

Роман Неволин

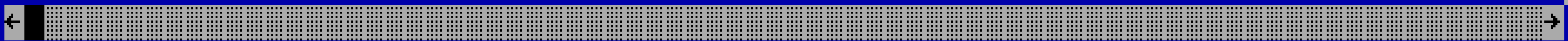
В поисках парадигмы

Парадигма – это...

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

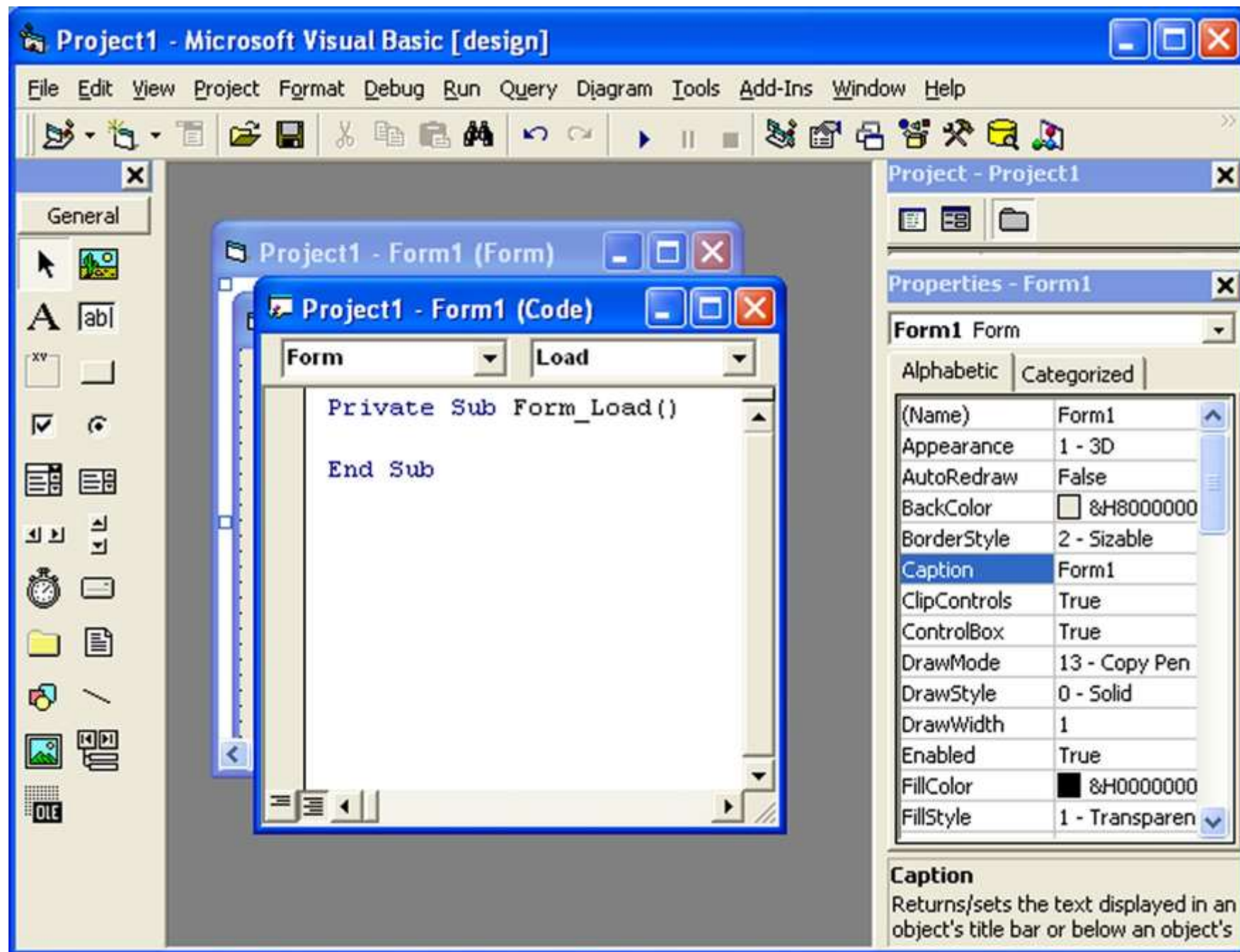
**Парадигма программирования —
это совокупность идей и понятий,
определяющих подход к
программированию**

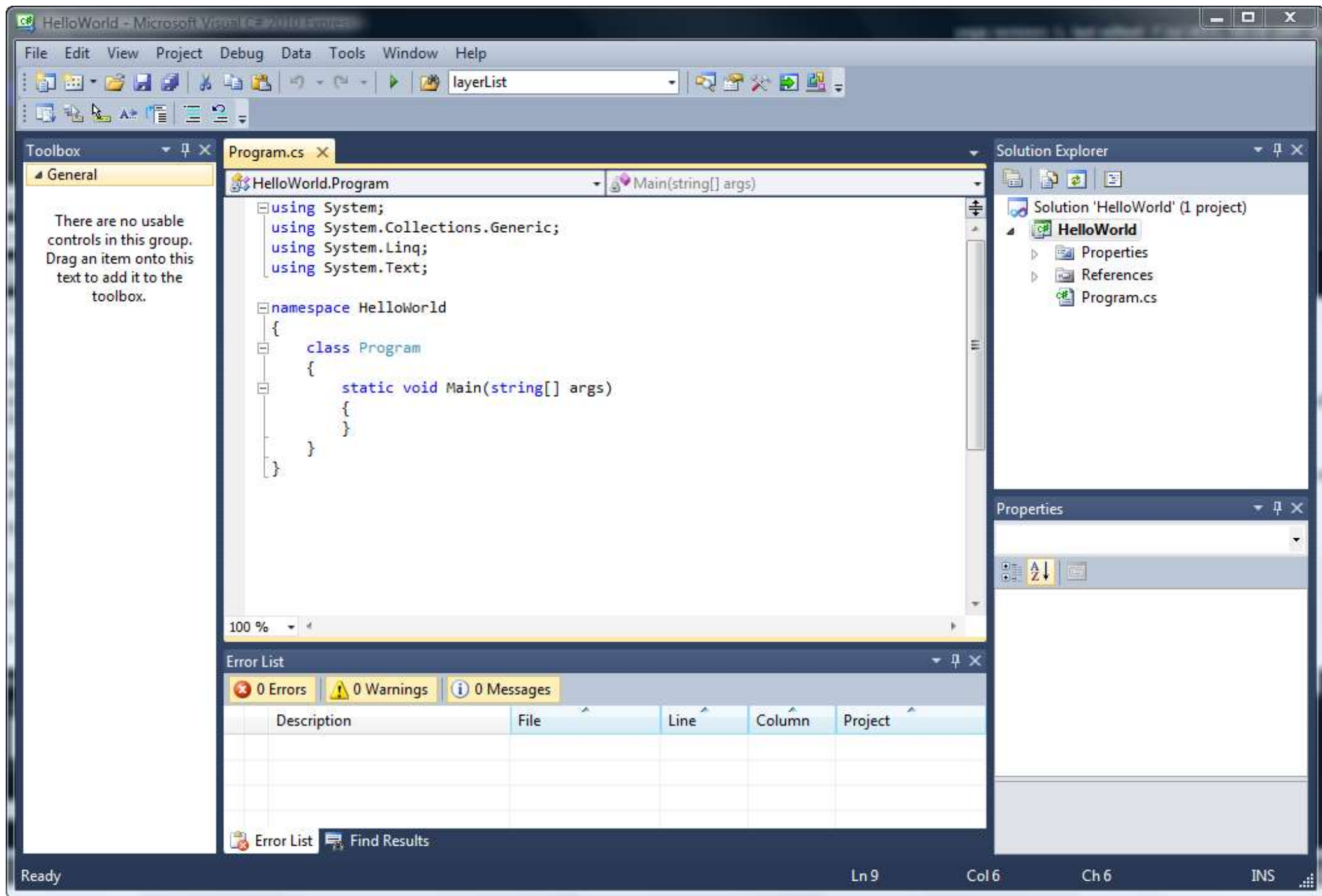
Untitled

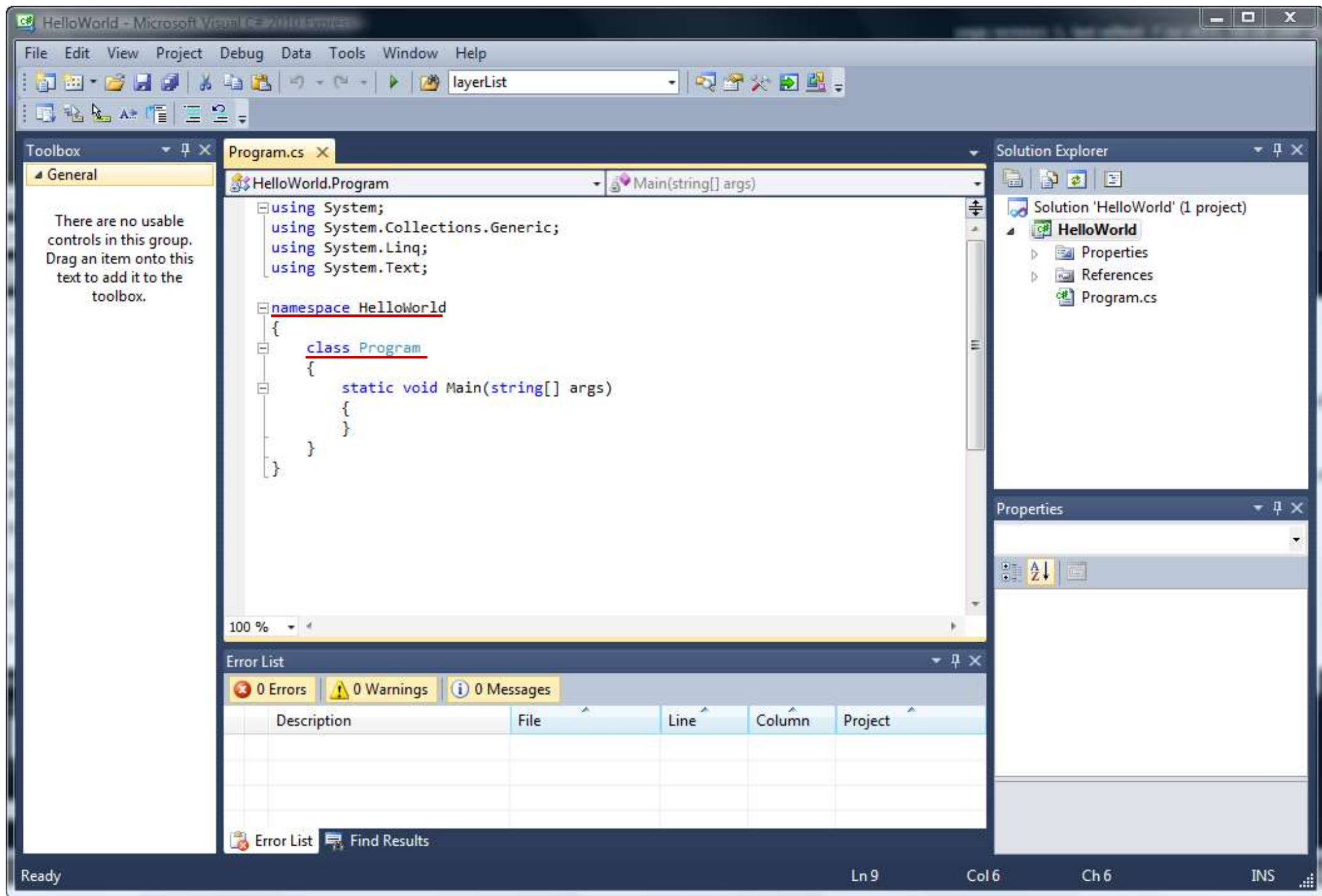


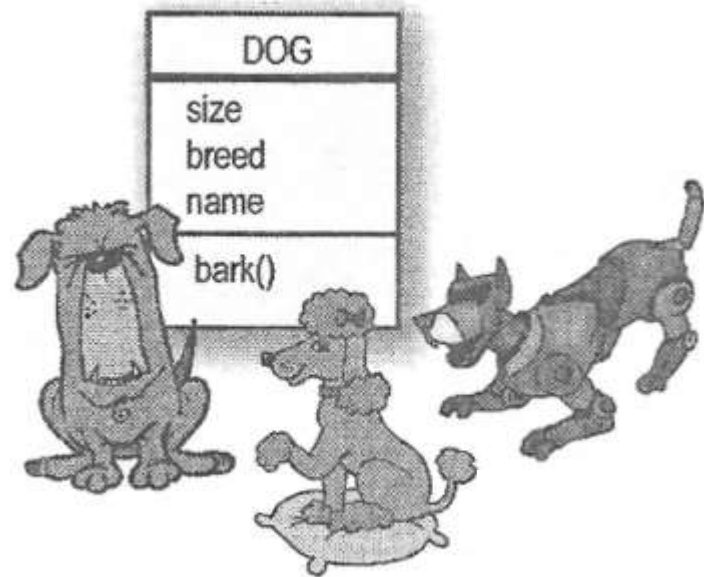
Immediate

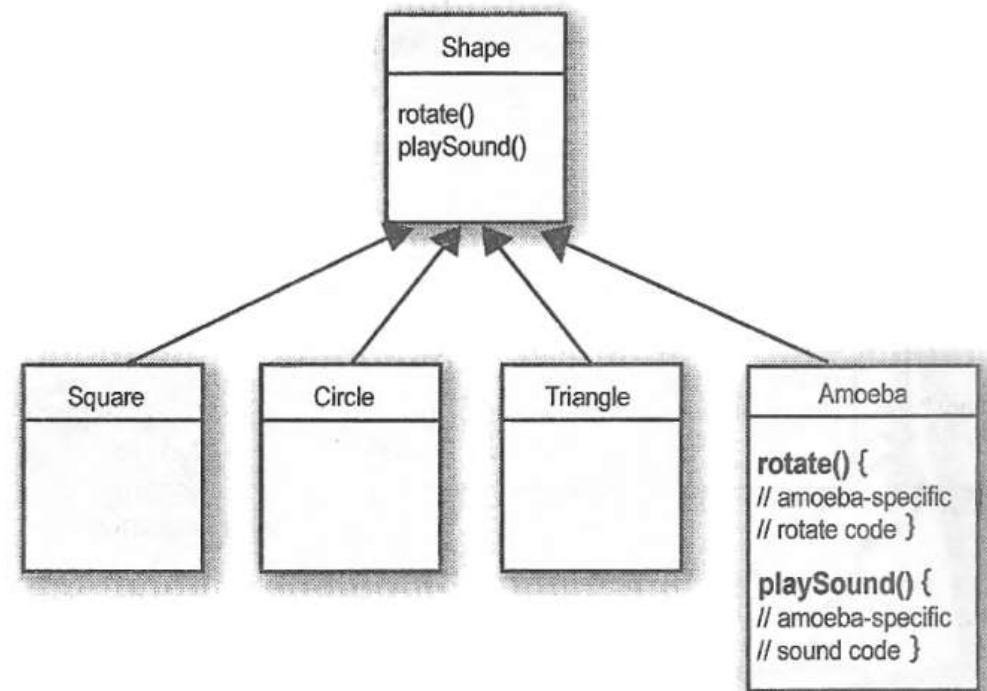
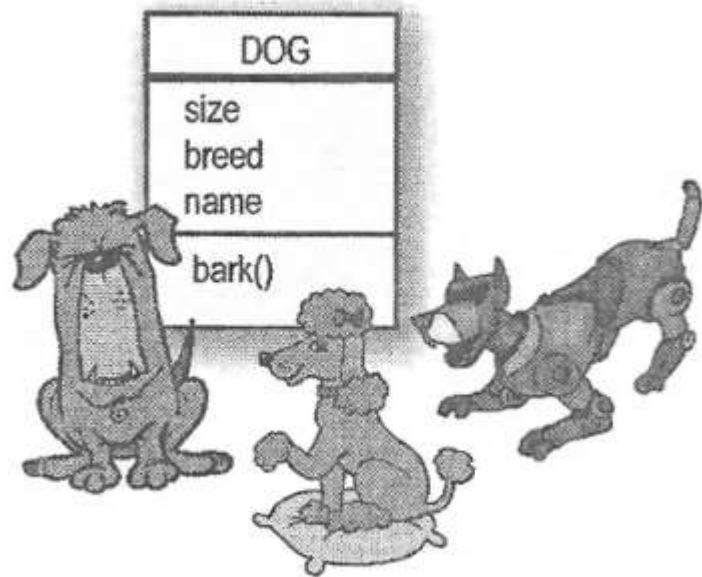
```
CLS  
i = 0  
10  
PRINT i  
i = i + 1  
IF (i < 10) GOTO 10
```

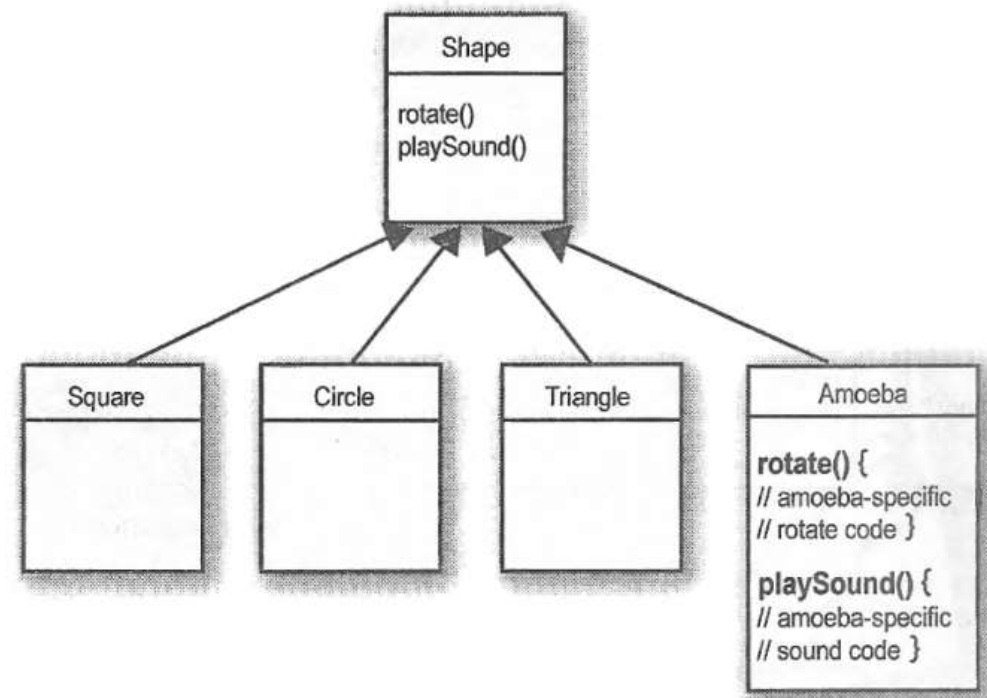
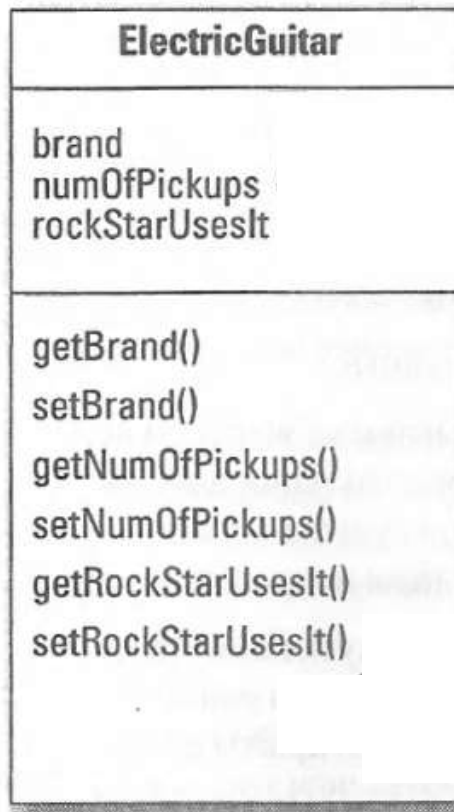
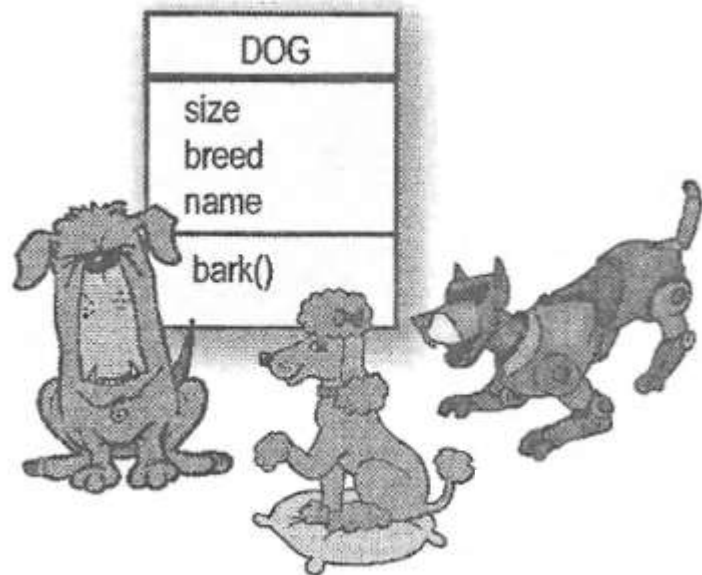










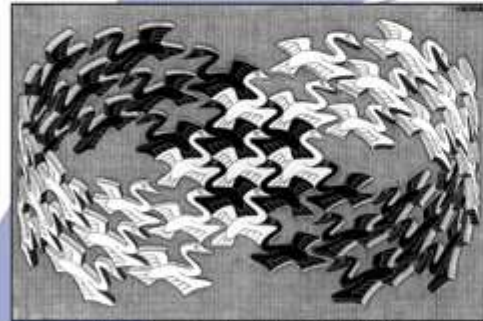




Design Patterns

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





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



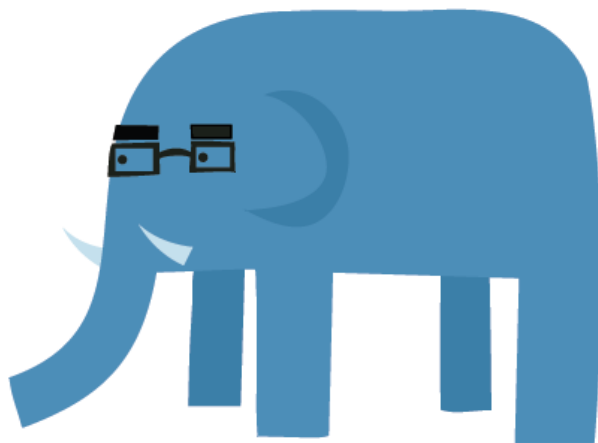
Поиск в Google

Мне повезёт!



Learn You a Haskell for Great Good!

A Beginner's Guide



Miran Lipovača



Стадии принятия функционального программирования

- Что это за чушь?

Стадии принятия функционального программирования

- Что это за чушь?
- ЧТО ЭТО ЗА ЧУШЬ?

Стадии принятия функционального программирования

- Что это за чушь?
- ЧТО ЭТО ЗА ЧУШЬ?
- А вот это выглядит интересно...

Стадии принятия функционального программирования

- Что это за чушь?
- ЧТО ЭТО ЗА ЧУШЬ?
- А вот это выглядит интересно...
- Хм, а вроде не так и сложно

Стадии принятия функционального программирования

- Что это за чушь?
- ЧТО ЭТО ЗА ЧУШЬ?
- А вот это выглядит интересно...
- Хм, а вроде не так и сложно
- ГОСПОДИ, ПОЧЕМУ ВСЕ НЕ ПИШУТ
ФУНКЦИОНАЛЬНО?!

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



Программирование — это решение реальных задач

Реальные задачи редко живут в рамках одной
парадигмы

Вы постоянно пишете мультипарадигменный код


```
public int Max(int a, int b)
{
    if (a > b) return a;
    else return b;
}
```

```
public int Max(int a, int b)
{
    return new If(
        new GreaterThen(a, b),
        a, b
    );
}
```

Глупо оставаться в рамках одной парадигмы

Глупо оставаться в рамках одной парадигмы

Вы все равно используете другие парадигмы,
просто делаете это неумело

Расширяйте свою область
знаний, изучайте новые
подходы.

Спасибо!