентиран език

-Програмата е съвкупност от обекти, които си взаимодействат

-Telephones, houses, traffic lights, TV

-Компютърната програма като съвкупност от взаимодействащи си обекти

Видове обекти :

-Активни(извършват действия самостоятелно)

-Пасивни(нямат собствено поведение)

Активни обекти футболен играч,притежател на банкова сметка,колоездач

Пасивни обекти – футболна топка(когато е в покой),кола,банкова сметка,модел на кухня

Групи от обекти с общи характеристики

-Атрибути (текущ статус) size, shape, colour and weight

-Поведени ball rolls,bounces,a baby cries,sleeps,walks and blinks

- **Обект – модел на реални неща или събития**

Изучаваме обектите като изследваме какво характеризира текущия им статус(атрибутите) и поведение(действията,които могат да извършват)

-ATM атрибути :

- location,cash,dateOfLastCashReload

-Car атрибути:

- color,model,power,price

Всеки атрибут се представя от определен **тип данни**

Пример за два обекта от тип ATM с различни характеристики:

ATM в ж.к.Студентски град – location=”studentski grad”;cash=7 100..

ATM пред НДК – location-”Serdica mall”;cash = 38 000..

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

office@progressbg.net www.progressbg.net

Действия и съобщения

Програмите имат активен характер-изпълняват действия,реализират поведение чрез изпъление на серия от инструкци.

Примерни действия на обект футболист :
-kickTheBall()-рита топката
-passTheBall()- подава топката

Примерни действия на ATM
-report() – връща текущият баланс

- Действията винаги са свързани с определен обект,който ги реализира

- Съобщение – предаване на информация или данни от един обект на друг

-ATM позволява теглене на пари от притежател на кредитна/дебитна/ карта
-windraw(500) – аргумент(500) указва желаната сума за теглене

Аргумент – определя съдържанието на съобщението по определен начин

-windraw(500,true) –сума за теглене (500)и (true)издаване на бележка с остатъка по сметката.При (500,false) – сума за теглене 500, без да се издава бележка с остатъка по сметката.

- За да се отговори на съобщение,обектът до когото е съобщението трябва да знае как да изпълни желаното действие.

  -АТМ трябва да знае как да пресметне баланса на потребителя,също да валидира желаната операция
  -Футболистът трябва да знае как да ритне топката и как да я подаде

- Обект може да отговаря на съобщения, които са в съответсвие с неговите характеристики,статус и поведение

Видове съобщения :

- Променят статуса на обекта до когото е съобщението – mutator

moveTo()
layOnTheFeet()

- НЕ променят статуса на обекта до когото е съобщението – accesor

reportBalance()
currentColour()

Всеки клас се характеризира с :
- Полета на данните
- Множество от методи за операции с данните и предлагащи услуги на клиент класовете-тези които ще изпращат съобщения

- ✓ Програмите използват съществуващи класове като градивни елементи за конструиране на нови класове
- ✓ Всеки клас е като матрица за създаване на обекти :
  - Еднакво описание на техния статус
  - Еднакво поведение – набор от методи за реализирането му

- Всяка програма може да използва множество обекти от еднакъв тип.
  - **Примери**

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

office@progressbg.net www.progressbg.net

Класове в ООП

- Класът е програмата, записана на файл
- Всяко стартиране на програмата създава обект от този клас,който се изпълнява като процес в оперативната памет на компютъра
- Поведението на програмата, характеризиращо нейното поведение, се описва на Java от метода **main()**

Пример – програма за рисуване на правоъгълник:

Rectangle (length, width)

Rectangle – class name (кръщаваме класовете смислово обособени)

Атрибути:

length : int

width : int

Методи:

calculateArea() : int

draw()

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

office@progressbg.net www.progressbg.net

Променливи и методи в ООП

- Действия и атрибути характеризират даден обект
- В термините на програмирането това се свежда до променливи и методи
- Променлива – представлява атрибут на дадения обект
  - Променлива : Именувано място в паметта за съхраняване на определен тип стойност
- Метод – изпълнява действие
  - Набор от програмни инструкции, които реализират определено действие в съответствие със статуса на обекта

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

office@progressbg.net www.progressbg.net

Променливи и методи на обект и клас

Променливи и методи могат да са определящи само за даден обект или за всички обекти от даден клас

Променлива или метод, принадлежат на даден обект – инстанция на променлива или инстанция на метод

Променлива или метод,общи за всички обекти от даден клас – клас променлива или клас метод

В Java клас променлива или клас метод се означава с ключовата дума **static**

Пример :

class Rectangle може да има

- променлива color,която да е обща за всички обекти от тип Rectangle

- метод - main() общ за всички обекти на клас Rectangle и да служи например, за създаване на нови обекти на class Rectangle

-променливите width и length могат да бъдат различни за всяка инстанция на класа Rectangle

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!

Първа програма на Java

- Java приложно програмиране,кратка илюстрация на :
  - Използване на структурата на Java класове
  - Първоначално запознаване със Java синтаксиса
  - Логиката в изпълнение на Java програмата

  Java приложение:

  - Необходим е клас(**активен**), който да стартира изпълнението на програмата

  - При изпълнението неговия **main()** метод, се създава обект от този клас, който от своя страна създава останалите обекти от класовете(пасивни),участващи в приложението

  - Изпълнението на **програмата започва със зареждането на обект от програмния клас** в паметта и интерпретацията му от JVM, чрез изпъление на командата java от командния ред

  Пример :

  - Отпечатване на ред от текст

  Welcome to java programming!

```java
// Fig. 2.1: Welcome1.java
// Text-printing program.

public class Welcome1
{
   // main method begins execution of Java application
   public static void main( String args[] )
   {
      System.out.println( "Welcome to Java Programming!" );

   } // end method main

} // end class Welcome1
```

```
Welcome to Java Programming!
```

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

office@progressbg.net www.progressbg.net

Коментари в Java:

- Игнорират се при изпълнението на програмата
- Документират и описват кода
- Допринасят за читаемост на кода
- Изискване на професионално програмиране

Видове:

1)Коментари в края на реда – започват с : **//**

```
1       //Fig. 2.1 Welcome1.java
```

*Заб.:номерацията на редовте на програмата не са част от програмния код.

2)Многоредови коментари : /* ..... */

```
/* This is traditional comment.
   It can be split over many lines */
```

Обичайна грешка при програмиране :

Забравянето на някой от **ограничителите** на коментарите е синтактична грешка.Синтаксисът на програмния език определя правилата за писане на програма на съответния език.**Синтактична грешка** възниква, когато компилаторът открие код, който нарушава синтаксиса на Java.В такъв случай компилаторът **не** създава съответния **.class** файл.Вместо това се показва съобщение за грешка, която подпомага намирането и отстраняването на грешката в кода.Синтактичните грешки, се наричат също грешки при компилация, защото се откриват във фазата на компилация.Не е възможно да се изпълни програмата, докато не се отстранят съответните синтактични грешки.

Правило за добро програмиране :

- Всяка програма да започва с коментар, обясняващ целта на програмата, описващ автора, датата и часа на нейното последно обновяване.

```
3
```

-Празен ред

- Прави програмата читаема
- Празни редове,шпации(празни позиции) и табулация се игнорират от компилатора

```
4   public class Welcome1
```

-Започва декларацията на програмния клас за class **Welcome1**

- Всяко приложение на Java има поне един потребителски дефиниран клас
- Ключови думи : думи, запазени за употреба само за езика на Java
  -**class** е ключова дума следвана от клас име
- Именуване на класове : Всеки клас започва с главна буква
  -SampleClassName,ListOfProperties,MailBoxChecker и тн.
  -Всяка съставна дума започва с главна буква
  -Включва само символи, позволени за идентификатори на Java

Правило за добро програмиране :

- Използвайте празни редове и празни символи за подобряване на читаемостта на програмата.

```
4   public class Welcome1
```

- **Java идентификатори**

- Серия от символи, състояща се от букви,цифри,подчертаване(_) и символ($)
- Не може да започва с цифра
- Не може да съдържа празни символи(шпации)
  Примери : Welcome1,$value,_value,button7
  - 7button е не валиден идентификатор
- Java е зависима от големи и малки букви
  -a1 и A2 са различни идентификатори

- Използваме (за момента) само **public class** (да не се спестява при писане на програмите)

```
4   public class Welcome1
```

- **Java Ключови думи**
  - Запазени от Java, имат специално предназначение
  - Не може да има идентификатори с имена на ключови думи
  - Пишат се само с малки букви
    Примери : class, public ,static,int,void
  - Спазва се определен порядък, когато участват в групи
    - public static void →правилно
    - void public static → грешно
    - public class → правилно
    - class public →грешно

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

office@progressbg.net www.progressbg.net

Ключови думи в Java(Java keywords)

Keywords in Java				
abstract	default	if	private	this
assert	do	implements	protected	throw
boolean	double	import	public	throws
break	else	instanceof	return	transient
byte	enum	int	short	try
case	extends	interface	static	void
catch	final	long	strictfp	volatile
char	finally	native	super	while
class	float	new	switch	
continue	for	package	synchronized	

Правила за добро програмиране :

- -По приетия стандарт за стил ВИНАГИ започвайте името на клас с Главна буква, а също и всяка следваща дума, участваща в името на класа – останалите букви да са малки.Java програмистите именуват класовете по такъв начин – стил на „камилата”.

  Пример : MyFirstClass, BankAccount,ChessBoard

- По приетия стандарт за стил ВИНАГИ започвайте името на методи и променливи с малка буква, а всяка следваща дума, участваща в имена на методи и променливи да започва с Главна буква – останалите да са малки.
  Пример: main(), calculateTotalAmount() , accountBalance, countItems, xCoordinate , yCoordinate

Обичайна грешка при програмиране :

Java е чувствителна по отношение на използването на големи и малки букви.Използването на погрешна последователност от големи и малки букви при реферирane към един и същ идентификатор води до синтактични грешки.

```
4   public class Welcome1
```

-Запазване на файла с Java Кода

- Името на файла трябва да е същото, както името на класа и да има .java окончание
- Welcome1.java

```
5   {
```

-Лява фигурна скоба {

- Започва тялото на код – в този случай тялото на class Welcome1

-Дясна фигурна скоба } затваря декларацията на класа(ред 13)

Обичайни грешки при програмиране :

Грешка е за public class да има име за файл, което е различно от името на този клас по отношение на ред от символи и последователност от малки и големи букви.

Грешка е файл с окончание .java да съдържа повече от един public class на Java.Пишете всеки public class на Java в отделен файл.

Грешка е да се пропусне окончанието .Java за файл, съдържащ Java програмен код.При отсъствие на окончанието, компилаторът няма да може да компилира класа.

Грешка е липсата на някоя от отваряща или затварящата скоба.

Правила за добро програмиране:

Когато пишете отваряща скоба, { , в програмата си,веднага пишете и затваряща скоба.Така избягвате излишни синтактични грешки.

Подравнявайте кода на клас декларациите между, { , и, } както е дадено в примерния код.Това допринася за читаемост и по-лесно поправяне на синтактични грешки.

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

- **Всяко Java приложение трябва да има програмен клас с main() метод дефиниран, както е показано :**

- Изпълнението на програмата започва от метода main()
  -Кръглите скоби след маин указват, че това е **метод**
  - static означава, че това е клас метод(общ за всички обекти)
  - Java приложенията съдържат един или повече методи
- Точно един клас метод може да се казва main

- Методите могат да изпълняват задания и да връщат информация

- void означава,че main методът не връща информация
- Методът main(String args[]) взима аргумент String[] args
- Използвайте винаги main(String[] args) с аргументи

Лявата(отварящата) скоба започва начало на тялото на метода main()

```
8       {
```

-Дясната скоба завършва тялото main() на метода –
} (линия 11)

Правило за добро програмиране :
Подравнявайте цялото тяло на всеки метод на едно ниво, както е показано в примерния код.

```
9         System.out.println( "Welcome to Java Programming!" );
```

- Инструкция до computer да изпълни действие
  - Отпечатва низ от символи
    -String – низ от символи в двойни кавички
  - Празните символи в низа НЕ СЕ игнорират от компилатора

- System.out
  - Обект използван за стандартен изход
  - Отпечатва в command window(MS-DOS promt)

- Използва метода **System.out.println**
  - Отпечатва низ от символи

- Това е пример за изпълнима инструкция
  - Всяка изпълнима инструкция завършва с ;
  - Инструкцията принадлежи към обекта **System.out**
  - Инструкцията и обекта й са разделени с точка

Обичайна грешка при програмиране :

Пропускането на точка и запетая в края на инструкцията е синтактична грешка.Опитайте,какво ще се получи при компилация, ако пропуснете да пишете ; в края на изпълнима инструкция

Поправка на грешки :

При грешка от компилация, получавате номера на реда със синтактична грешка.Ако този ред няма грешка, проверете на прдхождащите редове за синтактична грешка.

```
11      } // end method main
```

- Край на декларацията на метода

```
13  } // end class Welcome1
```

- Край на декларацията на класа

- Може да се добавят коментари за проследяване на затварящите скоби.

- Примери за подравняване и коментари.

Правило за добро програмиране :

Използването на коментари при (}) в края на тяло на клас или метод допринася за по-добро четене и избягване на грешки в програмата.

Компилиране(compiling) на програмата

-Отворете Command prompt window, идете в директорията на Welcome1.java

-Изпълнете javac Welcome1.java

- Ако няма синтактични грешки се създава Welcome1.class

- съдържа bytecode –а който позволява изпълнението на програмата
- Bytecode-а се интерпретира от JVM

Поправка на грешки :

Ако при компилация се получи съобщение от рода на „bad command or filename”,”javac: command not found” или “’javac’ is not recognized as an internal or external command,operable program or batch file”,то Java не е инсталирана правилно.

По специално PATH environment променлива не е зададена правилно.

Ако компилаторът даде съобщение за грешка „public class ClassName must be defined in a file called ClassName.java” то името на файла не съвпада с името на public class в този файл или файл името е неправилно написано при стартиране на компилатора.

- Изпълнение на програмата
  -Изпълнете от командния ред java Welcome1 (**.class** окончанието може да се пропусне)
  - Стартира JVM
  - JVM зарежда .class файла за клас Welcome1
  - JVM изпълнява метода main()

Поправка на грешки :

Ако при изпълнение на Java програма, получите съобщение за грешка от сорта „Exception in thread “main” java.lang.NoClassDefFoundError : Welcome1,” то CLASSPATH ENVIRONMENT variable не е зададена правилно.

Обобщение :

- ОО е естествен начин на мислене за света и писането на компютърни програми.
- UML е графичен език,позволяващ ОО модели да се представят по един стандартен метод за използване на означения
- Обектите притежават свойството на скриване на информация,обектите в общия случай не знаят структурата на други обекти или как другите обекти се изпълняват.
- Обектно ориентираното програмиране позволява на програмистите да прилагат ООД в работещи компютърни ситеми.
- В Java,единицата на програмиране е **class** от който обектите евентуално се създават.Java програмистите се концентрират върху създаване на собствени класове и повторно използване на съществуващи класове.Всеки class съдържа данни и функции(методи) които манипулират тези данни.
- Една инстанция от **class** се нарича **обект**.
- Посредством ОО технология, програмистите изграждат софтуер,който в по-голямата част използва стандартни библиотеки от класове.

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

office@progressbg.net www.progressbg.net

Въпроси и задачи :

1) Кои от следните идентификатори са валидни:

boolean int j1 public 74Bus

$test for break breakfast

2*k continue stop go return goto

2) Обяснете разликата между клас променлива и обект

3) Напишете два коментара в Java,единият едноредов,а другия многоредов.

4) Обяснете какво представлява :

- Метода main()
- static метод
- static променливи(данни) на class