



ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
УПРАВЛЕНИЕ ДИСТАНЦИОННОГО ОБУЧЕНИЯ И ПОВЫШЕНИЯ
КВАЛИФИКАЦИИ

Кафедра «Программное обеспечение вычислительной тех-
ники и автоматизированных систем»

Учебно-методическое пособие по дисциплине

«Языки программирования высокого уровня»

Автор
Долгов В.В.

Ростов-на-Дону, 2018

Аннотация

Учебно-методическое пособие предназначено для студентов очной формы обучения направления 09.04.04 «Программная инженерия».

Авторы

доцент, к.т.н.,
зав.каф. ПОВТиАС
Долгов В.В.



Оглавление

1. Общие указания к выполнению лабораторных работ	4
1.1. Требование к лабораторному оборудованию	4
1.2. Требования, предъявляемые при сдаче лабораторных работ	4
2. Лабораторная работа №1: Создание функциональных программ на языке F#	5
2.1. Цель работы	5
2.2. Задание к лабораторной работе	5
2.3. Контрольные вопросы	6
3. Лабораторная работа №2: Классы в языке F#	7
3.1. Цель работы	7
3.2. Задание к лабораторной работе	7
3.3. Контрольные вопросы	10
4. Лабораторная работа №3: Разработка параллельных программ на языке F#	11
4.1. Цель работы	11
4.2. Задание к лабораторной работе	11
4.3. Контрольные вопросы	12
5. Лабораторная работа №4: Математическая обработка больших массивов данных	12
5.1. Цель работы	12
5.2. Задание к лабораторной работе	13
5.3. Контрольные вопросы	14
Список литературы	15

1. Общие указания к выполнению лабораторных работ

1.1. Требование к лабораторному оборудованию

Аудитория для проведения лабораторных/практических занятий должна быть укомплектована специализированной мебелью и компьютерами следующей минимальной комплектации:

- Процессор: не менее двух исполнительных ядер, совместимый с системой команд x86 и x64, с поддержкой аппаратной виртуализации.

- Оперативная память: не менее 4 Гб.

- Монитор: не менее 22" (дюймов) по диагонали.

- Наличие сети не менее 100 Мб/сек.

- Наличие клавиатуры и манипулятора «мышь».

На компьютерах должно быть установлено следующее программное обеспечение:

- Операционная система: Microsoft Windows 7 (или выше).

- Среда программирования: Microsoft Visual Studio 2012 (или выше).

- Компьютер должен быть обеспечен доступом в глобальную сеть Интернет на скорости не менее 1 Мбит/сек.

1.2. Требования, предъявляемые при сдаче лабораторных работ

При сдаче студентом лабораторной работы отчетом выступает исходный код созданной в процессе выполнения работы программы.

Исходный код должен быть отформатирован согласно принятым для языка F# стандартам. Допускается наличие в исходном коде комментариев, описывающих основные части программы и особенности их функционирования. В то же время студент должен быть готов объяснить работу программы в целом и каждую отдельную ее часть при полном отсутствии комментариев (например, они могут быть удалены или сделаны нечитаемыми перед сдачей).

Обязательным условием сдачи является умение студента восстановить любой участок исходного кода программы после его удаления (при этом этот участок должен быть набран студентом заново, запрещается использование операции Undo текстовых редакторов).

Исходный код сдаваемой программы должен быть представлен в электронном виде.

Сдача исходного кода ранее сдававшихся программ или программ, код которых выложен в сети Интернет, не допускается.

2. ЛАБОРАТОРНАЯ РАБОТА №1: СОЗДАНИЕ ФУНКЦИОНАЛЬНЫХ ПРОГРАММ НА ЯЗЫКЕ F#

2.1. Цель работы

Получить практические навыки по созданию программ в функциональном стиле на языке программирования F# с использованием среды разработки Visual Studio 2012 или выше. Закрепить знания по использованию основных конструкций языка и навыков по использованию хвостовой рекурсии как замены циклам в императивных языках.

2.2. Задание к лабораторной работе

Каждый вариант задания содержит две задачи, которые должны быть выполнены без использования циклов. При написании программ разрешается объявлять функции с любым количеством аргументов, использовать вложенные функции и частичное применение.

Тексты заданий по вариантам приведены в таблице 2.1.

Таблица 2.1 – Варианты заданий к лабораторной работе

№1

№ варианта	Задание
1	<p>А) Напишите программу, вводящую с клавиатуры три вещественных числа и выводящую на консоль наименьшее из них.</p> <p>Б) Создайте функцию, печатающую таблицу значений другой функции от одного вещественного аргумента. Создаваемая функция должна принимать табулируемую функцию, начальное и конечное значение интервала и шаг табуляции.</p>
2	<p>А) Опишите функции, осуществляющие основные математические операции над комплексными числами. Использовать готовый класс Complex запрещено. Проверьте правильность работы созданных функций.</p>

	<p>Б) Создайте функции для выполнения основных операций над матрицами. В качестве структуры данных для хранения матриц разрешается использовать двумерные массивы, массивы массивов или списки (в различной конфигурации на выбор обучающегося). Напишите простое приложение для проверки правильности работы созданных функций.</p>
3	<p>А) Написать функцию, которая:</p> <ul style="list-style-type: none"> – возвращает квадратный корень из аргумента, если аргумент является числом; – последний элемент аргумента-списка, если аргумент – список; – сам аргумент во всех остальных случаях.
	<p>Б) Создайте собственные функции для выполнения основных операций над списками (добавление/удаление/поиск элемента, сцепка двух списков, получение элемента по номеру).</p>
4	<p>А) Написать функцию, которая по заданному вещественному числу формирует список из трех элементов. Первый элемент – знак числа, второй – модуль числа, третий – ближайшее к нему целое число.</p>
	<p>Б) Написать программу, которая в исходном списке заменяет все целые числа остатками от их деления на 2. Список может содержать подсписки произвольной глубины вложения.</p>
5	<p>А) Даны четыре числа. Если все числа положительные, то сформировать список из минимального и максимального. В противном случае вернуть пустой список.</p>
	<p>Б) Описать функцию сортировки массива. При написании программы можно использовать любой алгоритм сортировки, но нельзя использовать стандартные функции сортировки либо функции нахождения минимумов/максимумов.</p>

2.3. Контрольные вопросы

1. Существует ли в языке F# оператор, заменяющий или являющийся аналогом тернарного оператора «?:» в таких языках как C/C++? Почему? Приведите примеры.

2. В чем основное отличие хвостовой рекурсии от обычной? Какое условие должно выполняться, чтобы рекурсия могла счи-

таться хвостовой? Какие преимущества это дает?

3. Можно ли используя оператор сравнения с образцом отказаться от условного оператора «if»? Если да, приведите пример такой замены.

4. Приведите пример описания и использования вложенной функции.

5. Что такое анонимная функция? Приведите пример использования анонимной функции при обработке массива.

6. Какие ограничения накладываются на изменяемые (mutable) значения? Как способы предлагает язык для обхода этих ограничений?

7. Назовите основные операции, доступные программисту для работы со списками.

8. Что такое функции высшего порядка (ФВП)? Приведите пример описания и использования такой функции. В чем преимущество применения ФВП?

9. Что такое и для чего нужно связанное описание типов данных? Приведите пример, когда создать тип без использования связанного описания нельзя.

10. Что такое опциональный тип данных? Приведите пример использования опционального типа.

3. ЛАБОРАТОРНАЯ РАБОТА №2: КЛАССЫ В ЯЗЫКЕ F#

3.1. Цель работы

Научиться разрабатывать и описывать с использованием языка F# программные классы, реализовывать интерфейсы, использовать явный и неявный способы задания конструкторов классов. Закрепить навыки создания гибких алгоритмов с использованием виртуальных функций.

3.2. Задание к лабораторной работе

Описать множество классов согласно заданию (по вариантам). Разработанные классы в обязательном порядке должны содержать не менее одного виртуального метода, поведение которого должно быть перекрыто в классах-наследниках, и хотя бы один перегруженный (overloaded) метод.

Во всех вариантах заданий необходимо определить класс Person, содержащий:

– фамилию, имя и отчество человека с возможностью их изменения;

Языки программирования высокого уровня

- дату рождения с возможностью ее изменения;
- возраст человека без возможности его изменения прямым образом;
- перегруженную версию виртуального метода ToString, формирующую текстовую строку с описанием человека;
- виртуальный метод ToShortString, формирующий строку с краткой информацией о человеке.

Создать простое консольное приложение, позволяющее проверить правильность функционирования созданных классов.

Тексты заданий по вариантам приведены в таблице 3.1.

Таблица 3.1 – Варианты заданий к лабораторной работе №2

№ варианта	Задание
1	<p>Определить класс Exam содержащий наименование предмета, оценка за предмет, дата сдачи предмета и перегруженный метод ToString, возвращающий описание экзамена.</p> <p>Определить класс Student, являющийся наследником Person и содержащий:</p> <ul style="list-style-type: none"> – свойство Education, определяющее уровень образования; – свойство с именем группы обучения; – массив с элементами типа Exam, содержащий список экзаменов, сданных студентом; – вычисляемое поле «Средний балл», содержащее среднеарифметическое оценок, полученных на экзаменах; – метод AddExam, добавляющий экзамен в перечень сданных; – перегруженную версию метода ToString, формирующую строку описания студента включая список экзаменов; – перегруженный метод ToShortString формирующий краткую строку описания без экзаменов, но со средним общим баллом.

<p>2</p>	<p>Определить класс Article имеющий:</p> <ul style="list-style-type: none"> – свойство Author типа Person, хранящее ссылку на автора статьи; – свойство типа string с названием строки; – свойство типа double с рейтингом статьи; – перегруженную версию метода ToString, возвращающую строку с описанием статьи. <p>Определить класс Edition (издание) со следующими элементами:</p> <ul style="list-style-type: none"> – свойство типа string с наименованием издания; – свойство типа DateTime с датой выхода издания; – свойство типа int с тиражом издания (значение свойства не должно принимать отрицательные значения). <p>Определить класс Magazine как наследника класса Edition, со следующими элементами:</p> <ul style="list-style-type: none"> – свойство типа string с наименованием журнала; – свойство Frequency, хранящее информацию о периодичности выхода журнала; – список элементов типа Article (список статей журнала); – свойство типа double (только для чтения) со средним баллом статей в журнале; – метод AddArticle, добавляющий новую статью в журнал; – перегруженную версию метода ToString для формирования строки с описанием журнала (должна включать краткую информацию обо всех статьях); – виртуальный метод ToShortString, формирующий краткую строку описания журнала (без списка статей, но со средним баллом по статьям).
----------	---

3	<p>Определить класс Paper (научная статья), содержащий следующие методы и свойства:</p> <ul style="list-style-type: none"> – строковое свойство с наименованием публикации; – свойство типа Person, содержащее ссылку на автора публикации; – свойство с датой публикации; – перегруженный метод ToString возвращающий строку с текстовым описанием статьи. <p>Определить класс Team (команда), содержащий:</p> <ul style="list-style-type: none"> – строковое свойство с наименованием организации; – поле типа int с регистрационным номером (номер не может быть меньше нуля); – свойство-массив со списком людей (Person), работающих в команде; – метод AddPerson, добавляющий человека в команду; – перегруженный метод ToString, возвращающий строку описание. <p>Определить класс ResearchTeam как наследника Team, содержащий:</p> <ul style="list-style-type: none"> – свойство и наименованием темы исследования; – свойство со списком публикаций по теме исследования типа Paper[]; – свойство Paper, возвращающее ссылку на публикацию с самой поздней версией исследования, если таковая есть (иначе null); – метод AddPapers добавляющий перечень публикаций к списку; – перегруженную версию метода ToString, возвращающего описание исследования с перечнем публикаций.
---	---

3.3. Контрольные вопросы

1. В чем отличие и сходство методов явного и неявного описания конструкторов?
2. В чем отличие описания класса от интерфейса в языке F#?

3. В чем заключается основное преимущество объектно-ориентированного программирования?

4. Что такое хэш-значение объекта? Какая функция возвращает это значение? Каким свойством оно должно обладать?

5. Что такое самоидентификатор? Как его объявить (приведите пример) и какие ограничения него налагаются?

6. Что такое наследование, и каким образом оно осуществляется в языке F#? В чем отличие синтаксисов наследования при явном и неявном описании?

7. Что такое виртуальные методы? Каким образом происходит описание и реализация виртуальных методов в F#?

4. ЛАБОРАТОРНАЯ РАБОТА №3: РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ ПРОГРАММ НА ЯЗЫКЕ F#

4.1. Цель работы

Закрепить теоретические знания и отработать навыки создания параллельных вычислений с использованием языка программирования F#.

4.2. Задание к лабораторной работе

С использованием одного из подходов создания параллельных программ реализовать на языке F# задание согласно варианту задания из таблицы 4.1. При разработке программ запрещено использовать математическую библиотеку языка.

Таблица 4.1 – Варианты заданий к лабораторной работе №3

№ варианта	Задание
1	Реализовать программу параллельно решающую систему линейных алгебраических уравнений методом итераций. Значения коэффициентов матриц A и B считывать из внешних файлов, получаемых программой в качестве параметров.
2	Реализовать программу параллельного умножения двух матриц. Элементы исходных матриц считывать из внешних файлов, имена которых указаны в первом и втором параметре программы. Результаты вычислений сохранить в файле, имя которого задано третьим параметром.

3	Реализуйте функцию параллельного вычисления интеграла сложной функции методом трапеции. Параметры интегрирования и подинтегральная функция передаются в качестве параметра.
4	Напишите программу для оптимизации функции генетическим алгоритмом с параллельной оптимизацией. Функции скрещивания, оценки, мутации, а также сама оптимизируемая функция должны передаваться параллельному алгоритму в качестве параметров.

4.3. Контрольные вопросы

1. Опишите известные вам способы выполнения параллельных действий. В чем их сходство и различия?
2. Перечислите причины, требующие синхронизации параллельных действий.
3. В чем основное преимущество пулов потоков над ручным созданием одиночных потоков?
4. Что такое «гонка за ресурсами»? Каковы причины ее возникновения? Как избежать гонки?
5. Что такое критическая секция? Какие негативные последствия ее использования вы знаете?
6. Какую роль в асинхронном программировании играет интерфейс `IAsyncResult`? Приведите пример.
7. Что такое `Async Workflow`? Какие преимущества он дает по сравнению с асинхронными делегатами? Приведите пример.
8. Какие примитивы распараллеливания программ с использованием `Parallel Extension` вы знаете? Приведите примеры их использования.

5. ЛАБОРАТОРНАЯ РАБОТА №4: МАТЕМАТИЧЕСКАЯ ОБРАБОТКА БОЛЬШИХ МАССИВОВ ДАННЫХ

5.1. Цель работы

Освоить конвейерную обработку данных, закрепить знания, полученные в ходе изучения языка программирования `F#`, получить практические навыки по совместному использованию и интеграции различных методик и подходов, доступных программисту в языке.

5.2. Задание к лабораторной работе

Используя любые изученные в ходе курса возможности, подходы и библиотеки языка, а также знания, полученные в ходе самостоятельной работы, создать программу согласно варианту задания (табл. 5.1).

Во всех вариантах задания при написании программы считать, что данные получаются из внешнего источника. Кроме того, учесть, что объем обрабатываемых данных может превышать объем доступной оперативной памяти компьютера.

Таблица 5.1 – Варианты заданий к лабораторной работе №4

№ варианта	Задание
1	Имея на входе несколько текстовых журналов веб-сервера в формате W3C, разработать программу вычисляющую статистику обращений к серверу и сохраняющую ее в файл данных для последующего использования. Анализ должен включать как минимум следующие данные: <ul style="list-style-type: none"> – графики обращений к серверу в разрезе доменов, пользователей, дат/часов/минут, подсетей пользователей, типу контента; – данные по скользящему среднему количеству обращений в единицу времени с возможностью фильтрации по подсетям клиентов.
2	Используя внешнюю обезличенную базу данных (в любом формате) оценок студентов высшего учебного заведения по различным предметам и годам, рассчитать (и расчеты сохранить): <ul style="list-style-type: none"> – степень сложности сдачи предметов в разрезе групп и годов обучения; – скользящую среднюю оценку студентов в разрезе предметов, факультетов, групп, годов обучения; – количество студентов, обучающихся на «хорошо» и «отлично», в разрезе специальностей/направлений, факультетов, групп и годов обучения.

3	Используя данные об электропотреблении строений в городе, рассчитать среднее, максимальное и минимальное потребление в разрезе дат, часов, районов, улиц, типов строений. Предусмотреть возможность установления фильтров по указанным параметрам.
---	--

5.3. Контрольные вопросы

1. В чем отличие рефлексии значений от рефлексии типов данных?
2. Приведите пример использования комплексных чисел в языке F#.
3. Для чего предназначен интерфейс IDisposable? Какие способы работы с интерфейсом вы знаете?
4. В чем преимущество событий как первоклассных объектов языка? Приведите примеры их использования.
5. Что такое расщепление события? Приведите примеры такого расщепления.
6. Какие способы оптимизации скорости работы кода вы можете назвать?
7. Назовите коллекции данных, которые можно использовать в языке программирования F#.

СПИСОК ЛИТЕРАТУРЫ

1. Сошников Д.В. Функциональное программирование на F#. – ДМК Пресс, 2011.
2. Robert Pickering, Foundations of F#. – Apress, 2007.
3. Chris Smith, Programming F#. – O'Reilly, 2010.
4. Березина Н.И. Лабораторные работы по курсу «Объектно-ориентированное программирование: язык программирования C#» : Учебное пособие – М.: Издательский отдел факультета ВМиК МГУ имени М.В.Ломоносова, 2010.