

Эволюция парадигм

Роман Неволин, Careem

SPB
DOT
NET

careem

Парадигма – это ...

Парадигма – это ключевая идея, лежащая в
основе построения чего-либо

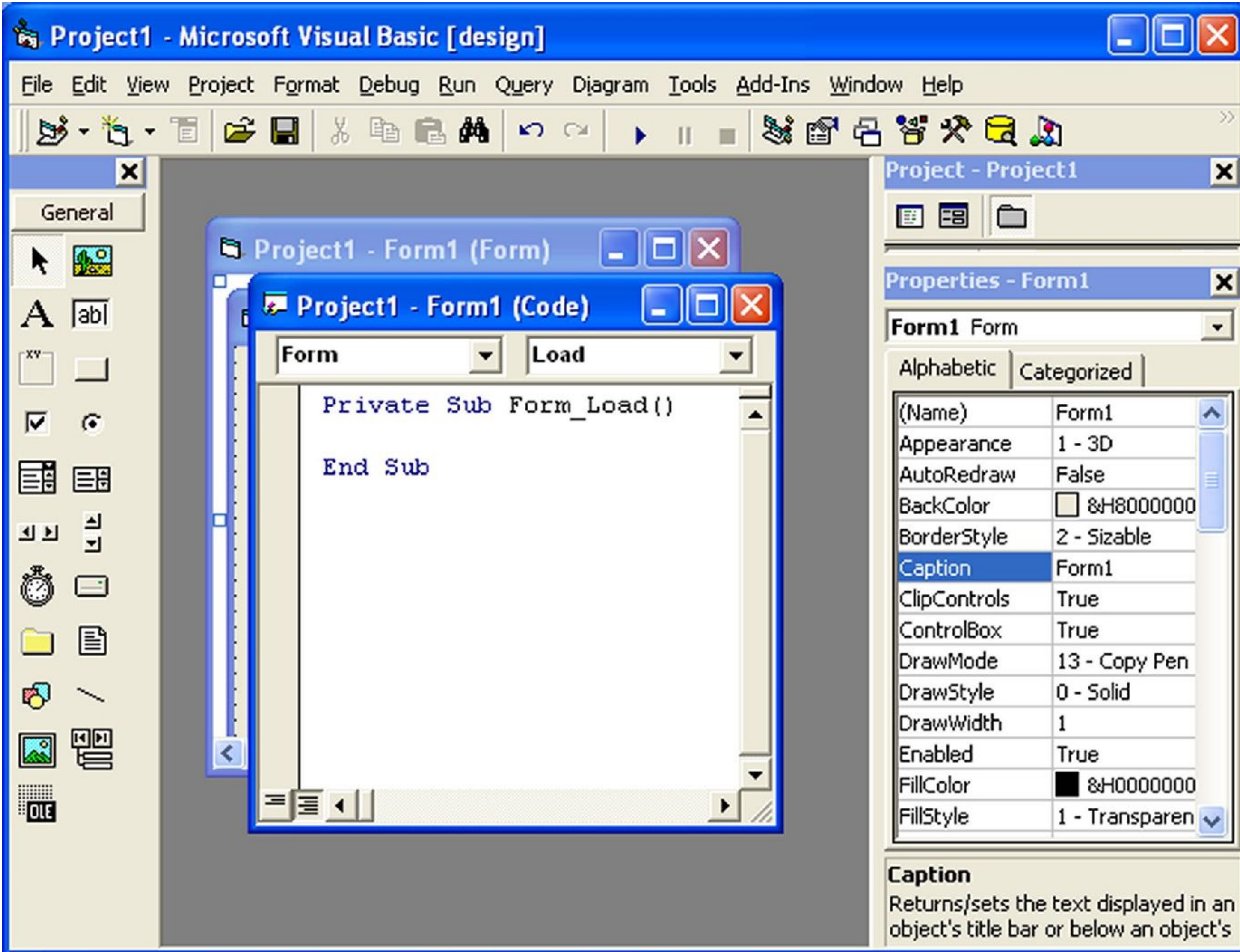
Парадигма программирования -
это совокупность идей и понятий,
определяющих подход к программированию

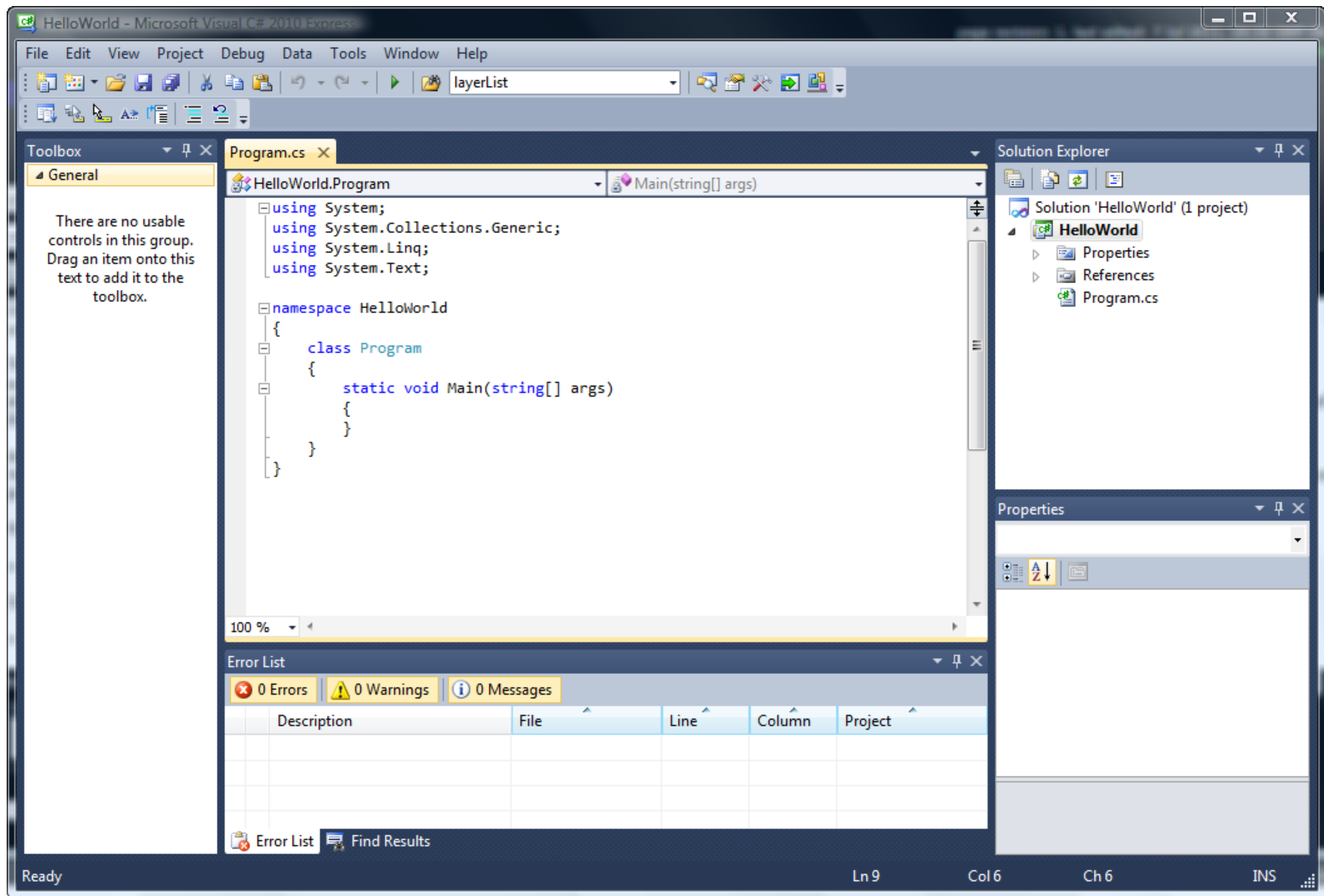
Untitled

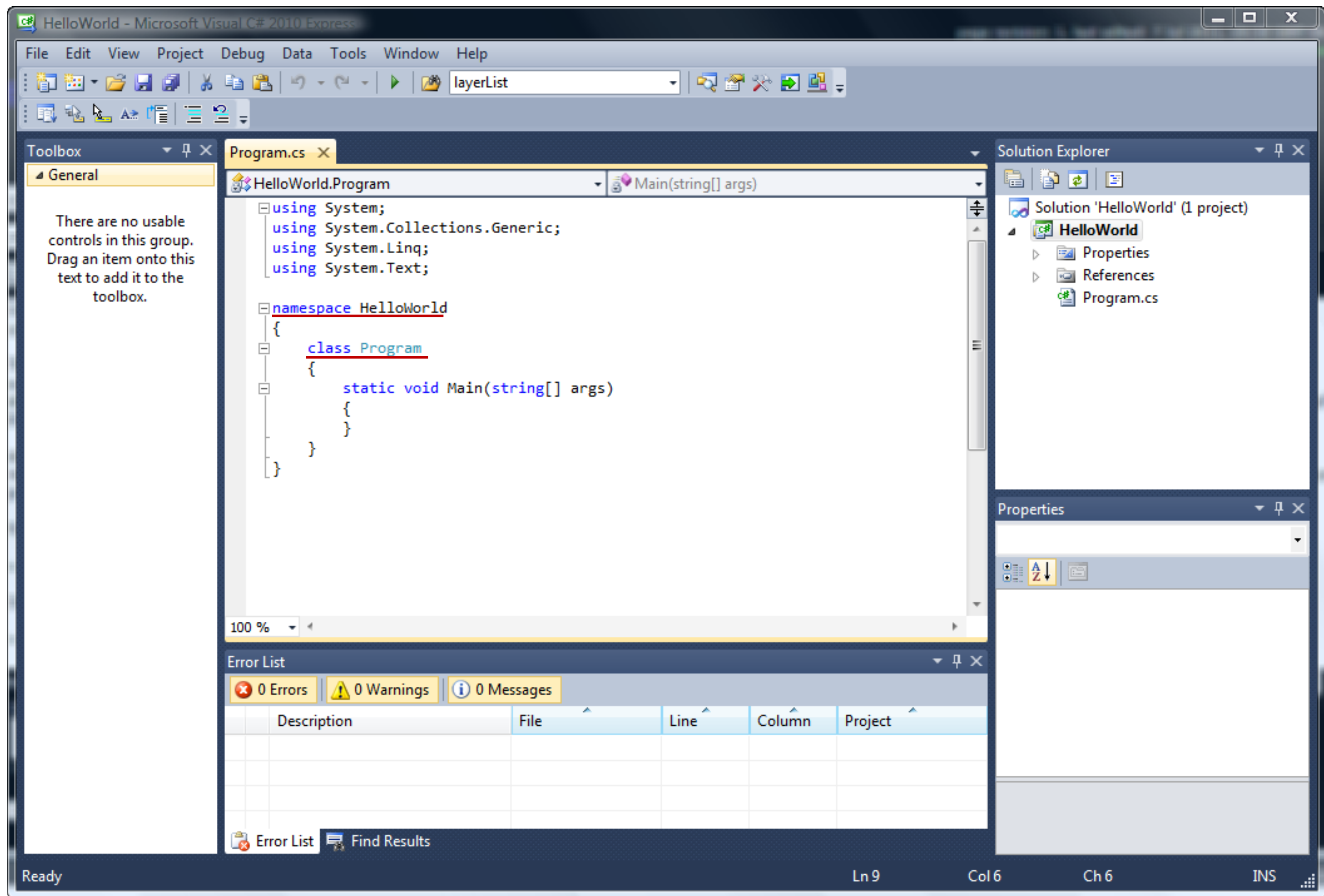


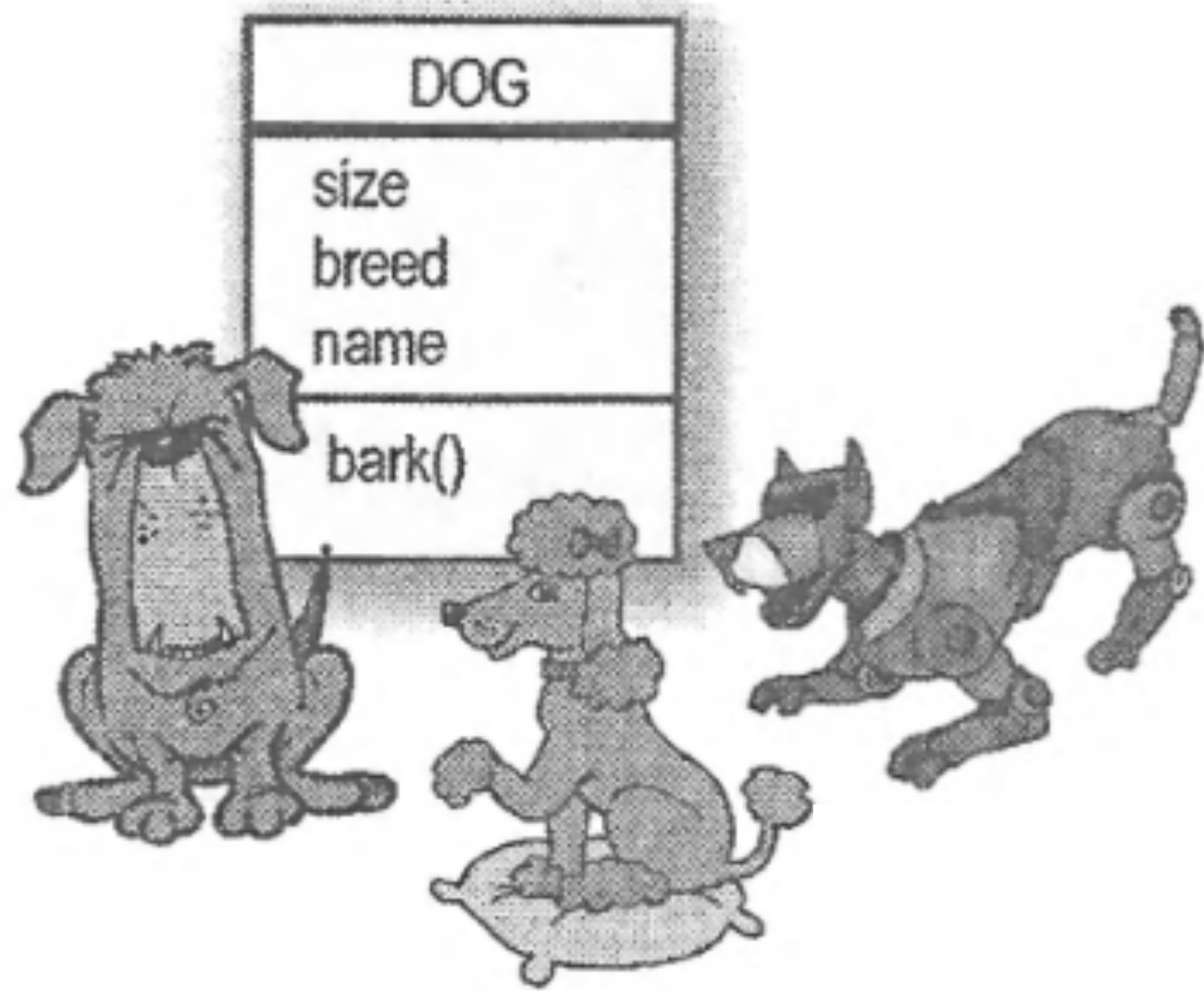
Immediate

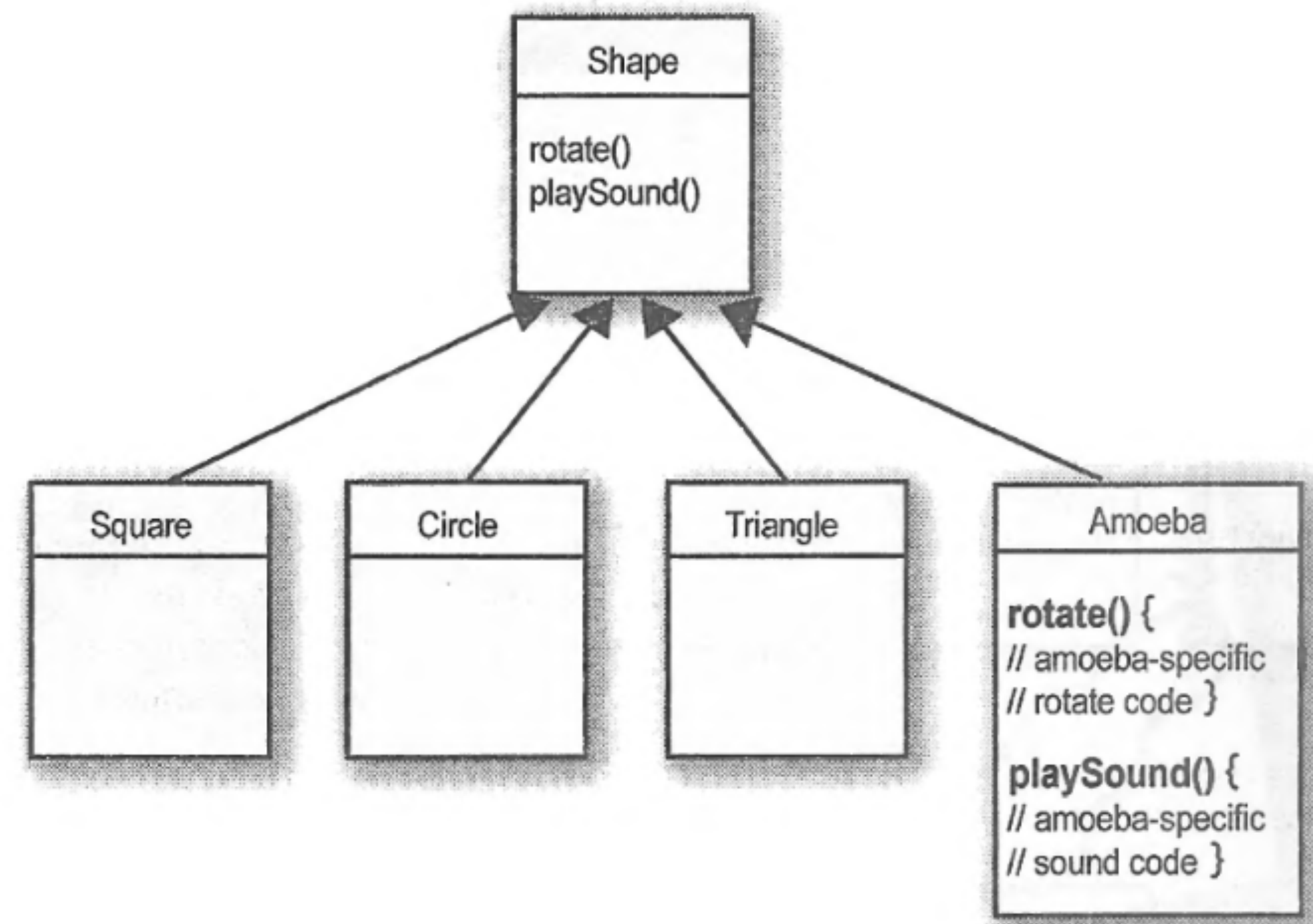
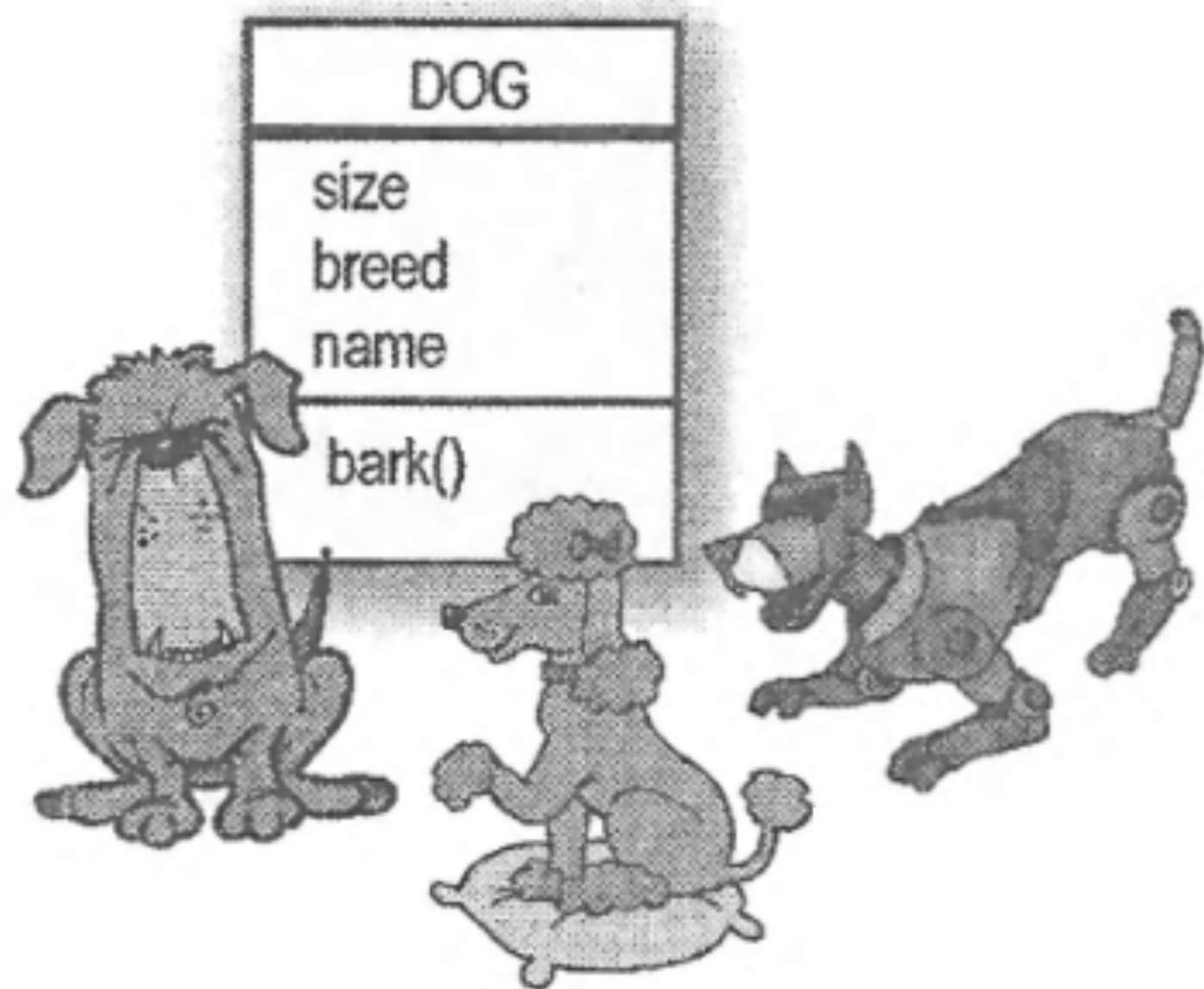
```
CLS  
i = 0  
10  
PRINT i  
i = i + 1  
IF (i < 10) GOTO 10
```

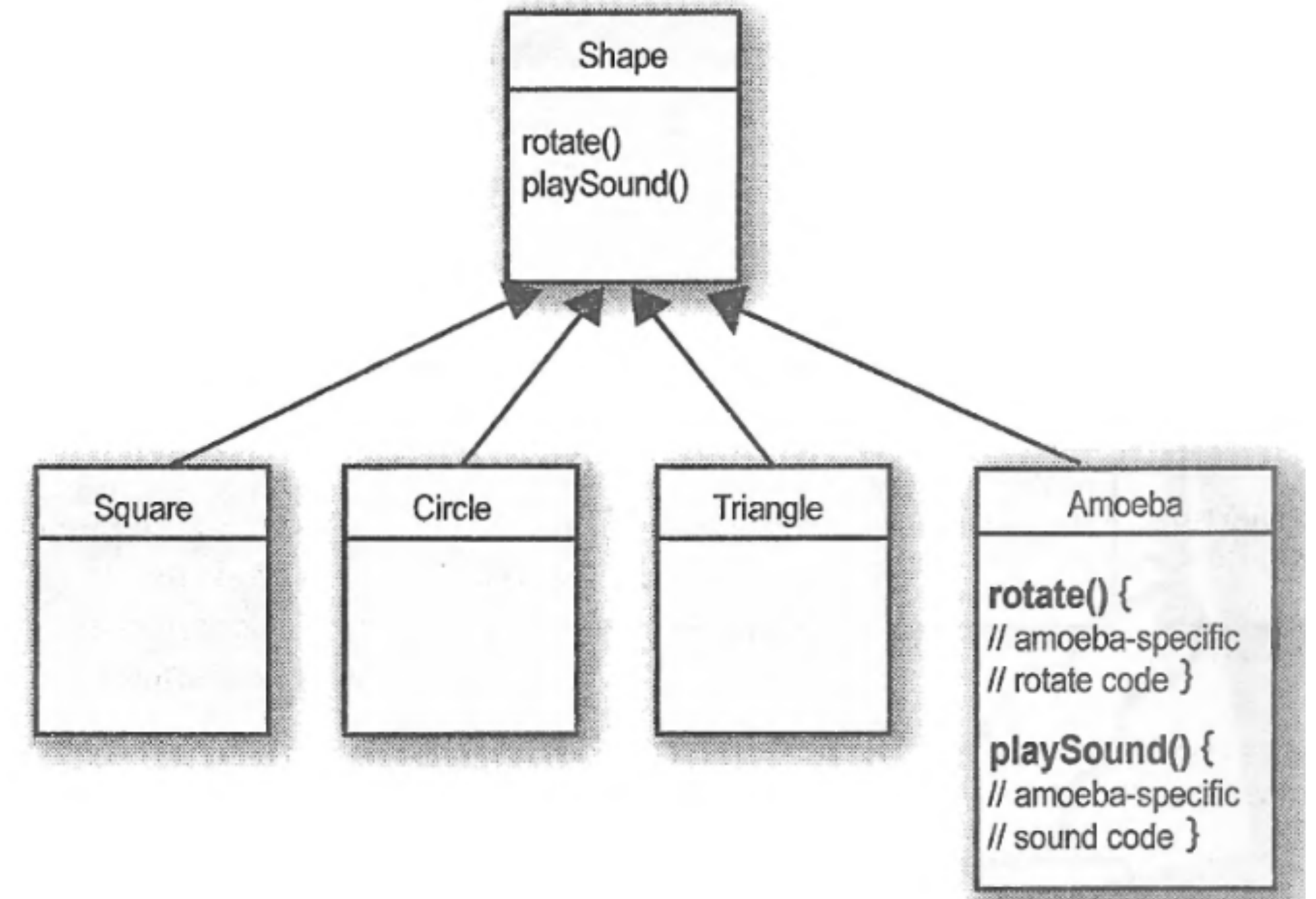
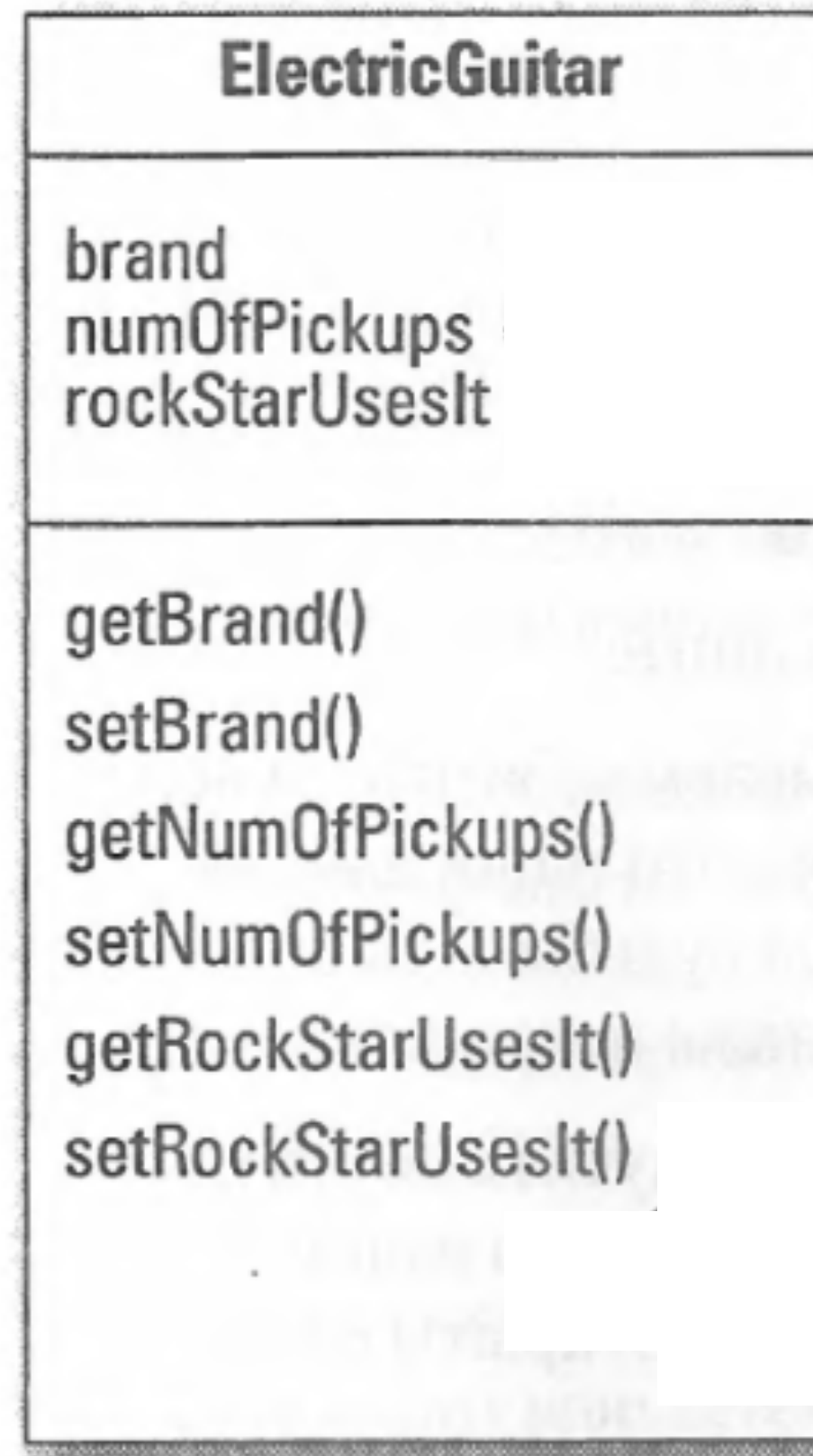
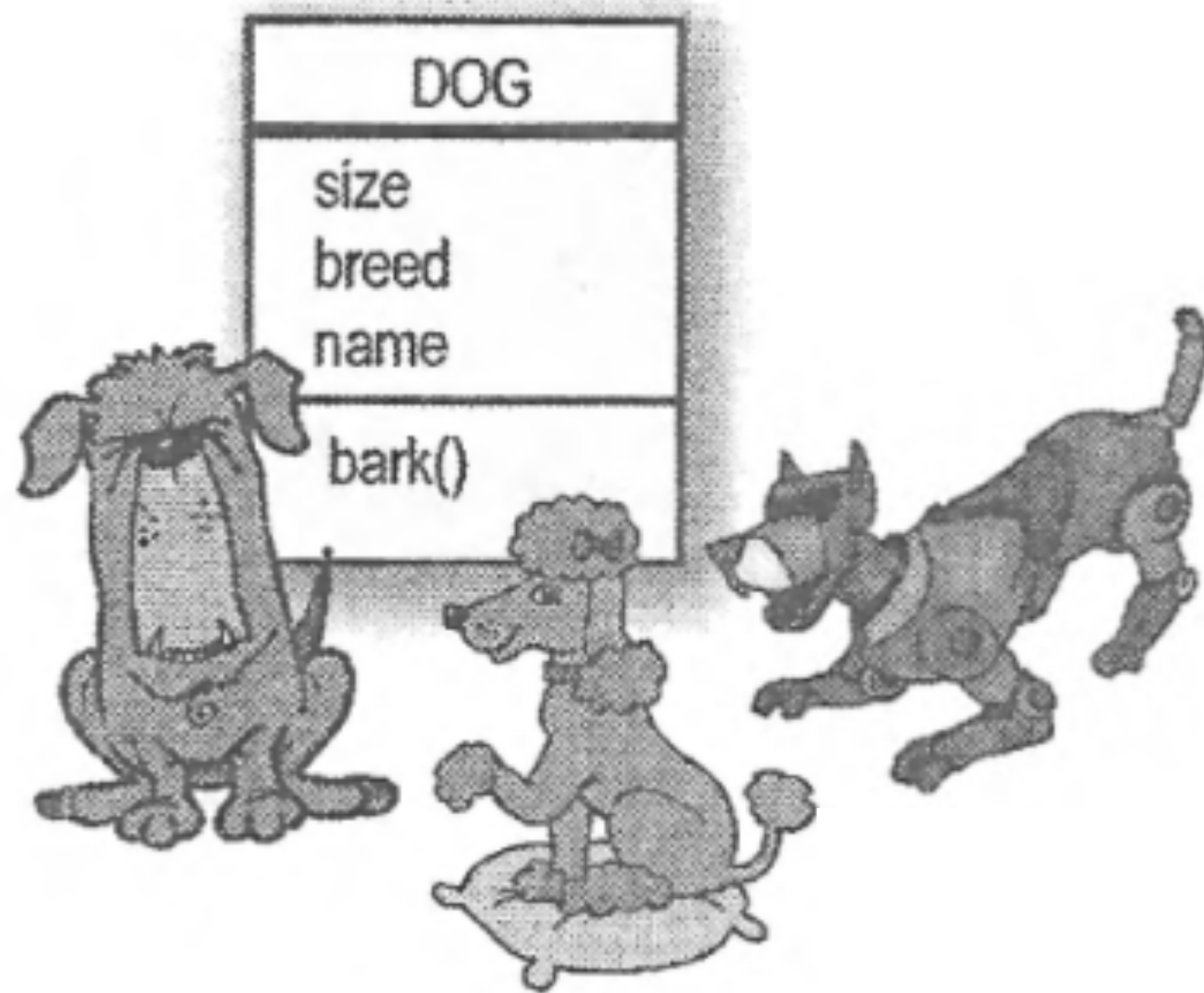



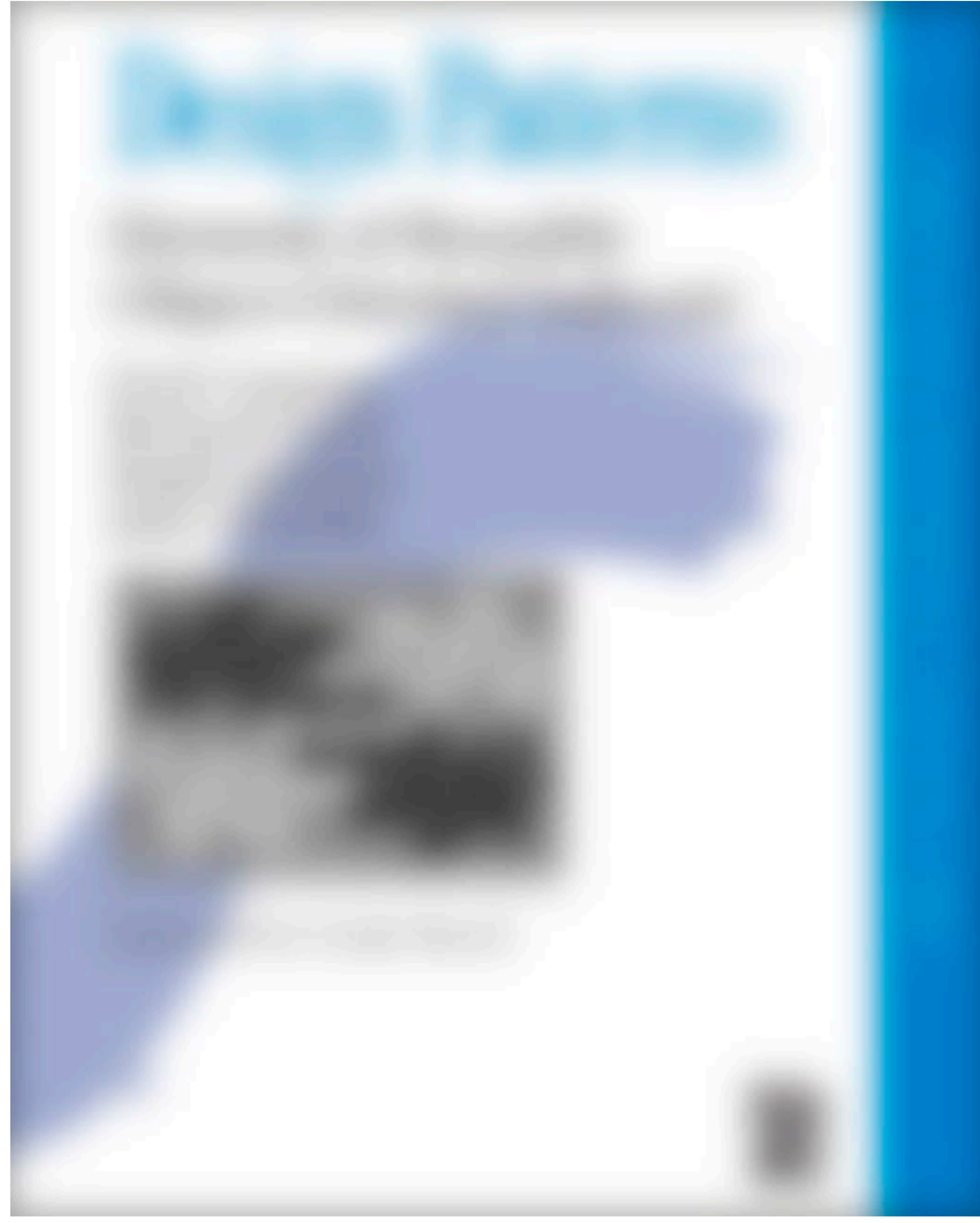








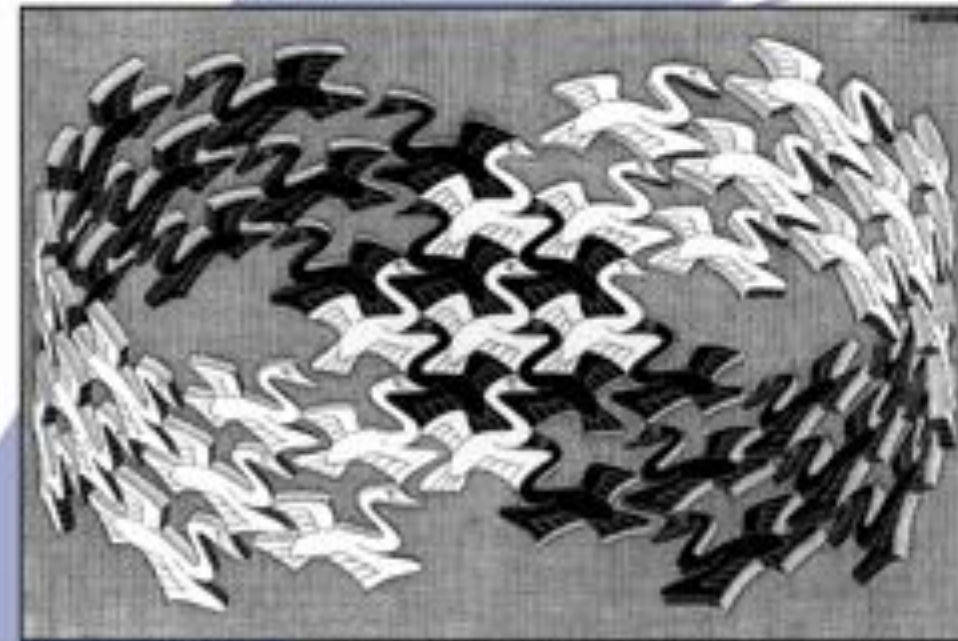




Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides



Cover art © 1994 M.C. Escher / Cordon Art - Baarn - Holland. All rights reserved.

Foreword by Grady Booch



✦ ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES



Patterns



OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also
- Functions

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also
- Functions
- Yes, functions

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also
- Functions
- Yes, functions
- Oh my, functions again!

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also
- Functions
- Yes, functions
- Oh my, functions again!
- Functions

OO pattern/principle

- Single Responsibility Principle
- Open/Closed principle
- Dependency Inversion Principle
- Interface Segregation Principle
- Factory pattern
- Strategy pattern
- Decorator pattern
- Visitor pattern

FP pattern/principle

- Functions
- Functions
- Functions, also
- Functions
- Yes, functions
- Oh my, functions again!
- Functions
- Functions 😊





Как выучить Haskell



Поиск в Google

Мне повезёт!

**AND THIS IS HOW
WE WRITE HELLO WORLD**

A photograph of a classroom. A man in a suit stands at the front, facing a large audience of students seated at desks. Behind him is a large chalkboard filled with handwritten mathematical equations and diagrams. The text 'AND THIS IS HOW WE WRITE HELLO WORLD' is overlaid in large white letters at the top, and 'IN HASKELL' is overlaid in large white letters at the bottom.

IN HASKELL



Как выучить Haskell и не сойти с ума



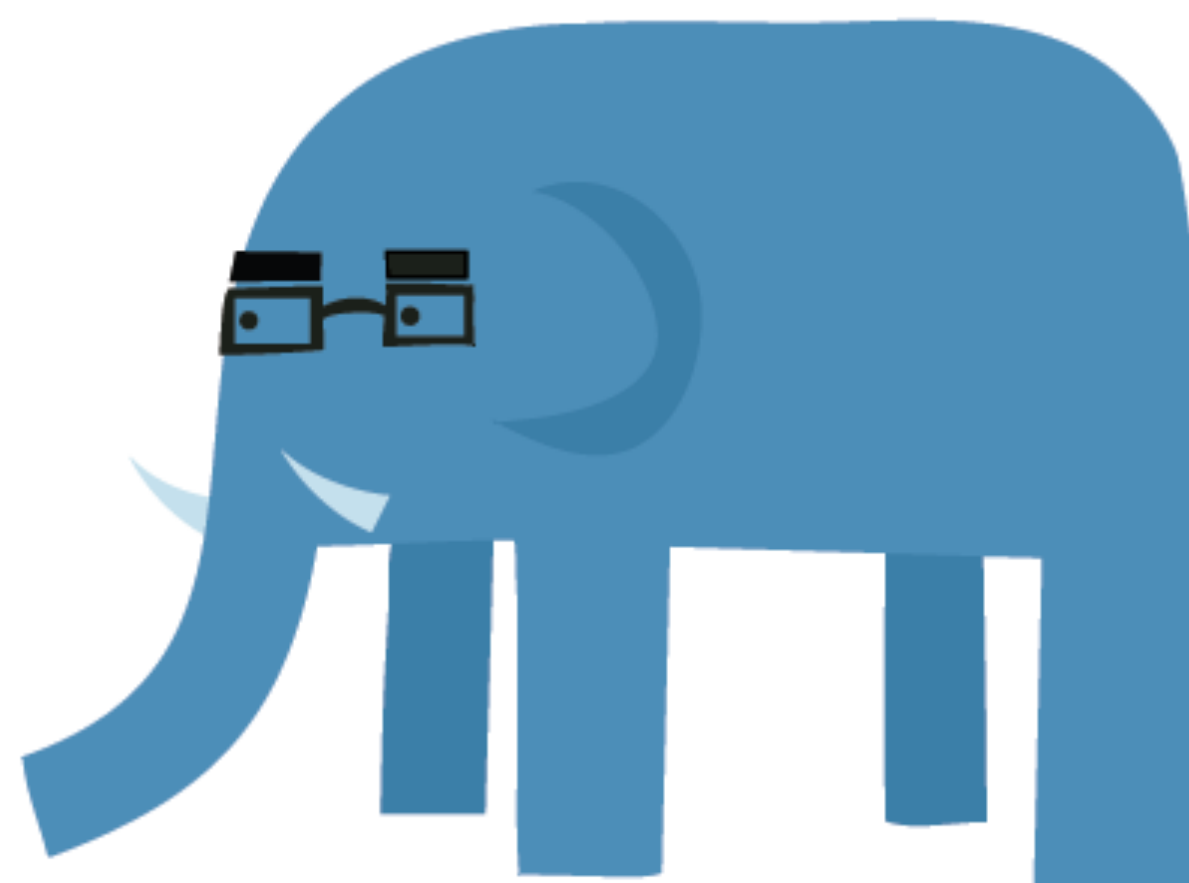
Поиск в Google

Мне повезёт!



Learn You a Haskell for Great Good!

A Beginner's Guide



Miran Lipovača



Стадии принятия функционального программирования

- «Я просто посмотрю»

Стадии принятия функционального программирования

- «Я просто посмотрю»
- «Что за ... ?»

Стадии принятия функционального программирования

- «Я просто посмотрю»
- «Что за ... ?»
- «О, а вот это выглядит интересно!»

Стадии принятия функционального программирования

- «Я просто посмотрю»
- «Что за ... ?»
- «О, а вот это выглядит интересно!»
- «Боже, я хочу все писать в функциональном стиле»

Функциоз (от англ. function, функция) – патологическое желание решать любую задачу в функциональном стиле, сопровождающееся непреодолимой потребностью удовлетворить это желание



Стадии принятия функционального программирования

- «Я просто посмотрю»
- «Что за ... ?»
- «О, а вот это выглядит интересно!»
- «Боже, я хочу все писать в функциональном стиле»
- «Я больше не хочу все писать функционально»

Ни одна парадигма не решает
ВСЕХ ваших проблем





The major features of C# 1.0 included:

- Classes
- Structs
- Interfaces
- Events
- Properties
- Delegates
- Expressions
- Statements
- Attributes
- Literals

The major features of C# 1.0 included:

- Classes
- Structs
- Interfaces
- Events
- Properties
- Delegates
- Expressions
- Statements
- Attributes
- Literals

The major features of C# 1.0 included:

- Classes
- Structs
- Interfaces
- Events
- Properties
- Delegates
- Expressions
- Statements
- Attributes
- Literals

C# version 6.0

With versions 3.0 and 5.0, C# had added major new features in an object-oriented language. With version 6.0, it would go away from doing a dominant killer feature and instead release many smaller features that made C# programming more productive. Here are some of them:

- Static imports
- Exception filters
- Property initializers
- Expression bodied members
- Null propagator
- String interpolation
- nameof operator
- Index initializers

C# version 7.0

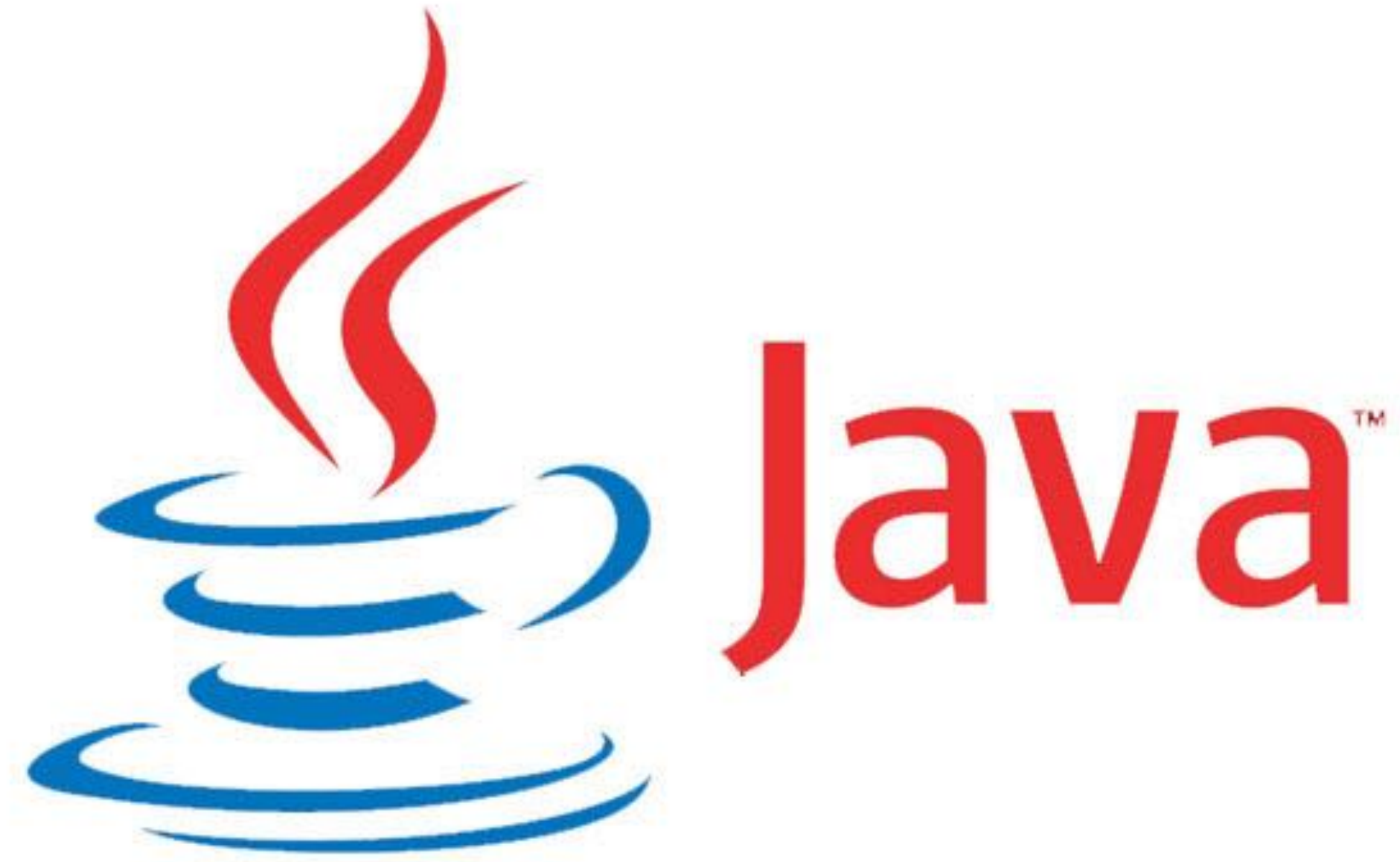
The most recent major version is C# version 7.0. This version has some evolutionary and cool stuff in the vein of C# 6.0, but without the compiler as a service. Here are some of the new features:

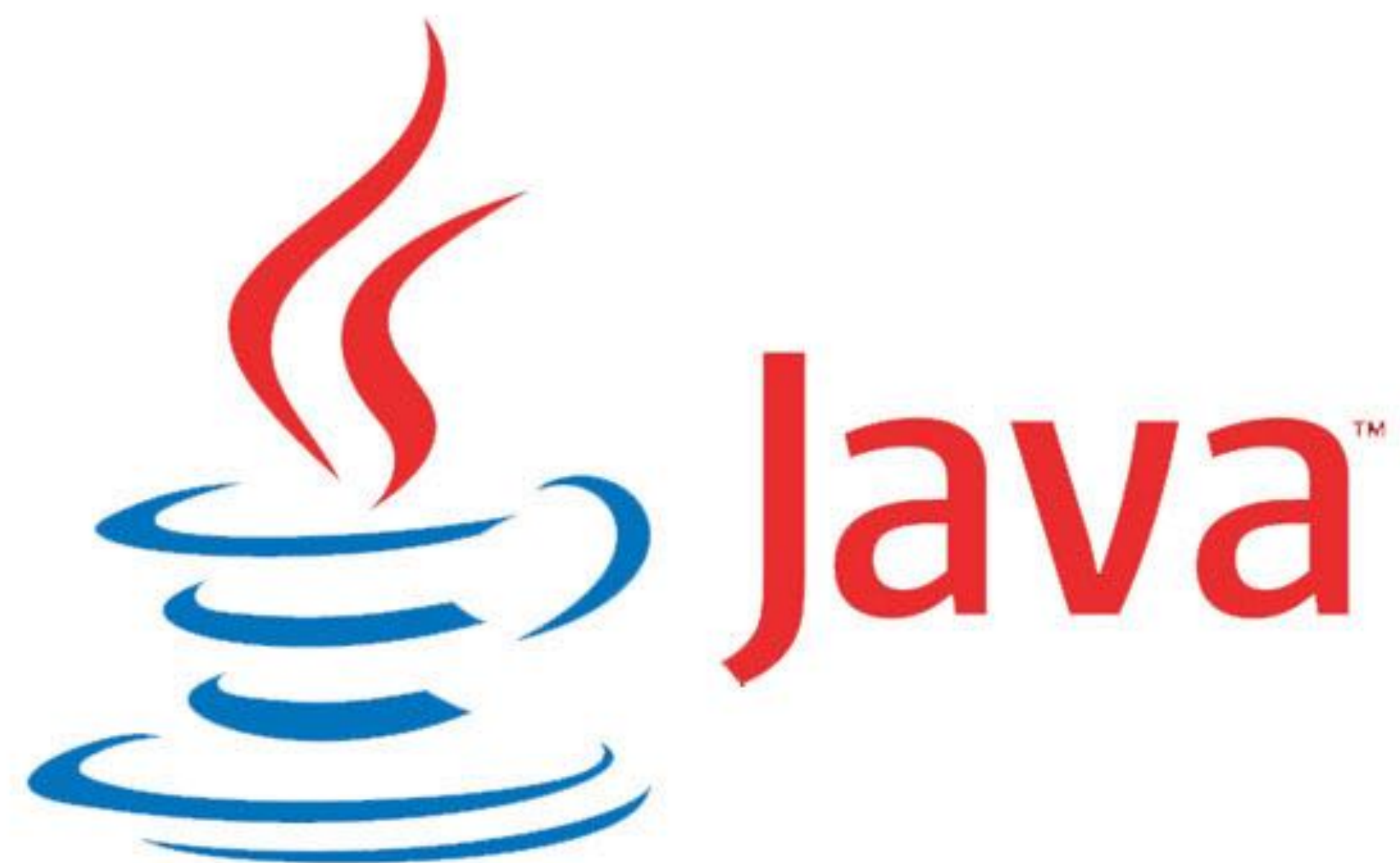
- Out variables
- Tuples and deconstruction
- Pattern matching
- Local functions
- Expanded expression bodied members
- Ref locals and returns

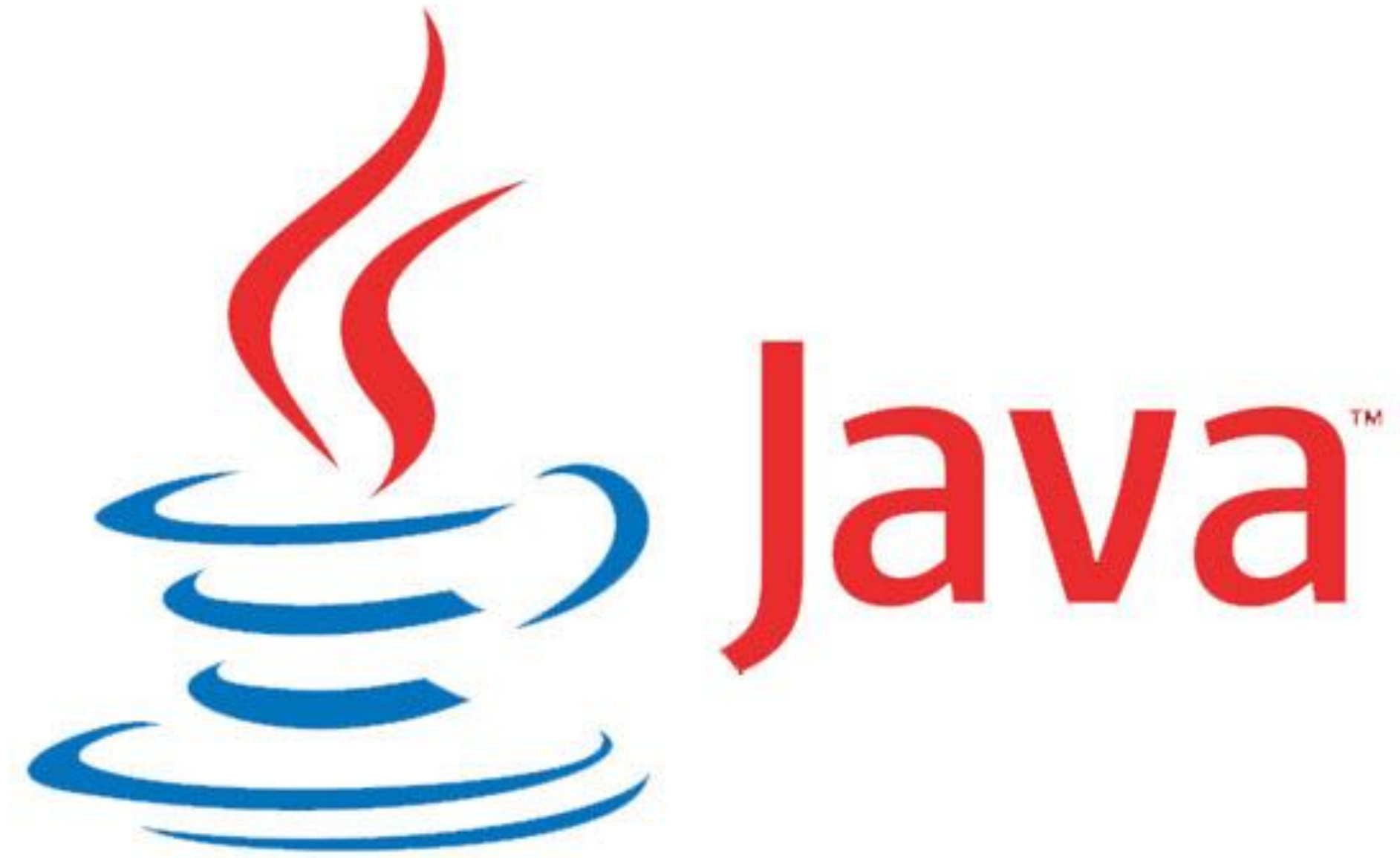
C# version 7.0

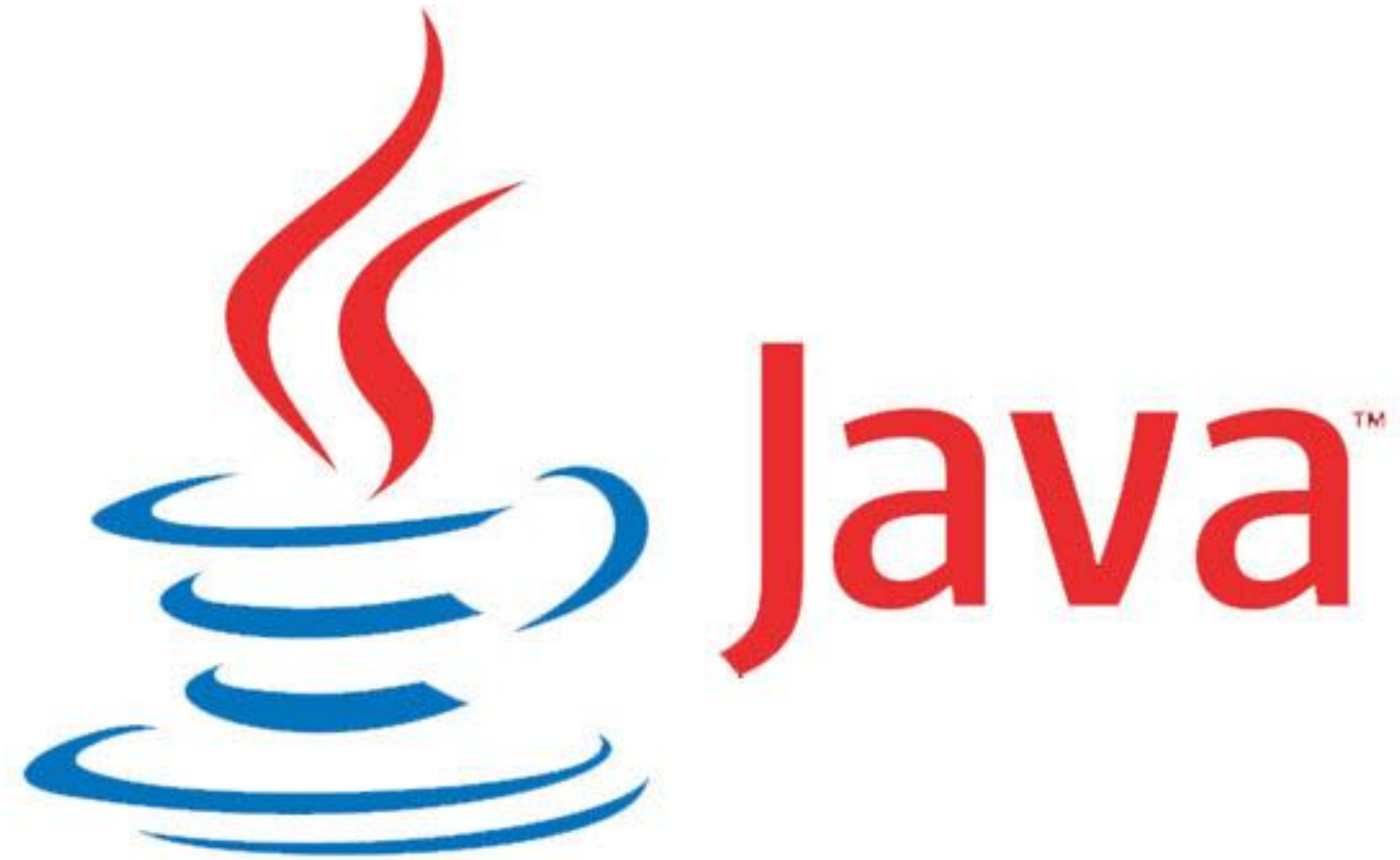
The most recent major version is C# version 7.0. This version has some evolutionary and cool stuff in the vein of C# 6.0, but without the compiler as a service. Here are some of the new features:

- [Out variables](#)
- [Tuples and deconstruction](#)
- [Pattern matching](#)
- [Local functions](#)
- [Expanded expression bodied members](#)
- [Ref locals and returns](#)









Principles

There were five primary goals in the creation of the Java language:

1. It must be "simple, object-oriented, and familiar".
2. It must be "robust and secure".
3. It must be "architecture-neutral and portable".
4. It must execute with "high performance".
5. It must be "interpreted, threaded, and dynamic".

Principles

There were five primary goals in the creation of the Java language:

1. It must be "simple, object-oriented, and familiar".
2. It must be "robust and secure".
3. It must be "architecture-neutral and portable".
4. It must execute with "high performance".
5. It must be "interpreted, threaded, and dynamic".

Converting array to list in Java



How do I convert an array to a list in Java?

761

I used the `Arrays.asList()` but the behavior (and signature) somehow changed from Java SE [1.4.2](#) (docs now in archive) to [8](#) and most snippets I found on the web use the 1.4.2 behaviour.



For example:



119

```
int[] spam = new int[] { 1, 2, 3 };  
Arrays.asList(spam)
```

- on 1.4.2 returns a list containing the elements 1, 2, 3
- on 1.5.0+ returns a list containing the array spam

Converting array to list in Java



How do I convert an array to a list in Java?

761

I used the `Arrays.asList()` but the behavior (and signature) somehow changed from Java SE [1.4.2](#) (docs now in archive) to [8](#) and most snippets I found on the web use the 1.4.2 behaviour.



For example:



119

```
int[] spam = new int[] { 1, 2, 3 };  
Arrays.asList(spam)
```

- on 1.4.2 returns a list containing the elements 1, 2, 3
- on 1.5.0+ returns a list containing the array spam



In Java 8, you can use streams:

96

```
int[] spam = new int[] { 1, 2, 3 };  
Arrays.stream(spam)  
    .boxed()  
    .collect(Collectors.toList());
```



[share](#) [improve this answer](#)

answered May 16 '15 at 23:49



[Ibrahim Arief](#)

5,669 ● 3 ● 27 ● 49



how to read private field



Все

Видео

Новости

Карты

Картинки

Ещё

Настройки

Инструменты

Результатов: примерно 325 000 000 (1,31 сек.)

How to read the value of a private field from a different class in ...

<https://stackoverflow.com/.../how-to-read-the-value-of-a-pri...> ▼ Перевести эту страницу

10 ответов

23 янв. 2018 г. - In order to access **private fields**, you need to get them from the class's Reflection isn't the only **way** to resolve your issue (which is to access ...

java - **How** do I access **private** methods and ...

6 ответов

21 февр. 2017 г.

Java reflection get all **private fields**

6 ответов

9 мар. 2013 г.

Is it possible in Java to access **private fields** via ...

3 ответа

12 окт. 2009 г.

c# - Find a **private field** with Reflection?

10 ответов

18 сент. 2008 г.

Другие результаты с сайта stackoverflow.com

How to read the value of a private field from a different class in Java?



389



102

I have a poorly designed class in a 3rd-party `JAR` and I need to access one of its **private** fields. For example, why should I need to choose private field is it necessary?

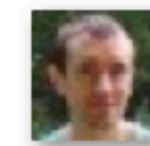
```
class IWasDesignedPoorly {  
    private Hashtable stuffIWant;  
}  
  
IWasDesignedPoorly obj = ...;
```

How can I use reflection to get the value of `stuffIWant` ?

`java` `class` `reflection` `field` `private`

[share](#) [improve this question](#)

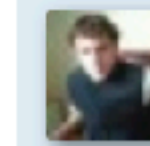
edited Jan 23 at 13:51



[Steve Chambers](#)

20.6k ● 11 ● 87 ● 132

asked Jul 28 '09 at 19:20



[Frank Krueger](#)

43.1k ● 39 ● 141 ● 195

1 is there any way to prevent this? – [BigDataScholar](#) Jun 2 '16 at 19:55

[add a comment](#)

10 Answers

[active](#)

[oldest](#)

[votes](#)

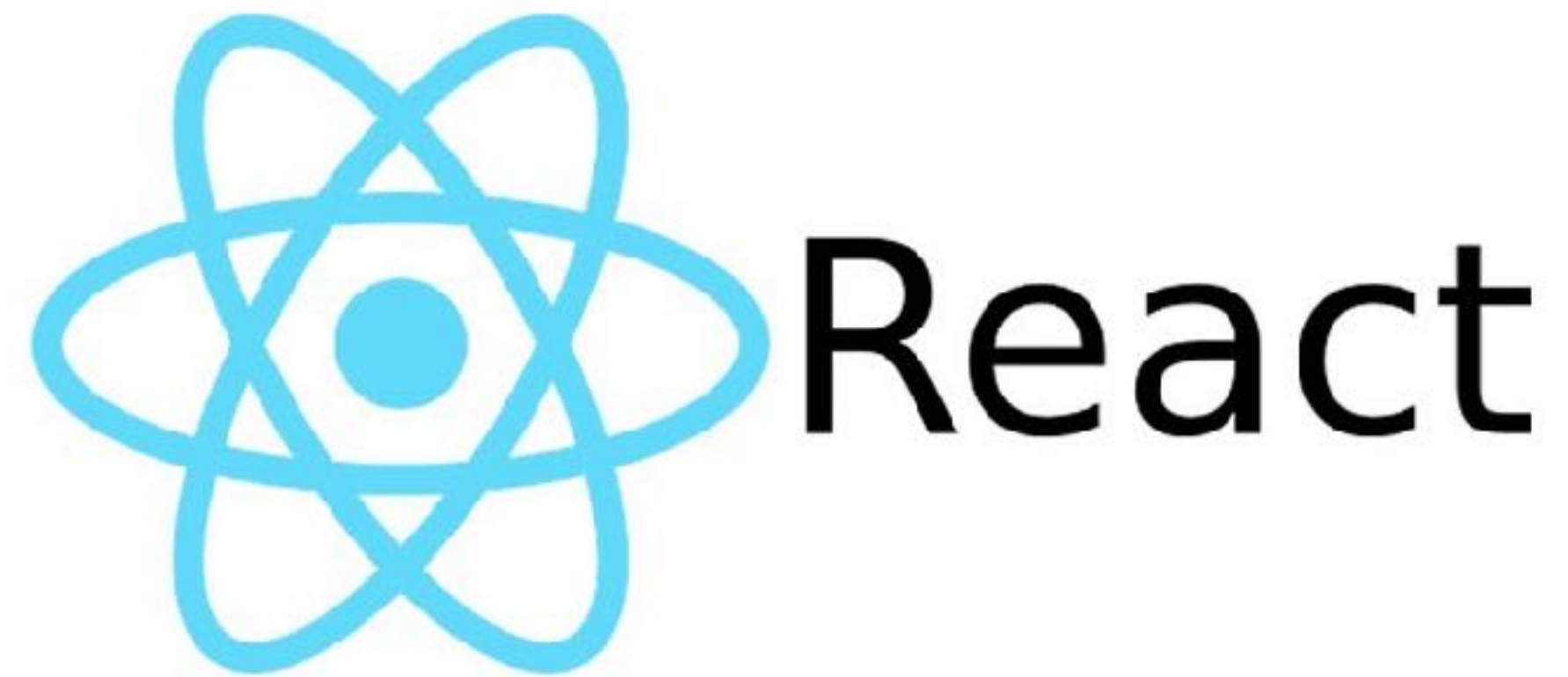


547



In order to access private fields, you need to get them from the class's *declared* fields and then make them accessible:

```
Field f = obj.getClass().getDeclaredField("stuffIWant"); //NoSuchFieldException  
f.setAccessible(true);  
Hashtable iWantThis = (Hashtable) f.get(obj); //IllegalAccessException
```





Shay Friedman

@ironshay



Follow

Drinking game for web devs:

(1) Think of a noun

(2) Google "<noun>.js"

(3) If a library with that name exists - drink

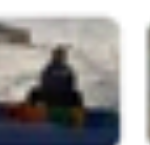


RETWEETS

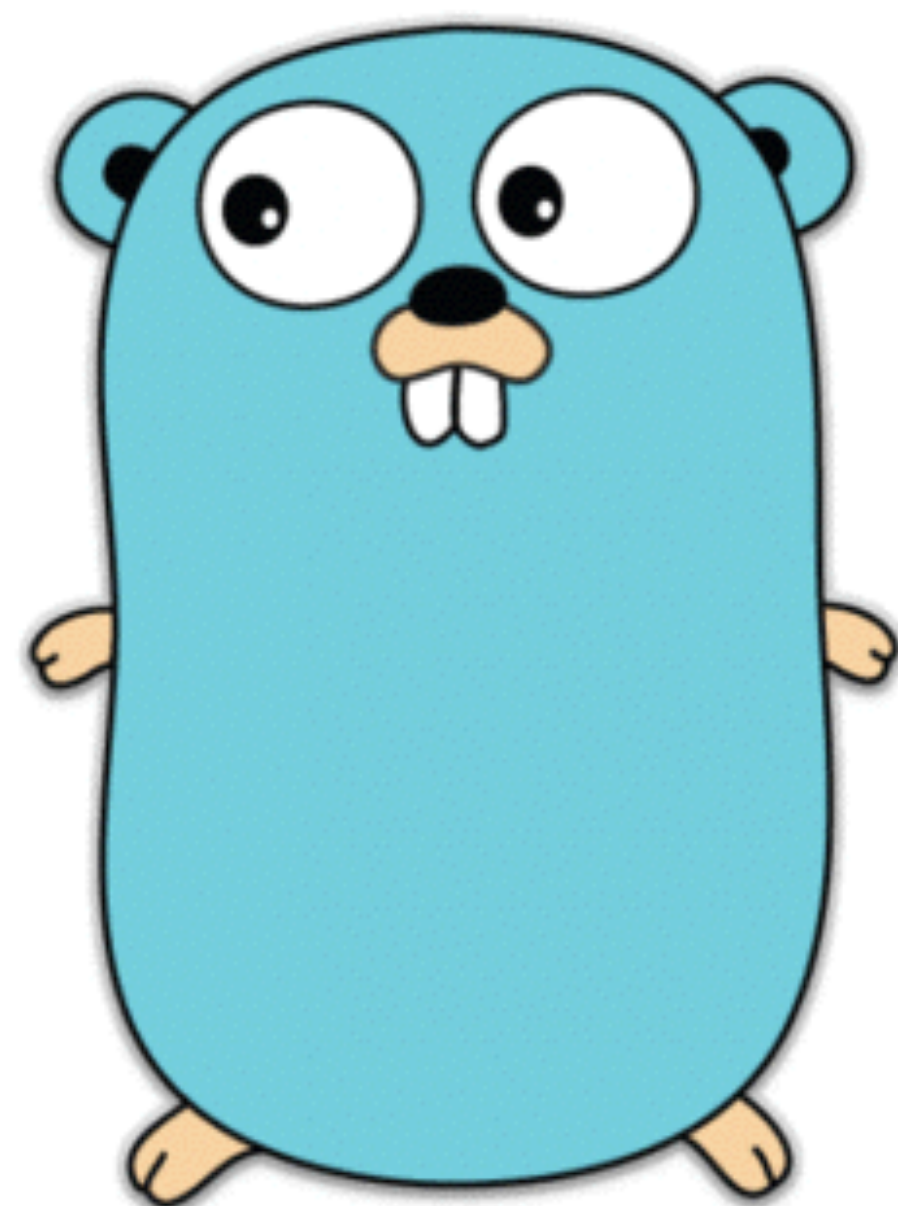
2,075

FAVORITES

667



5:40 AM - 22 Aug 2013



Go

«Каждый язык имеет свои задачи и
область применения»

— *Какой-то senior developer из какой-то компании*

«Мы используем микросервисы, потому
что микросервисная архитектура
отлично вписывается в наш продукт»

— СТО еще-одного-стартапа

Итак, проблемы

- Вариативность современных языков усложняет их изучение и понимание

Итак, проблемы

- Вариативность современных языков усложняет их изучение и понимание
- Исключить эту вариативность довольно сложно, потому что программисты сильно изобретательные

Итак, проблемы

- Вариативность современных языков усложняет их изучение и понимание
- Исключить эту вариативность довольно сложно, потому что программисты сильно изобретательные
- Выбирать язык под задачу стало куда труднее, чем прежде

Вариант 1 :

«Мы дадим программистам максимальную
свободу и вариативность, а они сами
выберут нужные им инструменты»

Вариант 2 :

«Мы дадим программистам минимальный
достаточный набор инструментов, и тогда
они будут решать проблемы единообразно»

Вырабатывайте стандарты
под проект

Какие стандарты?

- Стилистические
- Подмножества языка
- Границы парадигм



Не набирайте «программистов
на языке ...»

«Я Java-разработчик с девятью
годами опыта веб-разработки»

«В смысле писать
функционально?»»

А что нужно программисту?

- Быть в курсе идей, а не языков

А что нужно программисту?

- Быть в курсе идей, а не языков
- Уметь решать проблемы

А что нужно программисту?

- Быть в курсе идей, а не языков
- Уметь решать проблемы
- Уметь решать проблемы вместе с коллегами

Используйте автоматические
инспекции кода

Автоматические инспекции?

- Переложите заботу о следовании выбранному стилю на IDE

Автоматические инспекции?

- Переложите заботу о следовании выбранному стилю на IDE
- Используйте условный SonarQube с правильными инспекциями

Автоматические инспекции?

- Переложите заботу о следовании выбранному стилю на IDE
- Используйте условный SonarQube с правильными инспекциями
- А для всего, что не проверяется автоматически, есть ревью

Изучайте новые языки и идеи,
которые они в себе несут

«A language that doesn't affect the way you think
about programming, is not worth knowing.»

— *Alan Perlis*

Не следуйте парадигмам –
создавайте парадигмы

Спасибо