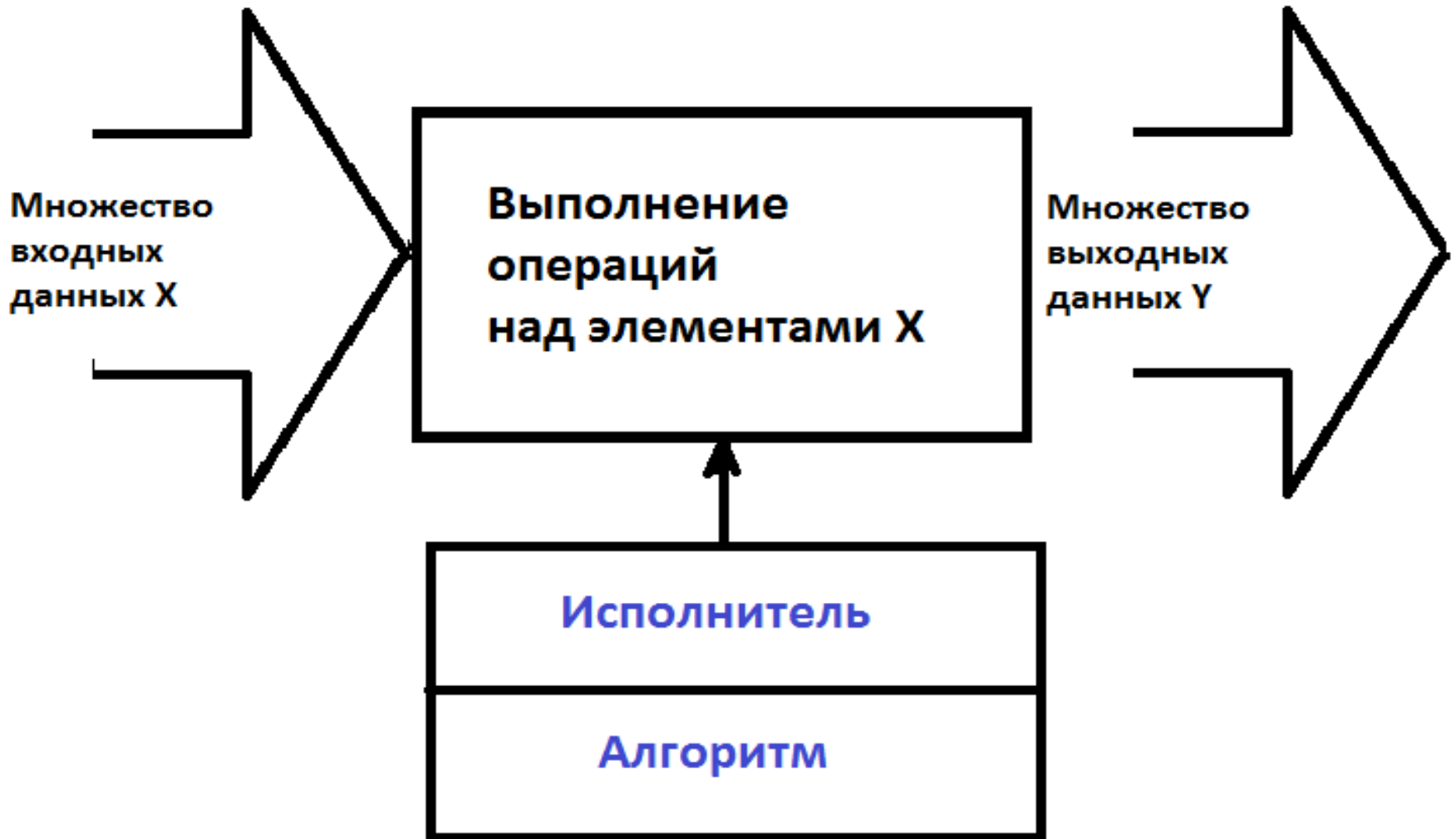


В.В. Подбельский

# Иллюстрации к курсу лекций по дисциплине «Программирование»

О языках программирования и  
дисциплине «Программирование»

# Обработка данных



# Парадигмы, выделенные для сравнения языков программирования

- Императивная
- Объектно-ориентированная
- Функциональная
- Рефлексивная
- Обобщенная
- Логическая
- Декларативная
- Распределенная

**“СРАВНЕНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ”:**

<https://ru.m.wikipedia.org/wiki/>

# Краткая Хронология Языков Программирования

## Процедурные языки

1954	1959	1960	1964	1964	1970	1972	1978-1983
<b>FORTRAN</b> Джон Бэкус	<b>COBOL</b> Грейс Хоппер	<b>ALGOL</b> IFIP	<b>BASIC</b> Томас Курц, Джон Кемени	<b>PL/1</b> IBM	<b>Pascal</b> Никлаус Вирт	<b>C</b> Денис Ритчи	<b>Ada</b> Honeywell

## Объектно-ориентированные языки

1967	1969	1980	1985
<b>Simula</b> Кристен Ньюгорд, Оле-Йохан Даль	<b>Smalltalk</b> Алан Кэй	<b>Smalltalk-80</b> Xerox PARK	<b>C++</b> Бьерн Страуструп

## «Мультипарадигменные» языки

1987	1990	1994	1994	1995	2001
<b>Perl</b> Ларри Уолл	<b>Python</b> Гвидо ван Россум	<b>PHP</b> Расмус Лердорф	<b>Java</b> Sun Microsystems	<b>Ruby</b> Юкихиро Мацумото	<b>C#</b> Андерс Хейлсберг

# Процедурные языки

- 1954 – FORTRAN (Джон Бэкус)
- 1959 – COBOL (Грейс Хоппер)
- 1960 – ALGOL (IFIP)
- 1963 – BASIC (Джон Кемени и Томас Куртц)
- 1964 – PL/1 (IBM)
- 1970 – Pascal (Никлаус Вирт)
- 1972 – C (Денис Ритчи)
- 1978-1983 – Ada (Honeywell)

# Объектно-ориентированные

- 1967 – Simula, Кристен Ньюгорд и Оле-Йохан Даль
- 1969 – Smalltalk, Алан Кэй;
- 1980 - Smalltalk-80, Xerox PARC
- 1985 – C++, Бьерн Страуструп

# Декларативные языки

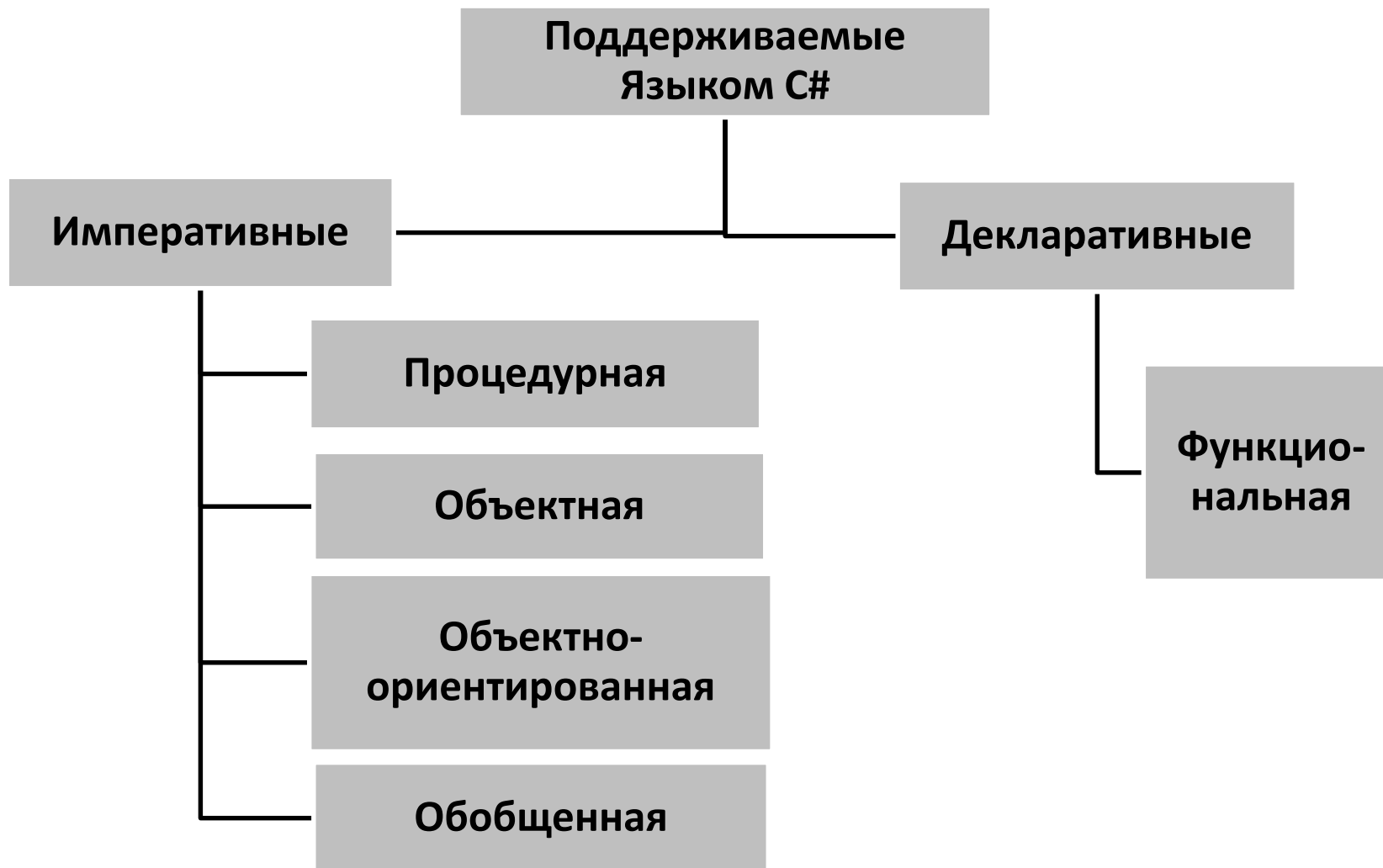
- 1958 – Лисп (LISP), Джон Маккарти (язык обработки списков)
- 1972 - Prolog (Пролог – язык логического программирования)
- 1974 - SQL, Язык управления реляционными базами данных (Oracle).
- 1990 - Haskell – (функциональный язык, назван в честь математика Хаскелла Карри)
- 2008 - F#, Дон Сайм (Microsoft Research)

# Мультипарадигменные языки

- 1987 – Perl, Ларри Уолл
- 1990 – Python, Гвидо ван Россум
- 1994 – PHP, Рasmus Лерддорф
- 1994 – Java, Sun Microsystems
- 1995 – Ruby, Юкихиро Мацумото
- 2001 – C#, Microsoft, Андерс Хейлсберг



# Парадигмы Программирования C#



# Рубрики классификации языков

- Семантика
- Класс языка
- Появился
- Автор
- Выпуск
- Система типов
- Основные реализации
- Диалекты
- Испытал влияние
- Повлиял
- Стандартизован

# Признаки классификации языков программирования

- Парадигмы
- Типизация
- Компилятор / интерпретатор
- Управление памятью
- Управление потоком вычислений
- Типы и структуры данных
- Объектно-ориентированные возможности
- Функциональные возможности
- Разное
- Стандартизация

**“СРАВНЕНИЕ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ”:**  
<https://ru.m.wikipedia.org/wiki/>

# Условные обозначения, используемые при классификации

<b>( + )</b>	<b>указанная возможность присутствует</b>
<b>( - )</b>	<b>указанная возможность отсутствует</b>
<b>( + / - )</b>	<b>возможность поддерживается не полностью</b>
<b>( - / + )</b>	<b>возможность поддерживается очень ограничено</b>

# Наличие парадигм для C#

- Императивная ( + )
- Объектно-ориентированная ( + )
- Рефлексивная ( -/+ )
- Функциональная ( +/- )
- Обобщенная ( + )
- Логическая ( - )
- Доказательная ( - )
- Декларативная ( -/+ )
- Распределенная ( -/+ )

# Типизация

Статическая типизация	(+)	Вывод типов переменных из инициализатора	(+)
Динамическая типизация	(+)	Вывод типов переменных из использования	(-)
Явная типизация	(+)	Вывод типов-аргументов при вызове метода	(+)
Неявная типизация	(-/ +)	Вывод сигнатуры для локальных функций	(-)
Явное приведение типов	(+)	Параметрический полиморфизм	(+)
Неявное приведение типов без потери (точности)	(+)	Параметрический полиморфизм с ковариантностью	(+/-)
Неявное приведение типов с потерей данных	(-)	Параметрический полиморфизм высших порядков	(-)
Неявное приведение типов в неоднозначных ситуациях	(+)	Информация о типах в runtime	(+)
Алиасы типов	(+)	Информация о типах-параметрах в runtime	(+)

# Возможности C#

1. Настольные приложения
2. Приложения для Windows-магазина (плюс **Xamarin** – разработка для iOS, Android, Win.Phone)
3. Web-приложения
4. Сервисы (WCF) и распределенные приложения

**Механизм ADO .NET (ActiveX Data Objects .NET) для работы с базами данных.**

- ADO.NET Entity Framework
- LINQ (Language Integrated Query)

# Microsoft Imagine (ex. DreamSpark)

1. Для получения доступа к Imagine (MSDN) Вам необходимо заполнить «[Заявку на подключение к DreamSpark \(MSDN\)](#)».
2. Заявки от студентов принимаются по адресу [msdn.reg@hse.ru](mailto:msdn.reg@hse.ru) со студенческой почты @edu.hse.ru.

**Полная информация о подключении к Microsoft Imagine доступна на странице:**  
**<https://it.hse.ru/dream>**



# План преподавания дисциплины

Распределение аудиторных часов по модулям:

**Модуль 1:** 42 часа, в неделю: лек. 2, практ. 1.

Экзамен.

**Модуль 2:** 42 часа, в неделю: лек. 1, практ. 2.

Контрольная; КДЗ. Экзамен

**Модуль 3:** 44 часа, в неделю: лек. 1, практ. 1.

Контрольная; КДЗ.

**Модуль 4:** 40 часов, в неделю: лек. 1, практ. 1.

Контрольная. Экзамен

**Итого:** 168 часов аудиторных занятий.

**Самостоятельная работа: 212 часов**

# Порядок формирования оценок

## Условные обозначения:

- **НО<i>** – накопленная оценка i-го модуля;
- **ТП<i>** – (сумма баллов по проверкам i-го модуля) / (количество проверок в i-м модуле); текущая проверка может быть компьютерным тестом (до 10 вопросов) или самостоятельной работой по решению задач на компьютере;
- **СР<i>\_<j>** – оценка самостоятельной работы j в i-м модуле;
- **КР<i>** – оценка контрольной работы i-го модуля;
- **КДЗ<i>** – оценка контрольного домашнего задания i-го модуля;
- **ЭК<i>** – оценка за экзаменационную контрольную работу (первая часть экзамена) i-го модуля;
- **ЭТ<i>** – оценка за экзаменационный тест (вторая часть экзамена) i-го модуля;
- **ЭО<i>** – экзаменационная оценка (является блокирующей) i-го модуля (задача и/или тест);
- **ИО<i>** – итоговая оценка i-го модуля;

# Порядок формирования оценок 1-го модуля

**ИО1** = Округление( $0,6 * ЭО1 + 0,4 * НО1$ ).

**НО1** = Округление( $CP1\_1 / 3 + CP1\_2 / 3 + CP1\_3 / 3$ ).

Если экзамен состоит из двух частей (контрольная и тест):  
 $ЭО1 = \text{Округление}(0,5 * ЭК1 + 0,5 * ЭТ1)$ , если  $\min(ЭК1; ЭТ1) > 3$ ;  
иначе  $ЭО1 = \min(ЭК1; ЭТ1)$ .

# Порядок формирования оценок 2-го модуля

**ИО2** = Округление( $0,6 * ЭО2 + 0,4 * НО2$ ).

**НО2** = Округление( $0,3 * (CP2\_1 / 3 + CP2\_2 / 3 + CP2\_3 / 3) + 0,5 * КР2 + 0,2 * КДЗ$ ).

Если экзамен состоит из двух частей (контрольная и тест):  
 $ЭО2 = \text{Округление}(0,5 * ЭК2 + 0,5 * ЭТ2)$ , если  $\min(ЭК2; ЭТ2) > 3$ ;  
иначе  $ЭО2 = \min(ЭК2; ЭТ2)$ .

Обозначения:

$ЭК<i>$  - экзаменационная контрольная  $<i>$ -го модуля;

$ЭТ<i>$  - экзаменационный тест  $<i>$ -го модуля.

# Порядок формирования оценок 4-го модуля

**ИО4** = Округление( $0,6 * ЭО4 + 0,1 * ЭО2 + 0,3 * НО4$ ).

**НО4** = Округление( $0,1 * (CP3\_1 / 4 + CP3\_2 / 4 + CP3\_3 / 4 + CP3\_4 / 4) + 0,2 * КР3 + 0,15 * (CP4\_1 / 3 + CP4\_2 / 3 + CP4\_3 / 3) + 0,25 * КР4 + 0,3 * КД33$ ).

Если экзамен состоит из двух частей (контрольная и тест):

$ЭО4 = \text{Округление}(0,5 * ЭК4 + 0,5 * ЭТ4)$ , если  $\min(ЭК4; ЭТ4) > 3$ ;

иначе  $ЭО4 = \min(ЭК4; ЭТ4)$ .

Обозначения:

ЭК4 - экзаменационная контрольная четвертого модуля;

ЭТ4 - экзаменационный тест четвертого модуля.

# Критерии выставления оценки за программу

## “НЕУДОВЛЕТВОРИТЕЛЬНО”:

### 1 балл:

1. Разработка программы не завершена.
2. Программа имеет синтаксические ошибки (не компилируется).

### 2 балла:

1. Программа не решает основную задачу или не соответствует спецификации.
2. В программе обнаруживаются не обработанные исключения при решении основных и второстепенных подзадач.

### 3 балла:

1. Программа не решает основную задачу при некоторых вариантах исходных данных.
2. Программа завершается аварийно при некоторых вариантах исходных данных.

# Критерии выставления оценки за программу

## “УДОВЛЕТВОРИТЕЛЬНО”:

### 4 балла:

1. Программа решает основную задачу, но имеет отклонения от спецификации.

### 5 баллов:

1. Программа соответствует критериям получения оценки 4 балла.
2. Программа соответствует отдельным дополнительным критериям.

## “ХОРОШО”:

### 6 баллов:

1. Программа решает поставленную задачу и соответствует спецификации. Отклонения от спецификации допущены при реализации второстепенных подзадач.
2. Исходный текст документирован.

### 7 баллов:

1. Программа соответствует критериям получения оценки 6 баллов.
2. Программа в целом соответствует дополнительным критериям.

# Критерии выставления оценки за программу

## “ОТЛИЧНО”:

### 8 баллов:

1. Программа решает все поставленные задачи и полностью соответствует спецификации.
2. Студент в комментариях обосновал принятые конструктивные решения.
3. Исходный текст документирован. Присутствуют сведения о назначении используемых переменных, параметров, методов, классов, объектов.
4. Программа остается работоспособной при вводе неверных исходных данных.
5. Предусмотрено повторное решение задачи без повторного запуска программы.

# Критерии выставления оценки за программу

## “ОТЛИЧНО”:

### 9 баллов:

1. Программа соответствует критериям получения оценки 8 баллов.
2. Программа соответствует некоторым дополнительным критериям.

### 10 баллов:

1. Программа соответствует критериям получения оценки 8 баллов.
2. Программа полностью соответствует всем дополнительным критериям.
3. Студент отразил в комментариях возможность альтернативных вариантов решения задачи.



# Дополнительные критерии оценивания

- 1. Исходный текст программы структурирован.
- 2. Программа имеет средства изменения размерности и формы представления данных без изменения исходного текста или при минимальных изменениях (динамическое выделение памяти, именованные константы, и т.п.).
- 3. Программа реализована по модульному принципу и соответствующим образом декомпозирована.
- 4. В коде программы обоснованно использованы конструкции, изученные студентом самостоятельно, и их применение не противоречит основным требованиям, предъявляемым к решаемой задаче.
- 5. Предусмотрено самодокументирование программы.

# План проверочных работ (модуль 1)

- **23-28 сентября** Самостоятельная работа 1
- **30 сентября - 5 октября** Самостоятельная работа 2
- **14-19 октября** Самостоятельная работа 3
- **21-27 октября** Экзамен

# Где искать материалы?

## Материалы занятий (OneDrive)

— <http://bit.do/progCS2019>

# Работа в компьютерном классе

- В начале сеанса работы в классе студент должен на диске **D** создать папку и назвать ее своей фамилией.
- В папке следует сохранять результаты работы на практическом занятии.



***Внимание: После выключения компьютера, все папки с рабочего стола и диска C:\ автоматически удаляются!***