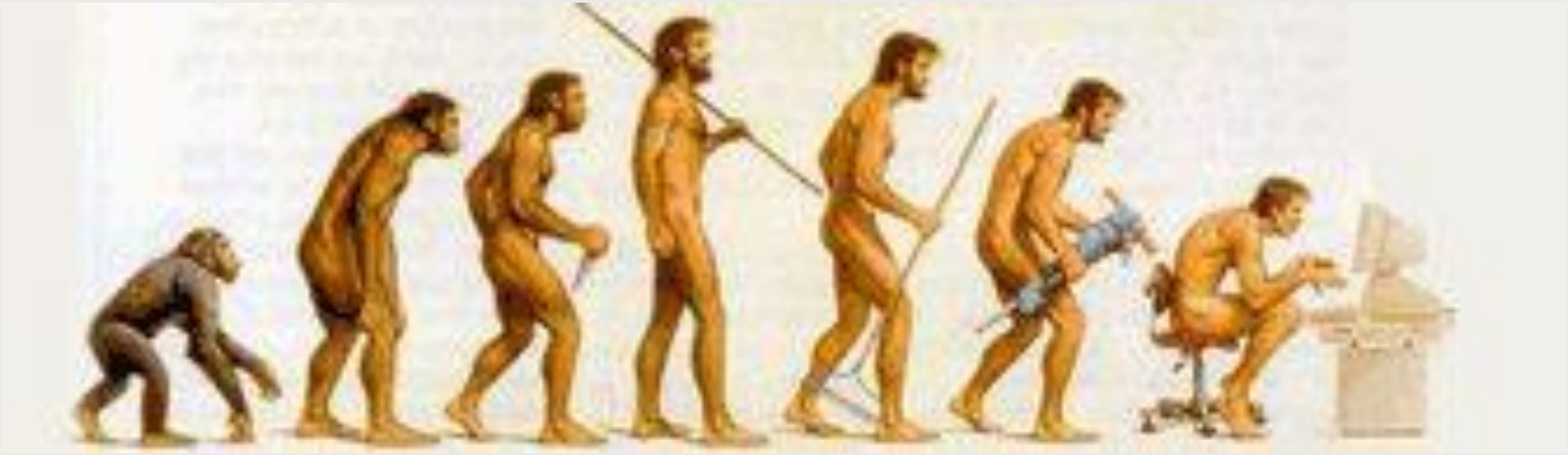
Основы программирования

Павловская **Т**атьяна **А**лександровна

профессор кафедры **и**нформатики и **п**рикладной **м**атематики (ауд. 378, тел.: (812)233-4690)

e-mail: pta-ipm@yandex.ru

Материалы на сайте: http://pta-ipm.narod.ru



Задача курса - обучение основам создания коммерческих программных продуктов:

- разработка эффективных **алгоритмов**
- выбор наиболее подходящих **языковых средств**
- выбор оптимальных или близких к ним **структур данных**
- профессиональное **кодирование** и **тестирование**

- введение в **технологии разработки ПО**

Содержание курса

- Парадигмы программирования
- Типы данных и управляющие конструкции языка высокого уровня
- Структуры данных
  - массивы, стеки, очереди, списки, деревья
- Объектно-ориентированное программирование
  - теория ООП
  - основы программирования под Windows
  - библиотеки классов

Программист должен обладать способностью первоклассного математика к абстракции и логическому мышлению в сочетании с эдисоновским талантом сооружать всё, что угодно, из нуля и единиц.

Он должен сочетать аккуратность бухгалтера с проницательностью разведчика, фантазию автора детективных романов с трезвой практичностью экономиста, сочетать в себе лёгкость и полет таланта Моцарта с усидчивостью и скрупулёзностью Сальери.

Академик Андрей Петрович Ершов

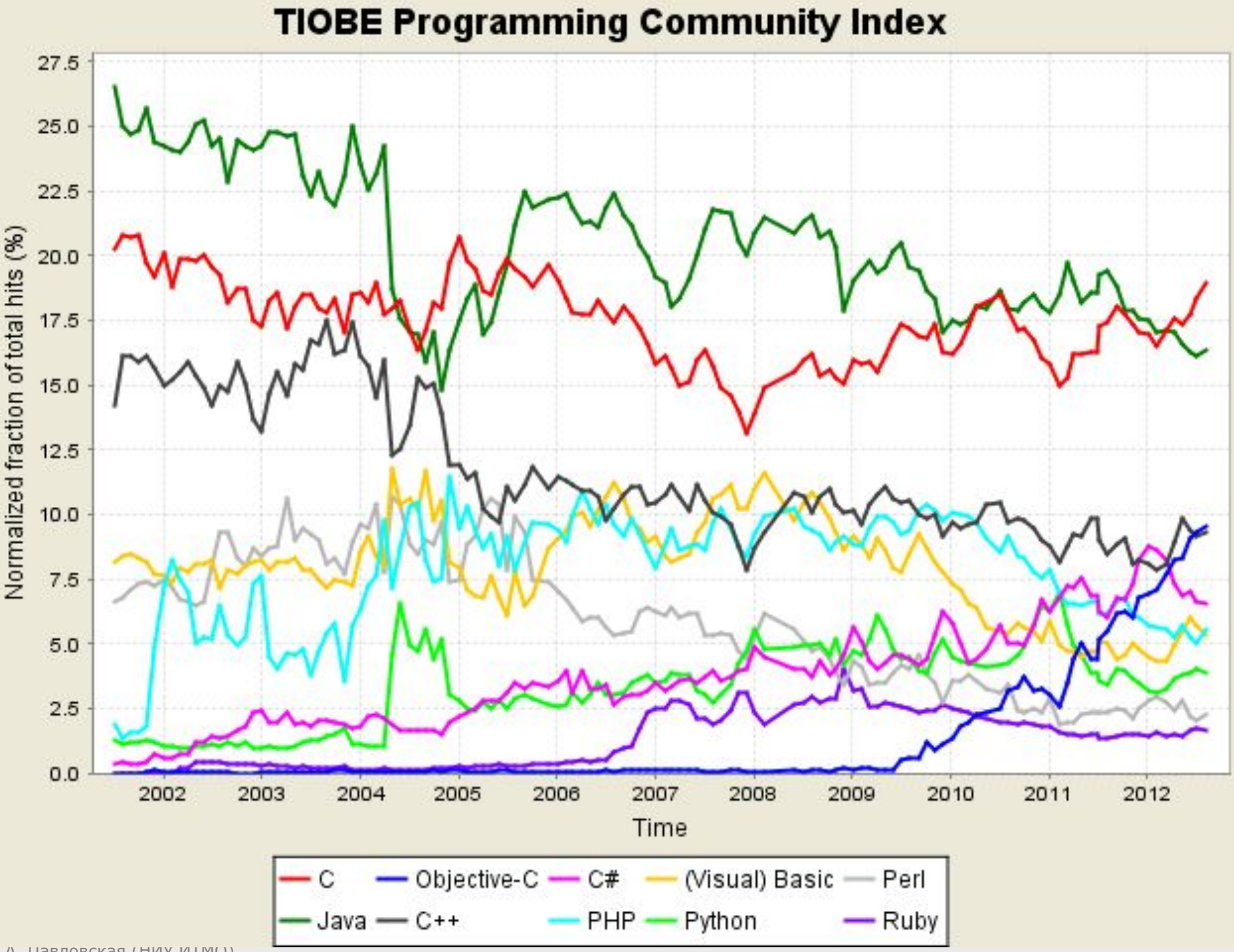
TIOBE Programming Community Index for August 2012

www.tiobe.com

Indicator of the popularity of programming languages.

The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. The popular search engines Google, Bing, Yahoo!, Wikipedia, YouTube and Baidu are used to calculate the ratings.

Position Aug 2012	Position Aug 2011	Programming Language	Ratings Aug 2012	Delta Aug 2011
1	2	C	18.937%	+1.55%
2	1	Java	16.352%	-3.06%
3	6	Objective-C	9.540%	+4.05%
4	3	C++	9.333%	+0.90%
5	5	C#	6.590%	+0.55%
6	4	PHP	5.524%	-0.61%
7	7	(Visual) Basic	5.334%	+0.32%
8	8	Python	3.876%	+0.46%
9	9	Perl	2.273%	-0.04%
10	12	Ruby	1.691%	+0.36%
11	10	JavaScript	1.365%	-0.19%
12	13	Delphi/Object Pascal	1.012%	-0.06%
13	14	Lisp	0.975%	+0.07%
14	26	Visual Basic .NET	0.877%	+0.41%


TIOBE Programming Community Index
Normalized fraction of total hits (%)
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
Time
C
Objective-C
C#
(Visual) Basic
Perl
Java
C++
PHP
Python
Ruby

Парадигмы (категории языков)

Category	Ratings August 2012	Delta August 2011
Object-Oriented Languages	57.1%	+1.8%
Procedural Languages	37.8%	-0.4%
Functional Languages	3.4%	-1.5%
Logical Languages	1.7%	+0.1%

Виды учебной нагрузки

1. **Лекции**
   - Модули I-II (осень) - 1 пара по нечет, 2 пары по чет.
   - Модули III-IV (весна) - 1 пара 1 раз в 2 недели
2. **Лаб. работы** - 1 пара 1 раз в неделю
3. **Самостоятельная работа** - круглосуточно
4. **Тестирование в ЦДО** – 1 в каждом модуле (ауд. 101)
5. **Рубежный контроль** – 1 в каждом модуле, состоит из теор. и практич. части
6. **Экзамены** (1 осень, 1 весна = тестирование + решение задачи, письменно, ручкой на бумаге)

БАРС: расчет рейтинга для модуля I (9 недель)

Вид учебной нагрузки	**баллы**	
	min	**max**
Вып. лаб. работ (3 штуки)	9 (**3** * 3 шт)	15 (**5** * 3 шт)
Рубежный контроль (тесты + защита лаб. работ)	6	10
Тестирование в ЦДО	7	12
Личностные качества	3	5
Итог	25	42

	МОДУЛЬ 1										МОДУЛЬ 2										Промежуточная аттестация	
	Текущий контроль по точкам								Рубежный контроль		Текущий контроль по точкам								Рубежный контроль			
	1		2		3		4				1		2		3		4					
	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max	[min]	max
Лабораторные работы			3	**5**	3	**5**	3	**5**					3	**5**	3	**5**	3	**5**				
Аттестующий тест					7	**12**									5	**8**						
Рубежное тестирование									6	**10**									6	**10**		
Экзамен																					12	**20**
Личностные качества							**3**	5									**3**	5				
Итого:									**25**	42											**60**	**100**

Темы лабораторных работ

1. Линейные программы

   (вычисление по двум формулам)

   - Константы
   - Типы данных
   - Переменные
   - Операции
   - Выражения
   - Исключительные ситуации
   - Простейший ввод-вывод
2. Разветвляющиеся программы (2 задания)
3. Циклические программы (3 задания)

Художественная литература по программированию (must read)

- Брукс Ф. Мифический человеко-месяц, или как создаются программные комплексы.
- Макконнелл С. Совершенный код (есть на моем сайте в разделе «Технологии программирования»).
- Спольски Дж. Джоэл о программировании (www. joelonsoftware.com).

«Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки в воздухе и из воздуха, творя силой воображения. <...> Однако программная конструкция, в отличие от поэтических творений, реальна, в том смысле, что она движется и работает, производя видимые результаты, которые отделимы от самой конструкции. Она печатает результаты, рисует картинки, производит звуки, приводит в движение рычаги. В наше время осуществилось волшебство мифа и легенды. С клавиатуры вводится верное заклинание, и экран монитора оживает, показывая то, чего никогда не было и не могло быть».

Ф. Брукс

«Чтобы стать специалистом, настоящим экспертом лишь в одной области программирования, нужны годы. Конечно, есть масса толковых подростков, которые за неделю осваивают Delphi, еще за неделю Python, еще за неделю Perl, после чего они считают себя знатоками. При этом у них нет даже самого отдаленного представления о том, сколь многого они не знают.»

"… изучить 90% необходимого в работе материала можно за неделю. Но на освоение оставшихся 10% может уйти пара лет. Вот тут-то и проявляется во всем блеске превосходство действительно опытных программистов над теми, кто говорит, что если он чего-то не знает, то возьмет книжку и прочтет".

Дж. Спольски

Общеобразовательная литература

- Вирт Н. Алгоритмы + структуры данных = программы
- Кнут Д. Искусство программирования, том 1. Основные алгоритмы, 3-е изд. – М.: Издательский дом "Вильямс", 2000.
- Кнут Д. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. – М.: Издательский дом "Вильямс", 2000 (есть на моем сайте в разделе «Комбинаторные алгоритмы»).
- Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. – М.: МЦНМО, 1999 (есть на моем сайте в разделе «Комбинаторные алгоритмы»).
- Фаулер М. Рефакторинг: улучшение существующего кода. СПб: Символ-Плюс, 2007. 432 с.

Литература - моя

1. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник — СПб.: ПИТЕР, 2010. — 432 с.
2. Павловская Т.А. «Основы языка C#. Учебно-методическое пособие» — РИО СПбГУИТМО, 2008.
3. Павловская Т.А. «Программирование на языке C#. Методические рекомендации по лабораторному практикуму» — РИО СПбГУИТМО, 2008.
4. Павловская Т.А. «Программирование на языке C#. Методические рекомендации по выполнению самостоятельной работы студентов» — РИО СПбГУИТМО, 2008.

Литература

1. **Стандарт языка**: http: http://: http://www: http://www.: http://www.ecma: http://www.ecma-: http://www.ecma-international: http://www.ecma-international.: http://www.ecma-international.org: http://www.ecma-international.org/: http://www.ecma-international.org/publications: http://www.ecma-international.org/publications/: http://www.ecma-international.org/publications/standards: http://www.ecma-international.org/publications/standards/: http://www.ecma-international.org/publications/standards/Ecma: http://www.ecma-international.org/publications/standards/Ecma-334.: http://www.ecma-international.org/publications/standards/Ecma-334.htm
(версия 1.0 в удобном формате есть на моем сайте)
2. Хейлсберг А. и др. Язык программирования C#. — СПб.: Питер, 2012. — 784 с.
3. Троелсен Э. C# и платформа .NET. Библиотека программиста. — СПб.: Питер, 2008. — 796 с.
4. Мартин Р., Мартин М. Принципы, паттерны и методики гибкой разработки на языке C#. — М.: «Символ+», 2011. — 768 с.
5. Нэш Т. C# 2008: ускоренный курс для профессионалов. — М.:

Электронные источники

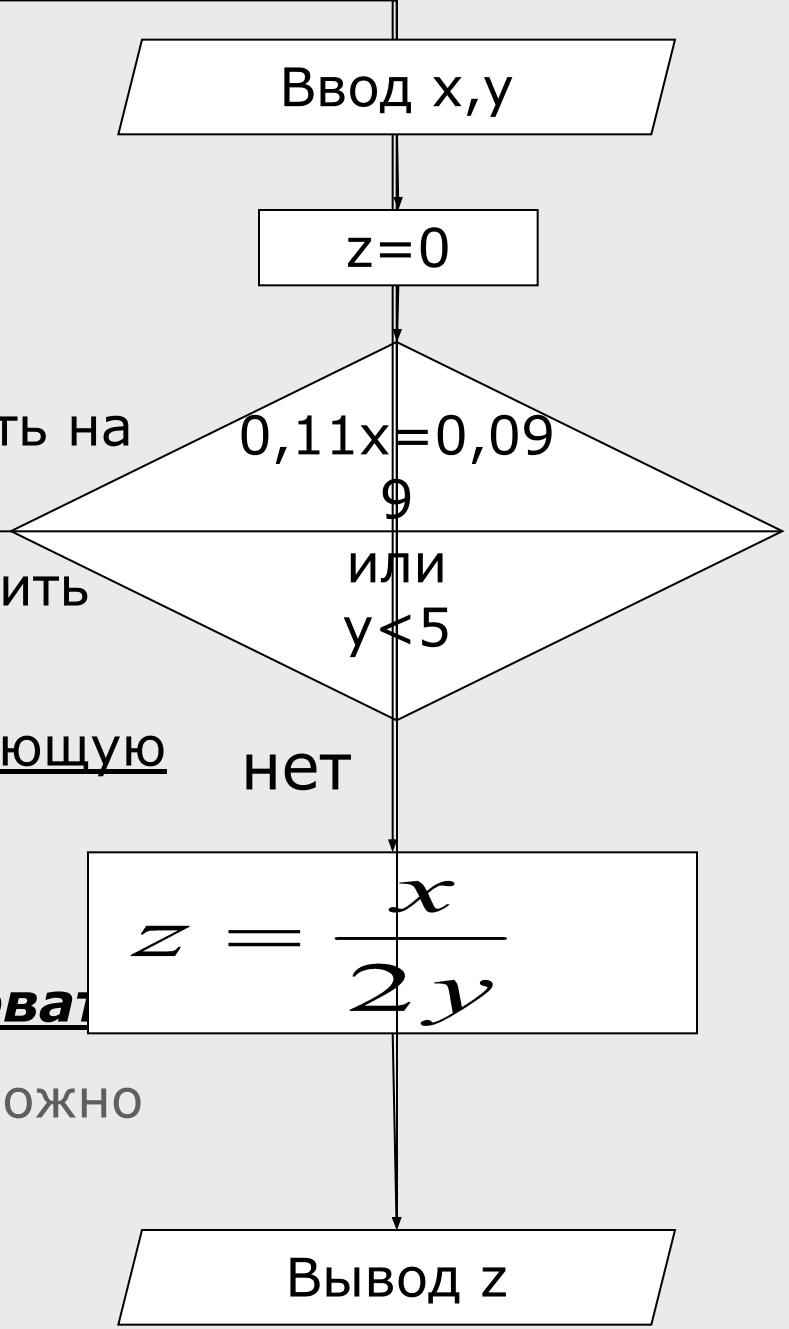
- **http**http**://**http://**pta**http://pta**-**http://pta-**ipm**http://pta-ipm**.**http://pta-ipm.narod**http://pta-ipm.narod**.**http://pta-ipm.narod.ru** — презентации лекций, текущие объявления
- **http**http**://**http://**de**http://de**.**http://de.ifmo**http://de.ifmo**.**http://de.ifmo.ru** — электронные тесты, программа курса
- **http://www.intuit.ru**/department/pl/phlcsharp/ — учебный курс (по моему учебнику)
- http**http**://http://ipshttp://ips.http://ips.ifmohttp://ips.ifmo.http://ips.ifmo.ru — конспекты лекций, автоматическая проверка программ
- http://msdn.microsoft.com/ru-ru/library/ — библиотека MSDN
- ...

И еще литература

- Эдвард де Боно. Почему мы такие тупые?. — СПб: ПИТЕР, 2009. — 128 с.

Анкета

1. Фамилия, имя, номер группы
2. Опыт программирования (языки, уровень, стаж, достижения)
3. Есть ли у Вас возможность работать на компьютере вне вуза?
4. Есть ли у Вас возможность приносить на занятия ноутбук?
5. Напишите программу, <u>соответствующую</u> приведенной схеме алгоритма, на любом языке программирования. *Все переменные – <u>вещественные</u>,* ***<u>блоки в программе не дублировать</u>***
6. Если у Вас есть ко мне вопросы, можно их задать, отвечу на следующей лекции.



Основные критерии качества программы

- надежность
- возможность точно планировать производство и сопровождение

Для достижения этих целей программа должна иметь простую структуру, чтобы

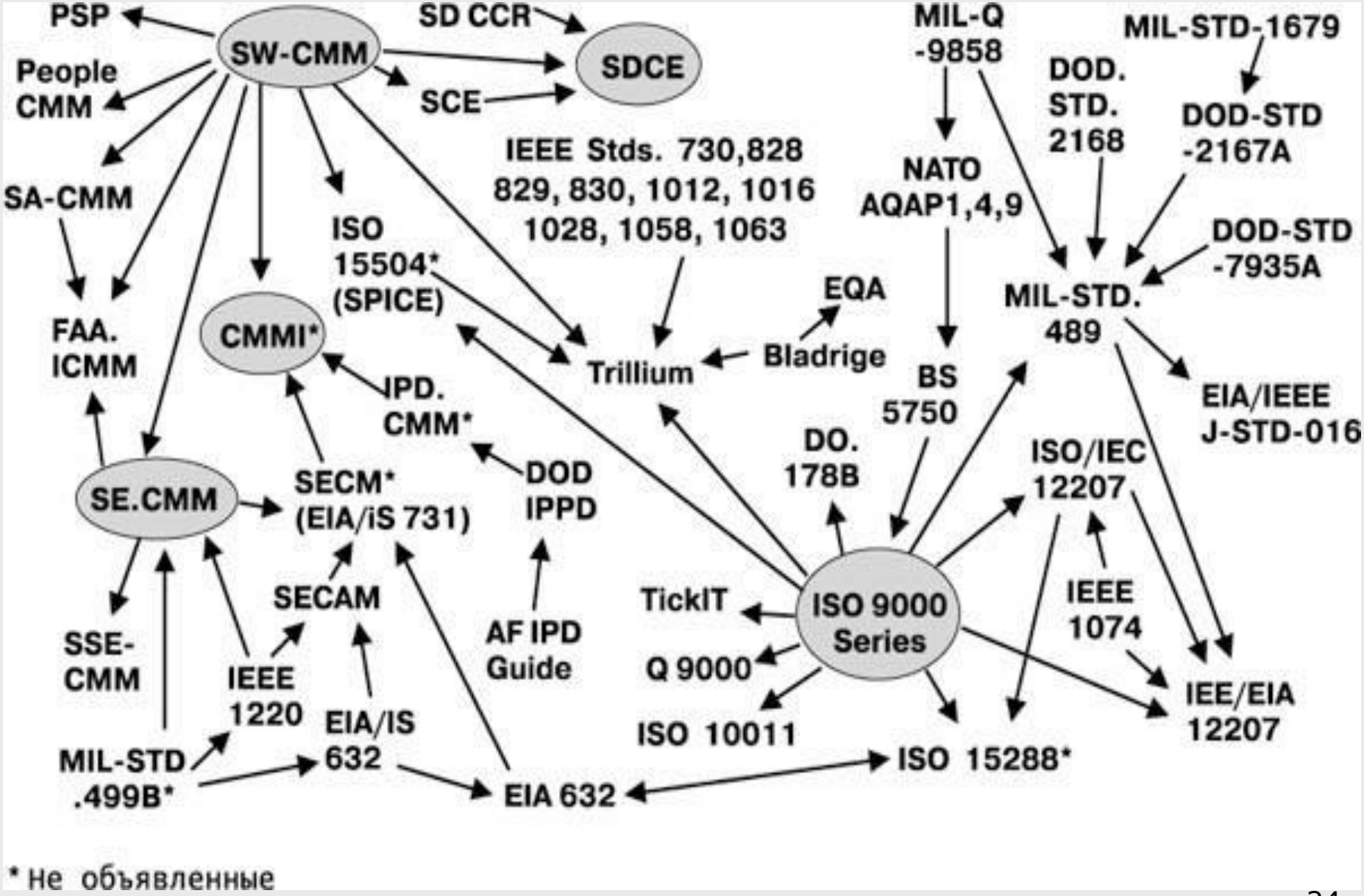
- быть читабельной
- быть легко модифицируемой

Характеристики качества ПО

Внешние характеристики

- корректность
  - наличие/отсутствие дефектов в спецификации, проекте и реализации
- практичность
  - легкость изучения и использования
- эффективность
  - степень использования системных ресурсов
- надежность
  - способность системы выполнять необходимые функции; интервал между отказами
- целостность
  - способность предотвращать неавторизованный или некорректный доступ
- адаптируемость
  - возможность использования в других областях и средах
- правильность
  - степень безошибочности данных, выдаваемых системой
- живучесть
  - способность продолжать работу при недопустимых данных или в напряженных условиях

Взаимосвязь наиболее признанных и применяемых в мире стандартов в области разработки программного обеспечения



Парадигмы программирования

Парадигма — способ организации программы, то есть принцип ее построения. Наиболее распространенными являются процедурная и объектно-ориентированная парадигмы.

Они различаются способом декомпозиции, положенным в основу при создании программы.

Процедурная декомпозиция состоит в том, что задача, реализуемая программой, делится на подзадачи, а они, в свою очередь — на более мелкие этапы, то есть выполняется пошаговая детализация алгоритма решения задачи.

Объектно-ориентированная декомпозиция предполагает разбиение предметной области на объекты и реализацию этих объектов и их взаимосвязей в виде программы.

Кроме того, существуют *функциональная* и *логическая* парадигмы.

Платформа .NET

- Средства разработки:
  - набор языков программирования
  - набор средств разработки, в т.ч. среда VisualStudio.NET;
  - **КАРКАС .NET FRAMEWORK:**
  - *библиотека классов*
  - *исполняющая система (виртуальная машина) CLR*
- Специализированные серверы:
  - механизмы поддержки баз данных, электронной почты и коммерции
- Веб-сервисы
- Устройства
  - устройства, поддерживающие .NET – игровые приставки и пр.

Среда разработки Visual Studio.NET

- текстовый *редактор с IntelliSense*;
- *компиляторы*;
- средства *отладки и запуска программ*;
- *справочная система;*
- редактор форм, веб-редактор, дизайнер классов, дизайнер схемы базы данных…

- Visual Studio Express – «облегченная версия» для студентов и любителей

Структура .NET

Платформа .NET

Среда разработки приложений **Visual Studio.NET**

Каркас **.NET Framework**

Библиотека классов

Общеязыковая среда выполнения (CLR)

.NET Enterprise Servers

.NET Building Block Services

Операционная система

Аппаратные средства компьютера

Первый взгляд на классы

- Понятие *класс* аналогично обыденному смыслу этого слова в контексте «класс членистоногих», «класс задач».
- Класс является обобщенным понятием, определяющим характеристики и поведение некоторого множества конкретных объектов этого класса, называемых *экземплярами класса (объектами)*.

- Класс содержит *данные*, задающие свойства объектов класса, и *функции (методы)*, определяющие их поведение.

- Все классы .NET имеют одного общего предка — класс object, и организованы в единую иерархическую структуру.
- Классы логически сгруппированы в так называемые *пространства имен*, которые служат для упорядочивания имен классов и предотвращения их конфликтов: в разных пространствах имена могут совпадать. Пространства имен могут быть вложенными.

Трансляция

Компиляция

Исходный текст программы → Компилятор, компоновщик → Программа на машинном языке → Выполнение программы

Интерпретация

Оператор на исходном языке → Интерпретатор → Команды на машинном языке → Выполнение команд

Гибридная схема трансляции

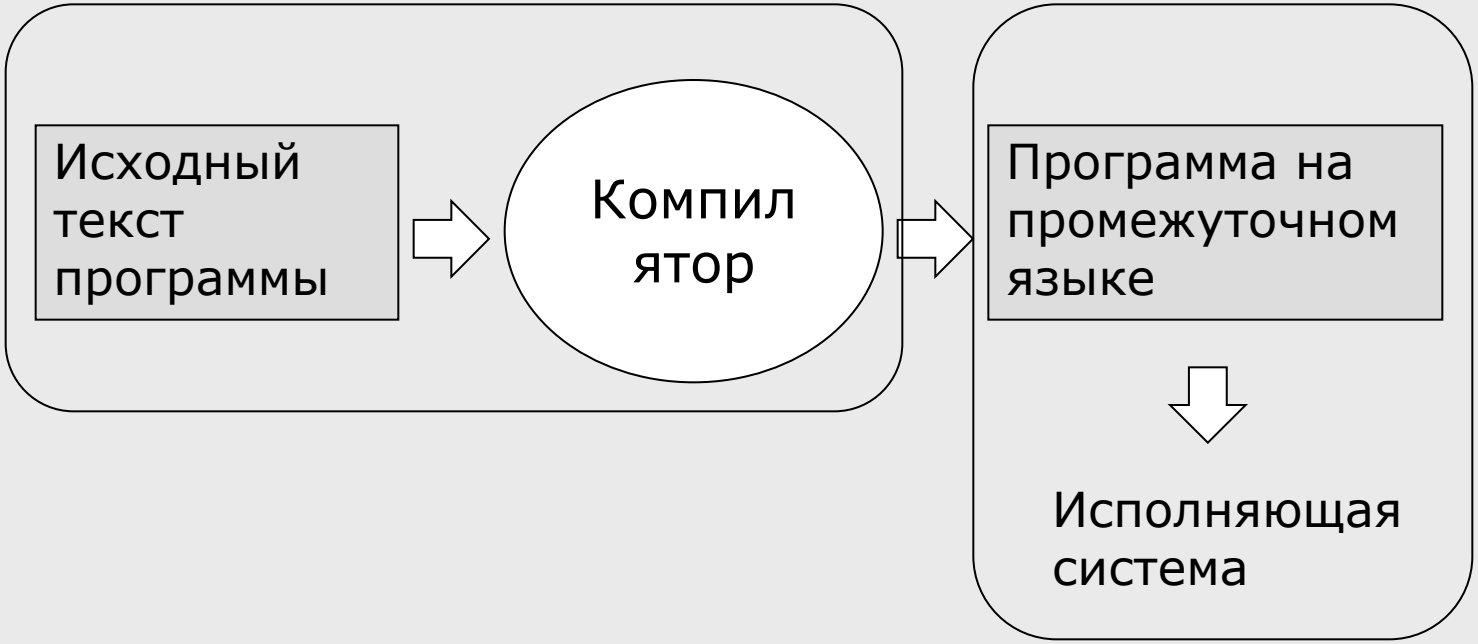
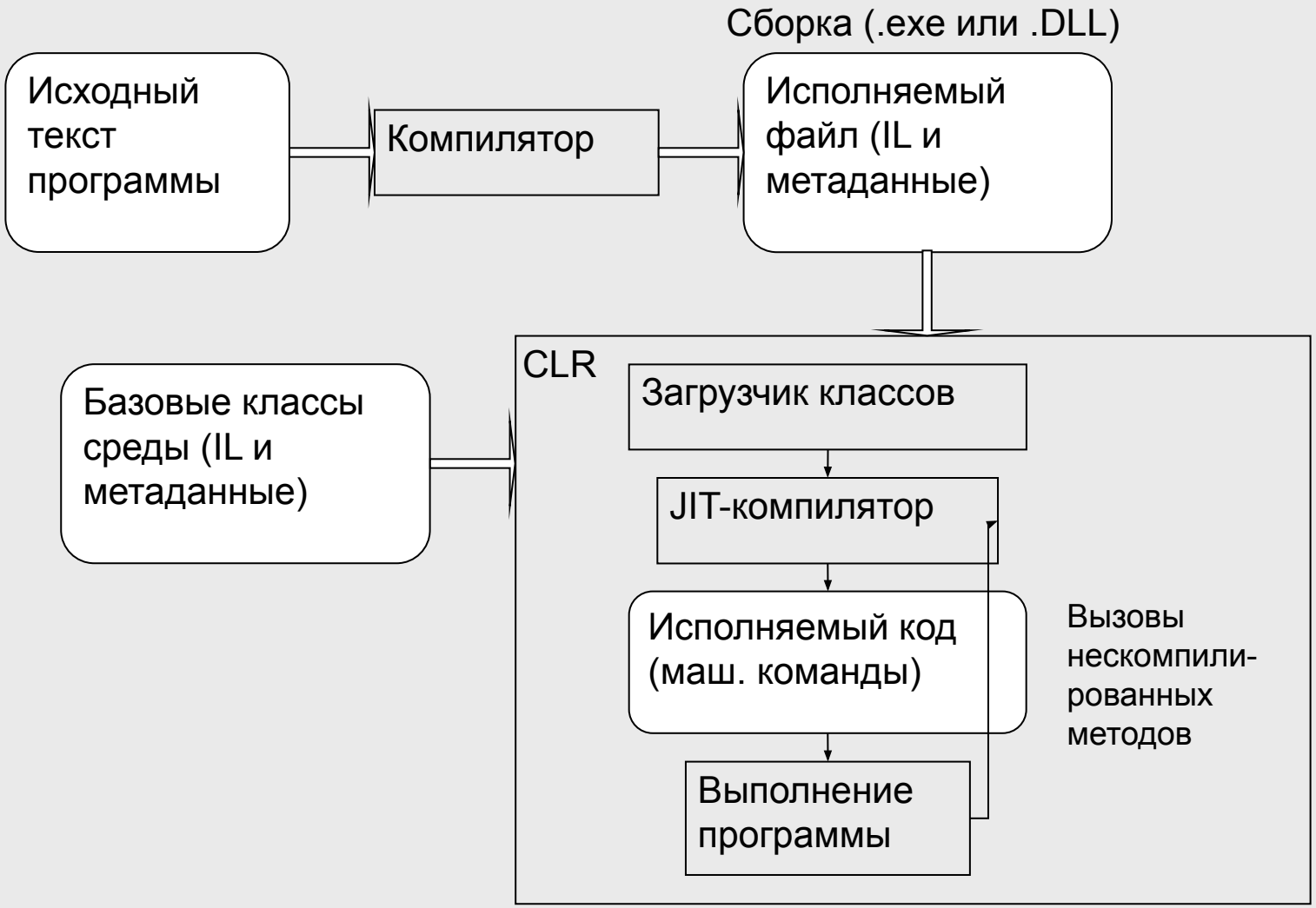
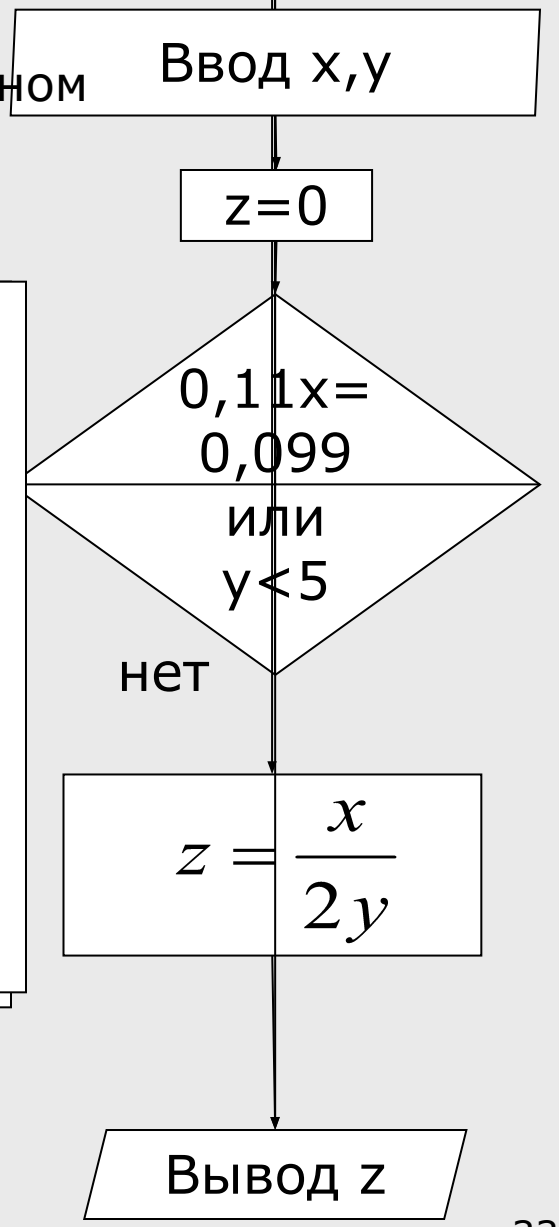


Схема выполнения программы в .NET



Решение

Напишите программу, **соответствующую приведенной схеме** алгоритма, на любом известном Вам языке программирования.
Все переменные – вещественные, блоки в программе не дублировать.

```
var x, y, z : real;                                  {4}
begin
    readln(x, y); z := 0;
    if ( (0.11*x <> 0.099) and (y>=5) )
            then z := x / ( 2 * y);
    writeln(z);
end.
```