ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Методические рекомендации к лабораторным работам для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения

Часть 1

Могилев 2018

УДК 621.01
ББК 36.4
О 87

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ПОИТ «7» марта 2018 г., протокол № 8

Составитель канд. техн. наук, доц. Н. Н. Горбатенко

Рецензент канд. техн. наук, доц. И. В. Лесковец

Методические рекомендации разработаны на основе рабочей программы по дисциплине «Объектно-ориентированное программирование» для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» дневной формы обучения и предназначены для использования при проведении лабораторных работ.

Учебно-методическое издание

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Часть 1

Ответственный за выпуск К. В. Овсянников

Технический редактор А. А. Подошевко

Компьютерная верстка Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский университет», 2018

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Содержание

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

1 Лабораторная работа № 1. Классы, свойства, индексаторы

Цель работы: получить навыки создания простейших классов с использованием принципа инкапсуляции, свойств, индексаторов, перегрузки операторов.

1.1 Необходимые теоретические сведения

1 Основные сведения о классах [1, с. 148–152].
2 Создание объектов [1, с. 153–154].
3 Объявление и использование методов класса [1, с. 155–157].
4 Создание и использование конструкторов [1, с. 166–170].
5 Перегрузка конструкторов [1, с. 241–242].
6 Вызов перегружаемого конструктора с помощью ключевого слова `this` [1, с. 245–246].
7 Модификаторы доступа к членам класса [1, с. 210–211].
8 Основы перегрузки операторов [1, с. 270–277].
9 Свойства [1, с. 313–317].
10 Автоматически реализуемые свойства [1, с. 318–319].
11 Индексаторы [1, с. 304–307].
12 Понятие инкапсуляции как принципа ООП [1, с. 42].

1.2 Индивидуальные задания

Спроектировать класс согласно варианту. Продемонстрировать работу класса в консольном приложении. В программе должна выполняться проверка всех разработанных элементов класса.

Вариант 1

Создать класс `Point`, содержащий следующие члены класса:

– поля: `int x, y`;

– конструкторы, позволяющие создать экземпляр класса с нулевыми координатами, с заданными координатами;

– методы, позволяющие вывести координаты точки на экран, рассчитать расстояние от начала координат до точки, переместить точку на плоскости на вектор `(a, b)`;

– свойства, позволяющие получить/установить координаты точки (доступное для чтения и записи), умножить координаты точки на скаляр (доступное только для записи);

– индексатор, позволяющий по индексу 0 обращаться к полю `x`, по индексу 1 – к полю `y`, при других значениях индекса выдается сообщение об ошибке;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение полей `x` и `y` на 1; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если значение полей `x` и `y` совпадает, иначе `false`; операции бинарный `+` – одновременно добавляет к полям `x` и `y` значение скаляра.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 2

Создать класс `Triangle`, содержащий следующие члены класса:

– поля: `int a, b, c`;

– конструктор, позволяющий создать экземпляр класса с заданными длинами сторон;

– методы, позволяющие вывести длины сторон треугольника на экран, рассчитать периметр треугольника, рассчитать площадь треугольника;

– свойства, позволяющие получить/установить длины сторон треугольника (доступное для чтения и записи), установить, существует ли треугольник с данными длинами сторон (доступное только для чтения);

– индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю `b`, по индексу 2 – к полю c, при других значениях индекса выдается сообщение об ошибке;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение полей `a, b` и `c` на 1; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если треугольник с заданными длинами сторон существует, иначе – `false`; операции * – одновременно умножает поля `a, b` и `c` на скаляр.

Вариант 3

Создать класс `Rectangle`, содержащий следующие члены класса:

– поля: `int a, b`;

– конструктор, позволяющий создать экземпляр класса с заданными длинами сторон;

– методы, позволяющие вывести длины сторон прямоугольника на экран, рассчитать периметр прямоугольника, рассчитать площадь прямоугольника;

– свойства, позволяющие получить/установить длины сторон прямоугольника (доступное для чтения и записи), установить, является ли данный прямоугольник квадратом (доступное только для чтения);

– индексатор, позволяющий по индексу 0 обращаться к полю `a`, по индексу 1 – к полю `b`, при других значениях индекса выдается сообщение об ошибке;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение полей `a` и `b`; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если прямоугольник с заданными длинами сторон является квадратом, иначе – `false`; операции * – одновременно умножает поля `a` и `b` на скаляр.

Вариант 4

Создать класс `Money`, содержащий следующие члены класса:

– поля: `int first` (номинал купюры); `int second` (количество купюр);

– конструктор, позволяющий создать экземпляр класса с заданными значениям полей;

– методы, позволяющие вывести номинал и количество купюр, определить, хватит ли денежных средств на покупку товара на сумму `N` рублей, определить, сколько штук товара стоимости n рублей можно купить на имеющиеся денежные средства;

– свойства, позволяющие получить/установить значение полей (доступное

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

для чтения и записи), рассчитать сумму денег (доступное только для чтения);

– индексатор, позволяющий по индексу 0 обращаться к полю `first`, по индексу 1 – к полю `second`, при других значениях индекса выдается сообщение об ошибке;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение полей `first` и `second`; операции `!` – возвращает значение `true`, если поле `second` не нулевое, иначе `false`; операции бинарный `+` – добавляет к значению поля `second` значение скаляра.

Вариант 5

Создать класс для работы с одномерным массивом целых чисел. Разработать следующие члены класса:

– поля: `int [] IntArray`;

– конструктор, позволяющий создать массив размерности `n`;

– методы, позволяющие ввести элементы массива с клавиатуры, вывести элементы массива на экран, отсортировать элементы массива в порядке возрастания;

– свойство, возвращающее размерность массива (доступное только для чтения), и свойство, позволяющее умножить все элементы массива на скаляр (доступное только для записи);

– индексатор, позволяющий по индексу обращаться к соответствующему элементу массива;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение всех элементов массива на 1; операции `!` – возвращает значение `true`, если элементы массива не упорядочены по возрастанию, иначе – `false`; операции бинарный `*` – умножить все элементы массива на скаляр; операции преобразования класса массив в одномерный массив (и наоборот).

Вариант 6

Создать класс для работы с двумерным массивом целых чисел. Разработать следующие члены класса:

– поля: `int [,] intArray`;

– конструктор, позволяющий создать массив размерности `n×m`;

– методы, позволяющие ввести элементы массива с клавиатуры, вывести элементы массива на экран, вычислить сумму элементов `i`-го столбца;

– свойства, позволяющие вычислить количество нулевых элементов в массиве (доступное только для чтения), установить значение всех элементов главной диагонали массива, равное скаляру (доступное только для записи);

– двумерный индексатор, позволяющий обращаться к соответствующему элементу массива;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение всех элементов массива на 1; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если двумерный массив является квадратным; операции бинарный `+` – сложить два массива соответствующих размерностей; операции преобразования класса массив в двумерный массив (и наоборот).

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 7

Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие функциональные члены класса:

– поля: `double [][] doubelArray`;

– конструктор, позволяющий создать ступенчатый массив;

– методы, позволяющие ввести элементы массива с клавиатуры, вывести элементы массива на экран, отсортировать элементы каждой строки массива в порядке убывания;

– свойство, возвращающее общее количество элементов в массиве (доступное только для чтения), и свойство, позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи);

– двумерный индексатор, позволяющий обращаться к соответствующему элементу массива;

– перегрузку: операции `++ (--)` – одновременно увеличивает (уменьшает) значение всех элементов массива на 1; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если каждая строка массива упорядочена по возрастанию, иначе – `false`; операции преобразования класса массив в ступенчатый массив (и наоборот).

Вариант 8

Создать класс для работы со строками. Разработать следующие члены класса:

– поле: `string line`;

– конструктор, позволяющий создать строку на основе заданного строкового литерала;

– методы, позволяющие подсчитать количество цифр в строке, выводить на экран все символы строки, встречающиеся в ней ровно один раз, вывести на экран самую длинную последовательность повторяющихся символов в строке;

– свойство, возвращающее общее количество символов в строке (доступное только для чтения);

– индексатор, позволяющий по индексу обращаться к соответствующему символу строки (доступный только для чтения);

– перегрузку: операции унарного `!` – возвращает значение `true`, если строка не пустая, иначе – `false`; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если строка является палиндромом, `false` – в противном случае; операции `&` – возвращает значение `true`, если строковые поля двух объектов посимвольно равны (без учета регистра), иначе – `false`; операции преобразования класса строка в тип `string` (и наоборот).

Вариант 9

Создать класс для работы со строками. Разработать следующие члены класса:

– поле: `StringBuilder line`;

– конструктор, позволяющий создать строку на основе заданного строкового литерала, и конструктор, позволяющий создавать пустую строку;

– методы, позволяющие подсчитать количество пробелов в строке,

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

заменить в строке все прописные символы на строчные, удалить из строки все знаки препинания;

– свойство, возвращающее общее количество элементов в строке (доступное только для чтения), и свойство, позволяющее установить значение поля в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи);

– индексатор, позволяющий по индексу обращаться к соответствующему символу строки;

– перегрузку: операции унарного `+ (-)` – преобразующей строку к строчным (прописным) символам; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если строка не пустая, иначе – `false`; операции `&` – возвращает значение `true`, если строковые поля двух объектов посимвольно равны (без учета регистра), иначе – `false`; операции преобразования строки в тип `StringBuilder` (и наоборот).

Вариант 10

Самостоятельно изучите тип данных `DateTime`, на основе которого необходимо создать класс для работы с датой. Данный класс должен содержать следующие члены класса:

– поле `DataTime data`;

– конструкторы, позволяющие установить заданную дату, дату 1.01.2000;

– методы, позволяющие вычислить дату предыдущего дня, вычислить дату следующего дня, определить сколько дней осталось до конца месяца;

– свойства, позволяющие установить или получить значение поле класса (доступно для чтения и записи), определить, является ли год високосным (доступно только для чтения);

– индексатор, позволяющий определить дату `i`-го по счету дня относительно установленной даты (при отрицательных значениях индекса отсчет ведется в обратном порядке);

– перегрузку: операции `!` – возвращает значение `true`, если установленная дата не является последним днем месяца, иначе – `false`; констант `true` и `false` – обращение к экземпляру класса дает значение `true`, если установленная дата является началом года, иначе – `false`; операции `&` – возвращает значение `true`, если поля двух объектов равны, иначе `false`.

Контрольные вопросы

1 Дайте определение терминам «класс» и «объект». Как соотносятся эти понятия между собой?

2 Приведите синтаксис создания объекта в общем виде. Проиллюстрируйте его фрагментом программы.

3 Какие члены класса содержат код?

4 Какие члены класса содержат данные?

5 В чём состоит назначение конструктора?

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

6 Чем конструктор отличается от обычного метода?
7 Перечислите типы конструкторов класса.
8 Сколько конструкторов может содержать класс?
9 Объясните механизм вызова перегружаемого конструктора с помощью ключевого слова `this`.
10 Что понимается под термином «деструктор»?
11 В чём состоит назначение деструктора?
12 Перечислите модификаторы доступа к членам класса.
13 В чём отличия свойств от полей?
14 Приведите формат объявления свойства.
15 Каким идентификатором представлено в `set`-аксессоре новое значение свойства?
16 Объясните назначение механизма автоматически реализуемых свойств.
17 Какова роль служебного слова `this` в индексаторе?
18 Может ли в одном классе быть несколько индексаторов?
19 Какой тип допустим для параметра индексатора?
20 Объясните принцип инкапсуляции и его применение к классам.
21 Объясните принцип работы индексатора.

2 Лабораторная работа № 2. Агрегация и композиция классов

Цель работы: приобрести навыки проектирования классов с использованием отношений агрегации и композиции.

2.1 Необходимые теоретические сведения

1 Основные отношения между классами: агрегация, композиция, наследование [2, с. 237–238].
2 Программная реализация на языке `C#` агрегации и композиции классов [2, с. 239–241].

2.2 Индивидуальные задания

Написать программы решения задач с использованием классов, находящихся в отношении агрегации (задача 1) и композиции (задача 2). Каждый из создаваемых классов должен иметь не менее трёх методов, свойств и конструкторов.

Вариант 1

1 Создать объект класса Здание, используя класс Отопительная система. Методы: включить отопительную систему, выключить отопительную систему, определить температуру в здании, заполнить отопительную систему водой, слить воду из отопительной системы.
2 Создать объект класса Завод, используя класс Склад деталей. Методы:

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

изготовить деталь, доставить деталь на склад, получить деталь из склада, подсчитать количество деталей на складе, использовать деталь при сборке изделия, вывести на консоль наименование детали.

Вариант 2

1 Создать объект класса Больница, используя класс Приемное отделение. Методы: регистрация больного, выписка больного, установить диагноз болезни, назначить курс лечения, направить на обследование.

2 Создать объект Аэропорт, используя класс Взлётная полоса. Методы: взлёт самолёта, посадка самолёта, вывести на консоль длину взлетной полосы, количество взлетных полос.

Вариант 3

1 Создать объект класса Ресторан, используя класс Кухня. Методы: приготовить суп, салат, кофе, составить меню, принять заказ клиента, выполнить заказ, принести приготовленную еду посетителю ресторана.

2 Создать объект Библиотека, используя класс Книгохранилище. Методы: задать название книги, дополнить книгохранилище книгой, вывести на консоль количество книг, выдать книгу читателю.

Вариант 4

1 Создать объект класса Текст, используя класс Абзац. Методы: дополнить текст, вывести на консоль текст, заголовок текста.

2 Создать объект класса Наседка, используя классы Птица, Кукушка. Методы: летать, петь, нести яйца, высиживать птенцов.

Вариант 5

1 Создать объект класса Автомобиль, используя класс Колесо. Методы: ехать, заправляться, менять колесо, вывести на консоль марку автомобиля.

2 Создать объект класса Текстовый файл, используя класс Файл. Методы: создать, переименовать, вывести на консоль содержимое, дополнить, удалить.

Вариант 6

1 Создать объект класса Самолет, используя класс Крыло. Методы: летать, задавать маршрут, вывести на консоль маршрут.

2 Создать объект класса Одномерный массив, используя класс Массив. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить).

Вариант 7

1 Создать объект класса Беларусь, используя класс Область. Методы: вывести на консоль столицу, количество областей, площадь, областные центры.

2 Создать объект класса Простая дробь, используя класс Число. Методы: вывод на экран, сложение, вычитание, умножение, деление.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 8

1 Создать объект класса Планета, используя класс Материк. Методы: вывести на консоль название материка, планеты, количество материков.

2 Создать объект класса Дом, используя классы Окно, Дверь. Методы: закрыть на ключ, вывести на консоль количество окон, дверей.

Вариант 9

1 Создать объект класса Звездная система, используя классы Планета, Звезда, Луна. Методы: вывести на консоль количество планет в звездной системе, название звезды, добавление планеты в систему.

2 Создать объект класса Роза, используя классы Лепесток, Бутон. Методы: расцвести, увянуть, вывести на консоль цвет бутона.

Вариант 10

1 Создать объект класса Компьютер, используя классы Винчестер, Дисковод, ОЗУ. Методы: включить, выключить, проверить на вирусы, вывести на консоль размер винчестера.

2 Создать объект класса Дерево, используя классы Лист. Методы: зацвести, опасть листьям, покрыться инеем, пожелтеть листьям.

Вариант 11

1 Создать объект класса Квадрат, используя классы Точка, Отрезок. Методы: задание размеров, растяжение, сжатие, поворот, изменение цвета.

2 Создать объект класса Пианино, используя класс Клавиша. Методы: настроить, играть на пианино, нажимать клавишу.

Вариант 12

1 Создать объект класса Круг, используя классы Точка, Окружность. Методы: задание размеров, изменение радиуса, определение принадлежности точки данному кругу.

2 Создать объект класса Фотоальбом, используя класс Фотография. Методы: задать название фотографии, дополнить фотоальбом фотографией, вывести на консоль количество фотографий.

Вариант 13

1 Создать объект класса Котёнок, используя классы Животное, Кошка. Методы: вывести на консоль имя, подать голос, рожать потомство (создавать себе подобных).

2 Создать объект класса Год, используя классы Месяц, День. Методы: задать дату, вывести на консоль день недели по заданной дате, рассчитать количество дней, месяцев в заданном временном промежутке.

Вариант 14

1 Создать объект класса Сутки, используя классы Час, Минута. Методы: вывести на консоль текущее время, рассчитать время суток (утро, день, вечер, ночь).

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

2 Создать объект класса Птица, используя класс Крылья. Методы: летать, добывать пищу, питаться.

Вариант 15

1 Создать объект класса Тигр, используя класс Когти. Методы: рычать, бежать, добывать пищу.

2 Создать объект класса Гитара, используя класс Струна. Методы: играть, натягивать струну.

Контрольные вопросы

1 В каких отношениях могут находиться классы при ООП?
2 Дайте определение агрегации и композиции.
3 В чем заключается различие между агрегацией и композицией классов?
4 Объясните механизм программной реализации на языке C# агрегации и композиции.
5 Какие преимущества есть у композиции и агрегации перед наследованием?

3 Лабораторная работа № 3. Наследование классов

Цель работы: получить представление о процессе объектно-ориентированной декомпозиции задачи, приобретение навыков программирования с использованием наследования классов.

3.1 Необходимые теоретические сведения

1 Основы наследования [1, с. 329–332].
2 Доступ к членам класса при наследовании [1, с. 333–335].
3 Организация защищённого доступа [1, с. 336–337].
4 Конструкторы и наследование [1, с. 337–339].
5 Вызов конструкторов базового класса [1, с. 339–343].
6 Наследование и сокрытие имен [1, с. 343–344].
7 Применение ключевого слова `base` для доступа к скрытому имени [1, с. 344–346].
8 Создание многоуровневой иерархии классов [1, с. 346–349].
9 Порядок вызова конструкторов [1, с. 349–350].
10 Ссылки на базовый класс и объекты производных классов [1, с. 351–355].
11 Динамическая идентификация типов [1, с. 538–540].

3.2 Индивидуальные задания

1 Построить иерархию классов предметной области согласно варианту. В качестве основы иерархии использовать обычный класс (не абстрактный).
2 В классах описать конструкторы с параметрами и конструкторы по

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

умолчанию, свойства для установки и получения значений полей классов, методы для описания поведения объектов. Каждый из создаваемых классов должен иметь не менее трёх методов, свойств, конструкторов.

3 Для каждого созданного класса переопределить методы `Equals()`, `ToString()` класса `object`. Метод `Equals` переопределить так, чтобы объекты считались равными, если равны значения полей объектов, а не ссылки объектов.

4 В методе `Main()`:

– продемонстрировать всю реализованную функциональность классов;

– создать массив из объектов базового класса, заполнить его ссылками на производные классы, вывести на экран элементы массива;

– создать два объекта базового класса с совпадающими данными и проверить, что ссылки на объекты не равны, а объекты равны, вывести значения хеш-кодов для объектов;

– продемонстрировать использование операторов `is`, `as` (например, выводя в консоль имя класса).

Варианты заданий

1 Студент, преподаватель, заведующий кафедрой, персона.
2 Небоскрёб, дача, коттедж, жилое здание.
3 Организация, страховая компания, нефтегазовая компания, завод.
4 Журнал, книга, печатное издание, учебник.
5 Тест, экзамен, выпускной экзамен, испытание.
6 Строительное сооружение, театр, производственный корпус, гостиница.
7 Игрушка, телевизор, товар, молоко.
8 Квитанция, накладная, документ, счёт.
9 Автомобиль, поезд, самолёт, транспортное средство.
10 Республика, монархия, королевство, государство.
11 Корабль, пароход, парусник, корвет.
12 Двигатель, бензиновый двигатель, дизельный двигатель, реактивный двигатель.
13 Деталь, узел, механизм, изделие.
14 Млекопитающее, парнокопытное, птица, животное.
15 Выпускник вуза, Бакалавр, Магистр, Инженер.

Контрольные вопросы

1 Для чего используется наследование классов?
2 Опишите синтаксис производного класса. Какие модификаторы доступа применяются в иерархиях классов?
3 Объясните правила доступа к членам базового класса для объектов производного класса.
4 Что такое защищённый член класса?
5 Объясните порядок вызова конструкторов базовых классов при работе конструктора производного класса.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

6 Какие действия выполняются автоматически при отсутствии в конструкторе производного класса обращения к конструктору базового класса?
7 Что такое сокрытие при наследовании классов?
8 Каково назначение ключевого слова `new` в производном классе?
9 Должны ли совпадать типы возвращаемых значений при сокрытии методов?
10 Какова последовательность выполнения конструкторов при наследовании?
11 Каков механизм раннего связывания?
12 Как осуществляется доступ к элементам производных и базовых классов?
13 Что такое многоуровневая иерархия классов?
14 Назовите альтернативы наследованию классов.
15 Какие методы и операции класса `object` часто перегружают в его потомках?
16 Каковы назначение и использование операторов `is` и `as`?

4 Лабораторная работа № 4. Виртуальные методы, абстрактные классы, интерфейсы

Цель работы: ознакомиться с понятиями полиморфизма, позднего связывания, приобрести навыки программирования с использованием виртуальных методов, абстрактных классов, интерфейсов.

4.1 Необходимые теоретические сведения

1 Полиморфизм [1, с. 43].
2 Виртуальные методы и их переопределение [1, с. 355–362].
3 Применение абстрактных классов [1, с. 363–367].
4 Интерфейсы. Применение интерфейсных ссылок [1, с. 375–382].
5 Интерфейсные свойства и индексаторы [1, с. 383–387].
6 Наследование интерфейсов [1, с. 387–390].

4.2 Индивидуальные задания

Задание 1

Разработать и реализовать в виде консольного приложения иерархию классов из лабораторной работы №3 с использованием абстрактного класса в качестве основы иерархии.

Для демонстрации полиморфизма в методе `Main()` создать массив из объектов базового класса. Присвоить элементам этого массива ссылки на производные классы, вывести элементы массива на экран, сравнить полученный результат с результатом выполнения лабораторной работы № 3.

Задание 2

Реализовать иерархию классов из лабораторной работы № 3 с использованием интерфейсов, при этом один из классов должен реализовывать как минимум два интерфейса. Продемонстрировать применение интерфейсных ссылок.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Задание 3

Создать консольное приложение в соответствии с выданным вариантом.

Вариант 1

1 Создать абстрактный класс `Figure` с функциями вычисления площади и периметра, а также функцией, выводящей информацию о фигуре на экран.

2 В абстрактном классе `Figure` реализовать метод `CompareTo` так, чтобы можно было отсортировать объекты по их площадям.

3 Создать производные классы: `Rectangle` (прямоугольник), `Circle` (круг), `Triangle` (треугольник).

4 В методе `Main()` создать массив `n` фигур и вывести полную информацию о фигурах на экран, отсортировав объекты по их площадям, а также организовать поиск фигур, площадь которых попадает в заданный диапазон.

Вариант 2

1 Создать абстрактный класс `Function` с функциями вычисления значения по формуле `y=f(x)` в заданной точке, а также функцией, выводящей информацию о виде функции на экран.

2 В абстрактном классе `Function` реализовать метод CompareTo так, чтобы можно было отсортировать функции по коэффициенту a.

3 Создать производные классы: `Line` (`y=ax+b`), `Kub` (`y=ax^3+bx+c`), `Hyperbola` (`y=a/x`).

4 В методе `Main()` создать массив `n` функций и вывести полную информацию о значении данных функций в точке `x`, отсортировав функции по коэффициенту `a`.

Вариант 3

1 Создать абстрактный класс `Edition` с функциями, позволяющими вывести на экран информацию об издании, а также определить, является ли данное издание искомым.

2 В абстрактном классе `Edition` реализовать метод `CompareTo` так, чтобы можно было отсортировать каталог изданий по фамилии автора.

3 Создать производные классы: `Book` (название, фамилия автора, год издания, издательство), `Article` (название, фамилия автора, название журнала, его номер и год издания), `OnlineResource` (название, фамилия автора, ссылка, аннотация).

4 В методе `Main()` создать массив из `n` изданий, вывести полную информацию из каталога, отсортировав каталог изданий по фамилии автора, а также организовать поиск изданий по фамилии автора.

Вариант 4

1 Создать абстрактный класс `Transport` с функциями, позволяющими вывести на экран информацию о транспортном средстве, а также определить грузоподъёмность транспортного средства.

2 В абстрактном классе `Transport` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о машинах по их грузоподъемности.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

3 Создать производные классы: `Car` (марка, номер, скорость, грузоподъемность), `Motorbike` (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна 0), `Truck` (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в 2 раза).

4 В методе `Main()` создать массив из `n` машин, вывести полную информацию из базы на экран, отсортировав массив данных о машинах по их грузоподъемности, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

Вариант 5

1 Создать абстрактный класс `Persona` с функциями, позволяющими вывести на экран информацию о персоне, а также определить её возраст (на момент текущей даты).

2 В абстрактном классе `Persona` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о персонах по дате рождения.

3 Создать производные классы: `Enrollee` (фамилия, дата рождения, факультет), `Student` (фамилия, дата рождения, факультет, курс), `Teacher` (фамилия, дата рождения, факультет, должность, стаж).

4 В методе Main() создать массив из `n` персон, вывести полную информацию из базы на экран, отсортировав массив данных о персонах по дате рождения, а также организовать поиск персон, чей возраст попадает в заданный диапазон.

Вариант 6

1 Создать абстрактный класс `Goods` с функциями, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли он сроку годности на текущую дату.

2 В абстрактном классе `Goods` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о товарах по их цене.

3 Создать производные классы: `Product` (название, цена, дата производства, срок годности), `Party` (название, цена, количество штук, дата производства, срок годности), `Kit` (название, цена, перечень продуктов).

4 В методе `Main()` создать массив из `n` товаров, вывести полную информацию из базы на экран, отсортировав массив данных о товарах по их цене, а также организовать поиск просроченного товара (на момент текущей даты).

Вариант 7

1 Создать абстрактный класс `Goods` с функциями, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли она искомому типу.

2 В абстрактном классе `Goods` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о товарах по возрасту детей, на которых он рассчитан.

3 Создать производные классы: `Toy` (название, цена, производитель, материал, возраст, на который рассчитана), `Book` (название, автор, цена, издательство,

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

возраст, на который рассчитана), `SportsEquipment` (название, цена, производитель, возраст, на который рассчитан).

4 В методе `Main()` создать массив из `n` товаров, вывести полную информацию из базы на экран, отсортировав массив данных о товарах по возрасту детей, на которых он рассчитан, а также организовать поиск товаров определенного типа.

Вариант 8

1 Создать абстрактный класс `TelephoneDirectory` с функциями, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска.

2 В абстрактном классе `TelephoneDirectory` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных справочника по номеру телефона.

3 Создать производные классы: `Persona` (фамилия, адрес, номер телефона), `Organization` (название, адрес, телефон, факс, контактное лицо), `Friend` (фамилия, адрес, номер телефона, дата рождения).

4 В методе `Main()` создать массив из `n` записей, вывести полную информацию из базы на экран, отсортировав массив данных справочника по номеру телефона, а также организовать поиск в базе по фамилии.

Вариант 9

1 Создать абстрактный класс `Client` с функциями, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска.

2 В абстрактном классе `Client` реализовать метод `CompareTo` так, чтобы можно было отсортировать базу данных о клиентах банка по дате открытия их счета.

3 Создать производные классы: `Depositor` (фамилия, дата открытия вклада, размер вклада, процент по вкладу), `Credited` (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), `Organization` (название, дата открытия счета, номер счета, сумма на счету).

4 В методе `Main()` создать массив из `n` клиентов, вывести полную информацию из базы на экран, отсортировав массив данных о клиентах банка по дате открытия их счета, а также организовать поиск клиентов, начавших сотрудничать с банком с заданной даты.

Вариант 10

1 Создать абстрактный класс `Software` с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).

2 В абстрактном классе `Software` реализовать метод `CompareTo` так, чтобы можно было отсортировать массив данных по названию ПО.

3 Создать производные классы: `FreeSoftware` (название, производитель), `SharewareSoftware` (название, производитель, дата установки, срок бесплатного использования), `ProprietarySoftware` (название, производитель, цена, дата

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

установки, срок использования).

4 В методе `Main()` создать массив из `n` видов программного обеспечения, вывести полную информацию из массива на экран, отсортировав массив данных по названию ПО, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.

Контрольные вопросы

1 Что понимается в ООП под термином «полиморфизм»? Объясните основной принцип полиморфизма.

2 Что такое виртуальный метод? Какое ключевое слово используется для определения виртуального метода?

3 Когда осуществляется выбор версии виртуального метода?

4 Какое ключевое слово используется при реализации виртуального метода в производном классе?

5 Какие модификаторы доступа нельзя использовать при определении виртуальных методов?

6 Что такое абстрактный класс? Какое ключевое слово используется при объявлении абстрактных методов?

7 Являются ли абстрактные методы виртуальными?

8 Можно ли создавать цепочку производных классов, используя абстрактный класс?

9 Могут ли быть в программе объекты абстрактного класса?

10 Что такое интерфейс?

11 Чем отличается объявление интерфейса от объявления абстрактного класса?

12 Какие элементы языка `C#` могут быть членами интерфейса?

13 Сколько интерфейсов может наследовать класс?

14 Где должны быть реализованы методы интерфейса?

15 Можно ли реализовать множественный интерфейс?

16 Как проявляется принцип полиморфизма при использовании интерфейсов?

17 Можно ли объявить интерфейс с модификатором `static`?

18 Возможно ли создание ссылочной переменной интерфейсного типа?

19 В чём различия и сходства интерфейса и абстрактного класса?

20 Доступ к каким членам класса, реализующего интерфейс, обеспечивает ссылка с типом интерфейса?

21 Что такое наследование реализации?

22 Что такое наследование функциональности?

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

5 Лабораторная работа № 5. Обобщённые классы, коллекции, итераторы

Цель работы: научиться разрабатывать обобщённые классы, коллекции, итераторы и применять их в программах.

5.1 Необходимые теоретические сведения

1 Создание обобщенного класса [1, с. 576–579, 583–584].
2 Применение ограничений на базовый класс [1, с. 586–602].
3 Класс `List<T>` [1, с. 961–965].
4 Класс `Dictionary<TKey, TValue>` [1, с. 969–972].
5 Назначение, применение итераторов [1, с. 1003–1005].
6 Создание именованного итератора [1, с. 1006–1007].

5.2 Индивидуальные задания

Задание 1

Разработать новую версию программы решения задания 3 из лабораторной работы № 4. В новой версии программы сохранить все классы из предыдущей версии программы и определить новый класс `Collection`, содержащий следующее.

1 Закрытое поле типа `List<T>` – список (варианты заданий с чётными номерами) или `Dictionary<TKey, TValue>` – словарь (варианты заданий с нечётными номерами). В качестве параметра типа `<T>` для `List<T>` использовать базовый тип в созданной иерархии классов. В коллекции `Dictionary<TKey, TValue>` в качестве параметра типа `<TKey>` использовать наименование производного класса, а в качестве `<TValue>` – значение одного из полей производного класса.

2 Конструктор с параметрами и конструктор по умолчанию для инициализации поля, объявленного в п. 1.

3 Метод для добавления элементов в коллекцию.

4 Метод для удаления элементов из коллекции.

5 Метод для просмотра элементов коллекции.

6 Перегруженную версию виртуального метода `ToString()` для формирования строки с информацией об элементах коллекции.

7 Метод, выполняющий сортировку коллекции по заданному условию.

8 Итератор для поиска элементов коллекции, удовлетворяющих заданному условию.

9 Обобщённый метод с типом возврата `List<T>` (здесь `<T>` базовый тип в созданной иерархии классов), который возвращает из коллекции лишь те объекты, тип которых указан в качестве обобщенного параметра метода, т. е. в качестве обобщённого параметра метода указывается тип дочернего класса, а из коллекции возвращаются только объекты указанного типа.

В методе `Main()` создать экземпляр класса `Collection` и продемонстрировать всю функциональность этого класса.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Задание 2

Для заданной предметной области создать программу, состоящую из трех-пяти классов. Каждый из создаваемых классов должен иметь не менее трёх методов, свойств, конструкторов. Предусмотреть использование типа данных – перечисление, коллекций `List<T>` (варианты заданий с нечётными номерами), `Dictionary<TKey, TValue>` (варианты заданий с чётными номерами). Ввод/вывод данных должен быть реализован вне классов.

Вариант 1

Предметная область: АТС. На АТС хранится информация о всех клиентах станции. АТС имеет список тарифов на междугородние разговоры. Клиент АТС может совершать множество звонков в различные города. Система должна:

– позволять вводить информацию о тарифах;

– вводить информацию о клиентах и регистрировать звонки;

– по введенной фамилии о клиенте определять стоимость всех сделанных им звонков в соответствии с действующими тарифами;

– вычислять общую стоимость всех выполненных на АТС звонков.

Вариант 2

Предметная область: Вокзал. Касса вокзала имеет список тарифов на различные направления. При покупке билета регистрируются паспортные данные пассажира. Пассажир покупает билеты на различные направления. Система должна:

– позволять вводить данные о тарифах;

– позволять вводить паспортные данные пассажира и регистрировать покупку билета;

– рассчитывать стоимость купленных пассажиром билетов;

– после ввода наименования направления выводить список всех пассажиров, купивших на него билет.

Вариант 3

Предметная область: ЖЭС. В ЖЭС хранятся тарифы на коммунальные услуги. ЖЭС имеет информацию о всех жильцах. При потреблении жильцами коммунальных услуг информация регистрируется в системе. Система должна позволять выполнять следующие задачи:

– ввод тарифов;

– ввод информации о жильцах и потребленных услугах;

– после ввода фамилии выводить сумму всех потребленных услуг;

– выводить стоимость всех оказанных услуг.

Вариант 4

Предметная область: Аэропорт. Касса аэропорта имеет список тарифов на различные направления. При покупке билета регистрируются паспортные данные. Система должна:

– позволять вводить данные о тарифах;

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

– позволять вводить паспортные данные пассажира и регистрировать покупку билета;
– рассчитывать стоимость купленных пассажиром билетов;
– рассчитывать стоимость всех проданных билетов.

Вариант 5

Предметная область: Банк. Информационная система банка хранит описание процентов по различным вкладам. Система хранит информацию о вкладчиках и сделанных ими вкладах. Каждый клиент может поместить в банк только один вклад. Система должна позволять выполнять следующие задачи:
– хранить информацию о процентах по вкладам;
– хранить информацию о клиентах;
– пополнять клиенту величину вклада;
– вычислять общую сумму выплат по процентам для всех вкладов.

Вариант 6

Предметная область: Отдел расчета зарплаты. Информационная система отдела расчета зарплаты на предприятии хранит данные о величине оплаты за различные виды работ. Система хранит информацию о работниках предприятия. Система должна позволять выполнять следующие задачи:
– вводить информацию о различных видах работ;
– вводить информацию о работниках и выполненных ими работах;
– после ввода фамилии выводить для работника зарплату;
– выводить сумму выплат всем работникам.

Вариант 7

Предметная область: Фирма грузоперевозок. Фирма имеет список тарифов по перевозке грузов. Клиент регистрируется в системе, после чего может заказать перевозку определенного объема груза. Система должна позволять выполнять следующие задачи:
– ввод тарифов;
– регистрация клиента и заказ на перевозку грузов;
– вывод суммы заказа для определенного клиента;
– подсчет суммарной стоимости всех заказов.

Вариант 8

Предметная область: Гостиница. Информационная система гостиницы хранит информацию о всех номерах и их стоимости. Система регистрирует клиентов. Каждый клиент может заказать один номер. При попытке заказа номера, который занят, выводится предупреждение. Система должна позволять выполнять следующие задачи:
– ввод информации о номерах и их стоимости;
– регистрация клиента и заказ номера;
– вывод списка незанятых номеров;
– после ввода фамилии клиента вывод стоимости проживания.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 9

Предметная область: Интернет-оператор. Провайдер имеет различные тарифы доступа в Интернет за 1 Мбайт в зависимости от величины абонентской платы. Информационная система провайдера хранит данные о клиентах. Система должна позволять выполнять следующие задачи:

– ввод тарифов;
– регистрация пользователя;
– ввод данных о потребленном трафике для конкретного пользователя;
– подсчет общей стоимости реализованного трафика;
– поиск клиента, заплатившего наибольшую стоимость за услуги.

Вариант 10

Предметная область: Интернет-магазин. В информационной системе хранятся данные о товарах. Клиент звонит в магазин и оставляет заказ на товар. Система должна позволять выполнять следующие задачи:

– ввод информации о товарах;
– регистрация заказа клиента на покупку определенного товара;
– после ввода фамилии покупателя вывод списка заказанных им товаров;
– после ввода фамилии покупателя вывод суммы заказа.

Контрольные вопросы

1 Что понимают в языке C# под термином обобщение?

2 Как создать класс обобщенного типа? Сколько у него может быть параметров?

3 Какие задачи решает ограничение в обобщении? Перечислите виды ограничений.

4 Можно ли наследовать от обобщенного класса?

5 Можно ли определять обобщенные свойства, методы?

6 Объясните назначение и работу операций `Boxing` и `Unboxing`.

7 Что такое коллекции?

8 Опишите состав пространства имен `System.Collections` и дайте характеристику основных типов-коллекций.

9 Объясните назначение классов `List<T>` и `Dictionary<TKey, TValue>`.

10 Назовите основные методы и свойства классов `List<T>` и `Dictionary<TKey, TValue>`.

11 Какие интерфейсы реализуются в классах `List<T>` и `Dictionary<TKey, TValue>`?

12 Перечислите стандартные интерфейсы `.NET`, которые определяют возможности сортировки и просмотра объектов с помощью оператора `foreach`.

13 Объясните, как используется конструкция `yield`.

14 Что такое итератор? Какой интерфейс описывает свойства и поведение объекта-итератора? Объясните принцип работы итератора.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

6 Лабораторная работа № 6. Делегаты, события, лямбда-выражения

Цель работы: научиться разрабатывать программы с использованием делегатов, событий и лямбда-выражений.

6.1 Необходимые теоретические сведения

1 Делегаты [1, с. 473–475].
2 Групповая адресация [1, с. 478–480].
3 Блочные лямбда-выражения [1, с. 492–493].
4 События [1, с. 494–495].
5 Групповая адресация событий [1, с. 496–497].
6 Применение лямбда-выражений вместе с событиями [1, с. 504–505].
7 Рекомендации по обработке событий в среде `.NET Framework` [1, с. 506–507].
8 Обобщенные делегаты [1, с. 610–611].

6.2 Индивидуальные задания

Задание 1

Разработать программу с использованием делегатов. В программе следует:

1) определить делегат, принимающий несколько параметров различных типов и возвращающий значение некоторого типа;

2) написать не менее двух методов, соответствующих данному делегату;

3) написать метод, принимающий разработанный делегат, в качестве одного из входных параметров. Осуществить вызов метода, передавая ему в качестве параметра-делегата:

– один из методов, разработанных в п. 2;
– лямбда-выражение;

4) повторить п. 3, используя вместо разработанного делегата обобщенный делегат `Func< >` или `Action< >`, соответствующий сигнатуре разработанного делегата.

5) используя многоадресный делегат организовать цепочку вызовов методов, разработанных в п. 2;

6) создать блочное лямбда-выражение, соответствующее делегату, описанному в п. 1, и продемонстрировать его применение.

Задание 2

Вариант 1

Разработать программу «Охота на волков». В лесу размером `N x N` хаотически движутся с некоторой скоростью четыре волка и два охотника. При совпадении координат волка и охотника волк исчезает и охотник оповещает другого охотника и лесничество о своей добыче. Если волк достигнет края леса, то он

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

спасается. Процесс движения волков и охотников визуализировать на дисплее с помощью символов псевдографики.

Вариант 2

Разработать программу «Авиаразведка». Создать условную карту местности размером `N x N`. В ячейках карты расположить некоторое количество различных целей (например, танки, пушки, склады с горюче-смазочными материалами и др.). Из произвольной точки, находящейся на границе карты, стартует самолёт-разведчик. Достигнув противоположной границы карты, он меняет курс и летит в обратном направлении. Так продолжается до тех пор, пока не будет покрыта вся карта местности. Самолёт-разведчик фиксирует цели, чьи координаты совпадают с его координатами и по достижении границ карты сообщает в штаб информацию о типе и количестве обнаруженных целей. Процесс полёта самолета и расположение целей визуализировать с помощью символов псевдографики.

Вариант 3

Создать программу, в которой реализуется следующая ситуация. В лесу размером `NxN` потерялась девочка. Она хаотично ходит по лесу со скоростью `V` и при этом кричит «ауу» в надежде, что кто-нибудь её услышит. По лесу также хаотично двигаются со скоростью `2V` – мама, папа, дедушка и бабушка девочки. Если кто-нибудь из них услышит девочку, а услышать её могут, если между девочкой и слушателем не более четырёх квадратов леса, то девочка будет спасена. Если она самостоятельно достигнет края леса, то также будет спасена. Процесс поиска визуализировать, используя символы псевдографики.

Вариант 4

Создать программу «Сапёр». В программе реализовать класс `Bomb` и класс `TimerBomb`.

Функциональность класса `Bomb`: конструктор, передаваемый параметр – время, через которое бомба взорвется; метод `Code` – принимает код обезвреживания бомбы, если принятый код совпал с кодом, сгенерированным бомбой, – бомба считается обезвреженной; при создании экземпляра бомбы генерируется случайный код в заданном диапазоне и задается время срабатывания бомбы (например, код от 1 до 10, время срабатывания – 5 с).

Функциональность класса `TimerBomb` – генерирует событие `OnTick` через заданный промежуток времени. Для генерации события `OnTick` можно использовать класс `Timer` из пространства имён `System.Timers`.

Контрольные вопросы

1 Что такое делегат?
2 Назовите этапы создания и применения делегатов.
3 Как осуществляется вызов методов с помощью делегата?
4 Что такое многоадресная передача делегатов?

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

5 Какие операторы языка C# используются для создания и удаления цепочки методов для многоадресных делегатов?

6 Каким должен быть тип возвращаемого значения для многоадресных делегатов?

7 Для чего используются обобщенные делегаты Func< > и Action< >?

8 Что такое лямбда-выражение?

9 Объясните синтаксис создания одиночных и блочных лямбда-выражений.

10 Как написать лямбда-выражение, соответствующее делегату?

11 Что такое событие?

12 В чём заключается механизм события?

13 Каков порядок создания пользовательского события?

14 Как используются методы класса в роли обработчика события?

15 Какой синтаксис должны иметь .NET-совместимые обработчики событий?

7 Лабораторная работа № 7. Использование LINQ для работы с данными

Цель работы: ознакомиться с языком интегрированных запросов LINQ и его использованием для работы с данными.

7.1 Необходимые теоретические сведения

1 Основы LINQ: простой запрос, неоднократное выполнение запросов, связь между типами данных в запросе, общая форма запроса [1, с. 638–643].

2 Отбор запрашиваемых значений с помощью оператора where [1, с. 644–645].

3 Сортировка результатов запроса с помощью оператора orderby [1, с. 646–648].

4 Оператор select [1, с. 649–652].

5 Применение вложенных операторов from [1, с. 653–654].

6 Группирование результатов с помощью оператора group [1, с. 655–657].

7 Продолжение запроса с помощью оператора into [1, с. 655–657].

8 Применение оператора let для создания временной переменной в запросе [1, с. 659–660].

9 Объединение двух последовательностей с помощью оператора join [1, с. 660–663].

10 Анонимные типы [1, с. 663–665].

11 Методы запроса. Формирование запросов с помощью методов запроса [1, с. 670–675].

7.2 Индивидуальные задания

Разработать консольное приложение согласно варианту. Запросы для поиска данных составлять с использованием технологии доступа к данным LINQ.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Набор данных реализовать с помощью динамических структур данных, используя обобщенный класс `List<T>` – список (варианты заданий с чётными номерами) или обобщенный класс `Queue<T>` – очередь (варианты заданий с чётными номерами). Запись в задании реализовать в виде класса.

1 Счет в банке представляет собой структуру с полями: номер счета, код счета, фамилия владельца, сумма на счете, дата открытия счета, годовой процент начисления. Поиск по номеру счета, дате открытия и владельцу.

2 Запись о товаре на складе представляет собой структуру с полями: номер склада, код товара, наименование товара, дата поступления на склад, срок хранения в днях, количество единиц товара, цена за единицу товара. Поиск по номеру склада, коду товара, дате поступления и сроку хранения (просроченные и непросроченные товары).

3 Запись о преподаваемой дисциплине представляется следующей структурой: код дисциплины в учебном плане, наименование дисциплины, фамилия преподавателя, код группы, количество студентов в группе, количество часов лекций, количество часов практики, наличие курсовой работы, вид итогового контроля (зачет или экзамен). Зачет – 0,35 ч на одного студента, экзамен – 0,5 ч на студента. Поиск осуществлять по фамилии преподавателя, коду группы, наличию курсовой, виду итогового контроля.

4 Информационная запись о книге, выданной на руки абоненту, представляет собой структуру следующего вида: номер читательского билета, фамилия абонента, дата выдачи, срок возврата (количество дней), автор, название, год издания, издательство, цена. Поиск по полям: номер читательского билета, автор, издательство, дата возврата (просроченные).

5 Информационная запись о файле содержит поля: каталог, имя файла, расширение, дата и время создания, атрибуты «только чтение», «скрытый», «системный», признак удаления, количество выделенных секторов (размер сектора принять равным 512 байт). Поиск выполнять по каталогу, дате создания, по признаку удаления.

6 Разовый платеж за телефонный разговор является структурой с полями: фамилия плательщика, номер телефона, дата разговора, тариф за минуту разговора, скидка (в процентах), время начала разговора, время окончания разговора. Поиск по фамилии, дате разговора, номеру телефона.

7 Модель компьютера характеризуется кодом и названием марки компьютера, типом процессора, частотой работы процессора, объемом оперативной памяти, объемом жесткого диска, объемом памяти видеокарты, стоимостью компьютера в условных единицах и количеством экземпляров, имеющихся в наличии. Поиск по типу процессора, объему ОЗУ, памяти видеокарты и жесткого диска.

8 Список абонентов сети кабельного телевидения состоит из элементов следующей структуры: фамилия, район, адрес, телефон, номер договора, дата заключения договора, оплата установки, абонентская плата помесячно, дата последнего платежа. Поиск по фамилии, району, дате заключения договора, дате последнего платежа.

9 Сотрудник представлен структурой Person с полями: табельный номер, номер отдела, фамилия, оклад, дата поступления на работу, процент надбавки,

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

подоходный налог, количество отработанных дней в месяце, количество рабочих дней в месяце, начислено, удержано. Поиск по номеру отдела, полу, дате поступления, фамилии.

10 Запись о багаже пассажира авиарейса содержит следующие поля: номер рейса, дата и время вылета, пункт назначения, фамилия пассажира, количество мест багажа, суммарный вес багажа. Поиск выполнять по номеру рейса, дате вылета, пункту назначения, весу багажа (превышение максимально допустимого).

11 Одна учетная запись посещения спорткомплекса имеет структуру: фамилия клиента, код и вид спортивного занятия, фамилия тренера, дата и время начала, количество минут, тариф за минуту. Поиск по фамилии клиента и тренера, по виду занятия, по дате начала, по количеству минут (больше или меньше).

12 Одна запись о медикаменте содержит следующие поля: номер аптеки, название лекарства, количество упаковок, имеющееся в наличии в данной аптеке, стоимость одной упаковки, дата поступления в аптеку, срок хранения (в днях). Поиск по номеру аптеки, наименованию препарата, дате поступления.

13 Одна запись журнала содержит поля: код игрушки, название игрушки, тип игрушки, возрастные границы (например, от 10 до 15), цена за единицу, количество в наличии, дата поступления в магазин, поставщик. Поиск по дате поступления, поставщику, возрастным границам.

14 Один элемент – автомобиль – представляет собой в базе данных структуру с полями: фамилия владельца, код марки автомобиля, марка автомобиля, требуемая марка бензина, мощность двигателя, объем бака, остаток бензина, объем масла. Дана фиксированная цена литра бензина и заливки масла. Поиск по марке автомобиля, марке бензина, мощности двигателя, фамилии владельца.

15 Одна запись в журнале зимней экзаменационной сессии представляет собой структуру с полями: курс, код группы, фамилия студента, номер зачетной книжки, дисциплина, оценка за экзамен по дисциплине. Вычисляются средние баллы по дисциплине, по группе, по курсу. Поиск по курсу, по группе, по номеру зачетной книжки, по фамилии, по оценкам.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Контрольные вопросы

1 Что такое LINQ?

2 Каковы структура и работа простого запроса LINQ?

3 Какой оператор используется для отбора данных, возвращаемых по запросу?

4 Как отсортировать результаты запроса LINQ?

5 Объясните назначение следующих операторов запроса LINQ: group, into, let, join.

6 Для чего предназначен анонимный тип? Опишите синтаксис анонимного типа.

7 Объясните назначение основных методов запроса LINQ.

8 Лабораторная работа № 8. Регулярные выражения

Цель работы: получить практические навыки по программированию регулярных выражений в языке C#.

8.1 Необходимые теоретические сведения

1 Регулярные выражения [3, с. 355–359].

2 Классы библиотеки .NET для работы с регулярными выражениями [3, с. 359–365].

8.2 Индивидуальные задания

1 В файле с телефонными переговорами клиента определить по какому номеру выполнено максимальное количество звонков.

2 Выполнить анализ кода программы на наличие в нем всех циклов for языка C#. Напечатать результаты анализа на экране монитора.

3 Считать текст из файла и вывести на экран монитора строку, содержащую максимальное количество знаков пунктуации.

4 В файле с телефонными переговорами клиента выполнить сортировку переговоров по их стоимости.

5 Выполнить анализ кода программы на наличие в нем комментариев языка C#. Напечатать результаты анализа на экране монитора.

6 Считать текст из файла и вывести его на экран монитора, заменив цифры словами, например, «0» на слово «ноль»; «1» на слово «один» и т. д.

7 В файле с телефонными переговорами клиента определить, по какому номеру выполнено максимальное количество звонков.

8 Выполнить анализ кода программы на наличие в нем всех операторов присваивания языка C# (не учитывать записи типа for (i=0; i<=10; i++) …).

9 Напечатать результаты анализа на экране монитора. Считать текст из файла и вывести на экран монитора строку, содержащую максимальное количество чисел (не цифр). В файле с телефонными переговорами клиента определить, по какому номеру был самый продолжительный по времени разговор.

10 Выполнить анализ кода программы на наличие в нем всех операторов вывода информации на экран (консоль) языка C#. Напечатать результаты анализа на экране монитора.

11 Считать текст из файла и вывести его на экран монитора только цитаты текста, т. е. предложения, заключенные в кавычки.

12 В файле с телефонными переговорами клиента определить, по какому номеру выполнены подряд (2 и более соединений) звонки.

13 Выполнить анализ кода программы на наличие в нем всех операторов ввода информации с клавиатуры (консоли) языка C#. Напечатать результаты анализа на экране монитора.

14 Считать текст из файла и вывести на экран монитора строку, содержащую максимальное количество повторяющихся слов.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

15 В файле с телефонными переговорами клиента определить все номера, время связи с которыми было менее 5 с.

Примечания

1 В некоторых вариантах индивидуальных заданий необходимо выполнить анализ файла телефонных переговоров клиента. В этом файле, кроме прочей информации, должны содержаться записи об абонентах, включающих следующие сведения:

– номер абонента, с которым выполнялось соединение;
– дату и время соединения;
– продолжительность соединения;
– стоимость переговоров.

2 Если требуется выполнить анализ кода программы на наличие в нем различных элементов языка C#, то код программы должен быть представлен текстовым файлом с расширением «.cs», в который следует записать необходимую информацию в соответствии с индивидуальным заданием.

Контрольные вопросы

1 Для чего предназначены регулярные выражения?
2 Перечислите основные действия, которые можно выполнять над строками с помощью регулярных выражений.
3 Из каких элементов состоит язык описания регулярных выражений?
4 Что такое метасимволы?
5 Перечислите наиболее употребительные метасимволы.
6 Что такое мнимые метасимволы? Приведите примеры использования мнимых метасимволов.
7 Какую роль в регулярных выражениях выполняют повторители? Приведите примеры повторителей.
8 Перечислите основные методы регулярных выражений.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

9 Лабораторная работа № 9. Многопоточное программирование

Цель работы: ознакомиться с организацией многопоточной обработки данных, получить основные навыки программирования с использованием потоков.

9.1 Необходимые теоретические сведения

1 Основы многопоточной обработки [1, с. 834–835].
2 Создание и запуск потока [1, с. 836–838].
3 Создание нескольких потоков [1, с. 839–841].
4 Определение момента окончания потока [1, с. 841–844].
5 Передача аргумента потоку [1, с. 844–846].
6 Приоритеты потоков [1, с. 847–849].
7 Синхронизация потоков [1, с. 849–853].

9.2 Индивидуальные задания

Разработать консольное приложение согласно варианту. Требования к программе: реализовать возможность задавать приоритет каждого из порожденных потоков; использовать символы псевдографики для визуализации потоков на экране дисплея.

1 Умножение матрицы на вектор. Обработку одной строки матрицы производить в порожденном потоке.

2 Поиск всех простых чисел (простым называется число, которое является своим наибольшим делителем) в указанном интервале чисел, разделенном на несколько диапазонов. Обработка каждого диапазона производится в порожденном потоке.

Классический алгоритм Евклида определения наибольшего общего делителя двух целых чисел (x, y) может применяться при следующих условиях:

– оба числа `x` и `y` неотрицательные;

– оба числа `x` и `y` отличны от нуля.

На каждом шаге алгоритма выполняются сравнения:

– если `x == y`, то ответ найден;

– если `x < y`, то `y` заменяется значением `y-x`;

– если `x > y`, то `x` заменяется значением `x-y`.

3 Винни-Пух и пчелы. Заданное количество пчел добывают мед равными порциями, задерживаясь в пути на случайное время. Винни-Пух потребляет мед порциями заданной величины за заданное время и столько же времени может прожить без питания. Работа каждой пчелы реализуется в порожденном потоке.

4 Шарики. Координаты заданного количества шариков изменяются на случайную величину по вертикали и горизонтали. При выпадении шарика за нижнюю границу допустимой области шарик исчезает. Изменение координат каждого шарика в отдельном потоке.

5 Противостояние нескольких команд. Каждая команда увеличивается на случайное количество бойцов и убивает случайное количество бойцов участника. Борьба каждой команды реализуется в отдельном потоке.

6 Контрольная сумма. Для нескольких файлов (разного размера) требуется вычислить контрольную сумму (сумму кодов всех символов файла). Обработка каждого файла выполняется в отдельном потоке.

7 Бег с препятствиями. Создается условная карта трассы в виде матрицы, ширина которой соответствует количеству бегунов, а высота фиксирована, которая содержит произвольное количество единиц (препятствий) в произвольных ячейках. Стартующие бегуны (потоки) перемещаются по трассе и при встрече с препятствием задерживаются на фиксированное время. По достижении финиша бегуны сообщают свой номер.

8 Создать два потока. Первый ищет числа Фибоначчи (каждое последующее число равно сумме двух предыдущих чисел), второй – простые числа. Результат работы каждого потока сохраняется в отдельный файл. После остановки

Электронная библиотека Белорусско-Российского университета http://e.biblio.bru.by/

потока программа производит анализ файлов, выводит их на экран, а так же показывает количество найденных чисел Фибоначчи и простых чисел.

9 Создать два потока. Первый поток производит запись в файл случайных данных, второй производит чтение данных из этого файла и вывод их на экран.

10 Создать приложение, выполняющее вычисление значений функции `y = 23*x^2 - 33`, с шагом `x=0.01`. Первый поток выполняет расчёт функции и добавляет результаты расчёта в конец массива. Второй поток извлекает из массива значения `x` и `y` и выводит их на экран.

11 Создать приложение, которое выполняет сортировку массива данных и отображает процесс сортировки на экране. Первый поток производит сортировку по возрастанию, второй – по убыванию. После каждого перемещения элементов производится вывод на экран текущего состояния сортировки. Каждый поток работает с отдельным экземпляром массива данных.

12 Создать игру, где будут 2…3 барана и волк. При совпадении координат волка с бараном баран исчезает. При совпадении координат баранов появляется новый баран. Все движутся хаотически.

13 Создать три потока, генерирующих случайным образом целые числа от 0 до 9. При нажатии на клавишу «Enter» потоки останавливаются и результат анализируется. Цель анализа – выявить наличие следующих комбинаций цифр в потоках: три одинаковых числа, два одинаковых числа, три единицы, три семерки, две единицы.

14 Создать три потока, каждый из которых управляет перемещением псевдосимвола на экране вдоль оси `X`, и устроить «тараканьи бега» среди них.

15 Поиск указанной строки в указанном файле. Обработка одной строки в порожденном потоке.

Контрольные вопросы

1 Что такое поток?
2 Как создать новый поток?
3 Чем поток отличается от процесса?
4 Как запустить поток после завершения другого потока?
5 Как присвоить потоку имя?
6 Как заблокировать данные от изменения другими потоками?
7 Как остановить выполнение потока?

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

10 Лабораторная работа № 10. Основы создания приложений с использованием Windows Forms

Цель работы: изучить принципы создания `Windows`-приложений, получить базовые навыки разработки приложений с использованием `Windows Forms`.

10.1 Необходимые теоретические сведения

1 Основные особенности операционной системы `Windows` [3, с. 311–312].
2 Событийно-управляемое программирование [3, с. 312–313].
3 Шаблон `Windows`-приложения [3, с. 314–322].
4 Класс `Control` [3, с. 323–325].
5 Элементы управления: `Label`, `Button`, `TextBox`, меню `MainMenu` и `ContextMenu`, `CheckBox`, `RadioButton`, `GroupBox`, `ListBox` [3, с. 325–336].

10.2 Индивидуальные задания

Общая часть задания: написать `Windows`-приложение, заголовок главного окна которого содержит ФИО, группу и номер варианта. В программе должна быть предусмотрена обработка исключений, возникающих из-за ошибочного ввода пользователя.

Вариант 1

Создать меню с командами `Input`, `Calc` и `Exit`. При выборе команды `Input` открывается диалоговое окно, содержащее:

– три поля типа `TextBox` для ввода длин трех сторон треугольника;
– группу из двух флажков (Периметр и Площадь) типа `CheckBox`;
– кнопку типа `Button`.

Обеспечить возможность:

– ввода длин трех сторон треугольника;
– выбора режима с помощью флажков: подсчет периметра и/или площади треугольника.

При выборе команды `Calc` открывается диалоговое окно с результатами. При выборе команды `Exit` приложение завершается.

Вариант 2

Создать меню с командами `Size`, `Color`, `Paint`, `Quit`. Команда `Paint` недоступна. При выборе команды `Quit` приложение завершается. При выборе команды `Size` открывается диалоговое окно, содержащее:

– два поля типа `TextBox` для ввода длин сторон прямоугольника;
– группу из трех флажков (`Red`, `Green`, `Blue`) типа `CheckBox`;
– кнопку типа `Button`.

Обеспечить возможность:

– ввода длин сторон прямоугольника в пикселах в поля ввода;

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

– выбора его цвета с помощью флажков.

После задания параметров команда `Paint` становится доступной. При выборе команды `Paint` в главном окне приложения выводится прямоугольник заданного размера и сочетания цветов или выдается сообщение, если введенные размеры превышают размер окна.

Вариант 3

Создать меню с командами `Input`, `Work`, `Exit`. При выборе команды `Exit` приложение завершает работу. При выборе команды `Input` открывается диалоговое окно, содержащее:

– три поля ввода типа `TextBox` с метками `Radius`, `Height`, `Density`;

– группу из двух флажков (`Volume`, `Mass`) типа `CheckBox`;

– кнопку типа `Button`.

Обеспечить возможность:

– ввода радиуса, высоты и плотности конуса;

– выбора режима с помощью флажков: подсчет объема и/или массы конуса.

При выборе команды `Work` открывается окно сообщений с результатами.

Вариант 4

Создать меню с командами `Input`, `Calc`, `Draw`, `Exit`. При выборе команды `Exit` приложение завершает работу. При выборе команды `Input` открывается диалоговое окно, содержащее:

– поле ввода типа `TextBox` с меткой `Radius`;

– группу из двух флажков (`Square`, `Length`) типа `CheckBox`;

– кнопку типа `Button`.

Обеспечить возможность:

– ввода радиуса окружности;

– выбора режима с помощью флажков: подсчет площади круга (`Square`) и/или длины окружности (`Length`).

При выборе команды `Calc` открывается окно сообщений с результатами. При выборе команды `Draw` в центре главного окна выводится круг введенного радиуса или выдается сообщение, что рисование невозможно (если диаметр превышает размеры рабочей области).

Вариант 5

Создать меню с командами `Input`, `Calc`, `About`. При выборе команды `About` открывается окно с информацией о разработчике. При выборе команды `Input` открывается диалоговое окно, содержащее:

– три поля ввода типа `TextBox` с метками `Number 1`, `Number 2`, `Number 3`;

– группу из двух флажков (`Summ`, `Least multiple`) типа `CheckBox`;

– кнопку типа `Button`.

Обеспечить возможность ввода трех чисел и выбора режима вычислений с помощью флажков: подсчет суммы трех чисел (`Summ`) и/или наименьшего общего

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

кратного двух первых чисел (Least multiple). При выборе команды Calc открывается диалоговое окно с результатами.

Вариант 6

Создать меню с командами Input, Calc, Quit.

Команда Calc недоступна. При выборе команды Quit приложение завершается. При выборе команды Input открывается диалоговое окно, содержащее:

– два поля ввода типа TextBox с метками Number 1, Number 2;
– группу из трех флажков (Summa, Max divisor, Multiply) типа CheckBox;
– кнопку типа Button.

Обеспечить возможность:

– ввода двух чисел;
– выбора режима вычислений с помощью флажков (можно вычислять в любой комбинации такие величины, как сумма, наибольший общий делитель и произведение двух чисел).

При выборе команды Calc открывается окно сообщений с результатами.

Вариант 7

Создать меню с командами Begin, Help, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее:

– поле ввода типа TextBox с меткой input;
– метку типа Label для вывода результата;
– группу из трех переключателей (2, 8, 16) типа RadioButton;
– две кнопки типа Button – Do и OK.

Обеспечить возможность:

– ввода числа в десятичной системе в поле input;
– выбора режима преобразования с помощью переключателей: перевод в двоичную, восьмеричную или шестнадцатеричную систему счисления.

При щелчке на кнопке Do должен появляться результат перевода.

Вариант 8

Создать меню с командами Input color, Change, Exit, Help.

При выборе команды Exit приложение завершает работу. При выборе команды Input color открывается диалоговое окно, содержащее:

– три поля ввода типа TextBox с метками Red, Green, Blue;
– группу из двух флажков (Left, Right) типа CheckBox;
– кнопку типа Button.

Обеспечить возможность ввода RGB-составляющих цвета. При выборе команды Change цвет главного окна изменяется на заданный (левая, правая или обе половины окна в зависимости от установки флажков).

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 9

Создать меню с командами `Input size`, `Choose`, `Change`, `Exit`. При выборе команды `Exit` приложение завершает работу. Команда `Change` недоступна. При выборе команды `Input size` открывается диалоговое окно, содержащее:

– два поля ввода типа `TextBox` с метками `Size x`, `Size y`;

– кнопку типа `Button`.

При выборе команды `Choose` открывается диалоговое окно, содержащее:

– группу из двух переключателей (`Increase`, `Decrease`) типа `RadioButton`;

– кнопку типа `Button`.

Обеспечить возможность ввода значений в поля `Size x` и `Size y`. Значения интерпретируются как количество пикселов, на которое надо изменить размеры главного окна (увеличить или уменьшить в зависимости от положения переключателей).

После ввода значений команда `Change` становится доступной. При выборе этой команды размеры главного окна увеличиваются или уменьшаются на введенное количество пикселов.

Вариант 10

Создать меню с командами `Begin`, `Work`, `About`. При выборе команды `About` открывается окно с информацией о разработчике. При выборе команды `Begin` открывается диалоговое окно, содержащее:

– поле ввода типа `TextBox` с меткой `Input word`;

– группу из двух переключателей (`Upper case`, `Lower case`) типа `RadioButton`;

– кнопку типа `Button`.

Обеспечить возможность ввода слова и выбора режима перевода в верхний или нижний регистр в зависимости от положения переключателей. При выборе команды `Work` открывается диалоговое окно с результатом перевода.

Вариант 11

Создать меню с командами `Input color`, `Change`, `Clear`. При выборе команды Input color открывается диалоговое окно, содержащее:

– группу из двух флажков (`Up`, `Down`) типа `CheckBox`;

– группу из трех переключателей (`Red`, `Green`, `Blue`) типа `RadioButton`;

– кнопку типа `Button`.

Обеспечить возможность:

– выбора цвета с помощью переключателей;

– ввода режима, определяющего, какая область закрашивается: все окно, его верхняя или нижняя половина.

При выборе команды `Change` цвет главного окна изменяется на заданный (верхняя, нижняя или обе половины в зависимости от введенного режима). При выборе команды `Clear` восстанавливается первоначальный цвет окна.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Вариант 12

Создать меню с командами `Translate`, `Help`, `About`, `Exit`. При выборе команды `Exit` приложение завершает работу. При выборе команды `Translate` открывается диалоговое окно, содержащее:

– поле ввода типа `TextBox` с меткой `Binary number`;

– поле ввода типа `TextBox` для вывода результата (`read-only`);

– группу из трех переключателей (8, 10, 16) типа `RadioButton`;

– кнопку `Do` типа `Button`.

Обеспечить возможность:

– ввода числа в двоичной системе в поле `Binary number`;

– выбора режима преобразования с помощью переключателей: перевод в вось-меричную, десятичную или шестнадцатеричную систему счисления.

При щелчке на кнопку `Do` должен появляться результат перевода.

Вариант 13

Создать меню с командами `Reverse`, `About`, `Exit`.

При выборе команды `About` открывается окно с информацией о разработчике. При выборе команды `Reverse` открывается диалоговое окно, содержащее:

– поле ввода типа `TextBox` с меткой `Input`;

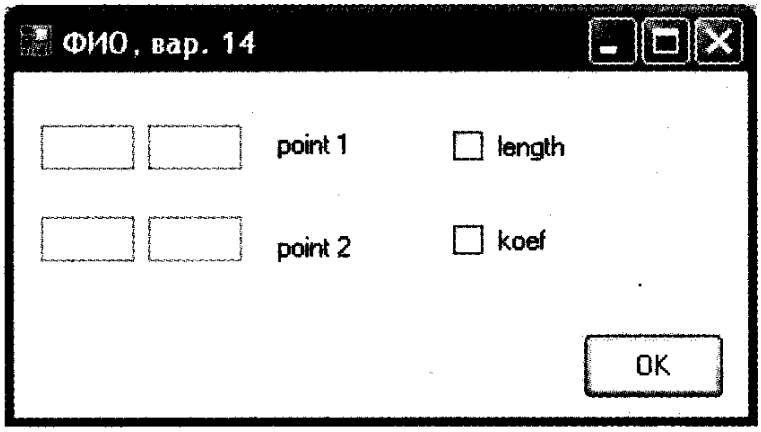
– группу из двух переключателей (`Upper case`, `Reverse`) типа `CheckBox`;

– кнопку `OK` типа `Button`.

Обеспечить возможность ввода фразы и выбора режима: перевод в верхний регистр и/или изменение порядка следования символов на обратный в зависимости от состояния переключателей. Результат преобразования выводится в исходное поле ввода.

Вариант 14

Создать меню с командами `Input`, `Show` и `Exit`. При выборе команды `Exit` приложение завершает работу. При выборе команды `Input` открывается диалоговое окно вида:



Обеспечить возможность ввода координат двух точек и выбора режима с помощью флажков `length` и `koef`: подсчет длины отрезка, соединяющего эти

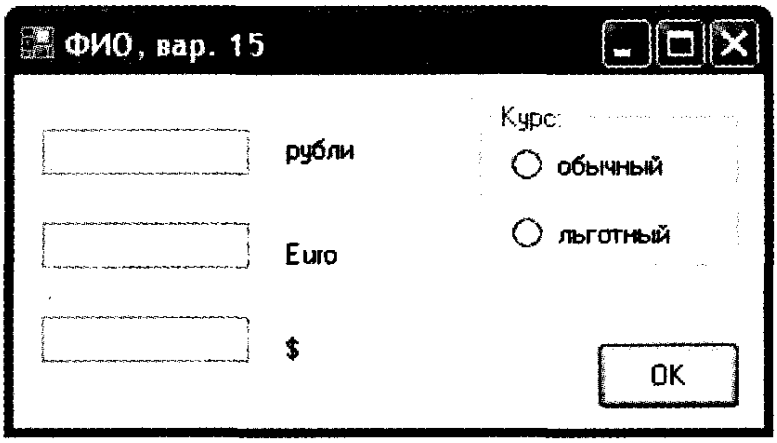
Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

точки, и/или углового коэффициента.

При выборе команды Show открывается окно сообщений с результатами подсчета.

Вариант 15

Создать меню с командами Input, About и Exit. При выборе команды Exit приложение завершает работу. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Input открывается диалоговое окно вида:



Обеспечивается возможность ввода суммы в рублях и перевода ее в евро и доллары по обычному или льготному курсу. Поля Euro и $ доступны только для чтения.

Контрольные вопросы

1 Что понимают под термином «событийно-управляемое программирование»?

2 Назовите основные структурные элементы Windows-приложения.

3 Опишите процесс создания Windows-приложения с графическим интерфейсом на основе Windows Forms.

4 Что такое обработчик события?

5 Как создаются обработчики событий?

6 Опишите синтаксис обработчика событий.

7 Какие параметры передаются в обработчик события?

8 Перечислите основные свойства стандартных визуальных элементов управления.

9 Каким образом можно добавить элемент управления на форму?

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

11 Лабораторная работа № 11. Программирование с использованием Windows Forms

Цель работы: ознакомиться с основой разработки `Windows`-приложений на языке `C#` с использованием `Windows Forms`.

11.1 Необходимые теоретические сведения

1 Класс `Form` [3, с. 337–339].
2 Диалоговые окна [3, с. 339–342].
3 Класс `Application` [3, с. 342–344].

11.2 Индивидуальные задания

Создать параметризованную коллекцию для хранения экземпляра класса, созданного при выполнении лабораторной работы № 1. Вид коллекции выбрать самостоятельно. Написать `Windows`-приложение для работы с этой коллекцией, позволяющее выполнять:

– добавление элемента в коллекцию с клавиатуры;
– считывание данных из файла;
– запись данных в тот же или указанный файл;
– сортировку данных по различным критериям;
– поиск элемента по заданному полю;
– вывод всех элементов, удовлетворяющих заданному условию;
– удаление элемента из коллекции.

Приложение должно содержать меню и диалоговые окна и предусматривать обработку возможных ошибок пользователя с помощью исключений.

Контрольные вопросы

1 Объясните назначение класса `Form`.
2 Назовите основные свойства, методы, события класса `Form`.
3 Чем диалоговое окно отличается от обычной формы?
4 Какие виды диалоговых окон Вам известны?
5 Объясните последовательность действий, которые надо совершить для отображения диалогового окна.
6 Для чего предназначен класс `Application`?
7 Перечислите основные элементы класса `Application`.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/

Список литературы

1 **Шилд, Г.** C# 4.0 : полное руководство : пер. с англ. / Г. Шилд. – Москва : И. Д. Вильямс, 2011. – 1056 с.

2 **Подбельский, В. В.** Язык Си#. Базовый курс : учебное пособие / В. В. Подбельский. – Москва: Финансы и статистика, 2011. – 384 с.

3 **Павловская, Т. А.** C#. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. – Санкт-Петербург : Питер, 2014. – 432 с.

4 C# 5.0 и платформа .NET 4.5 для профессионалов : пер. с англ. / К. Нейгел [и др.]. – Москва : И. Д. Вильямс, 2014. – 1440 с.

5 **Зиборов, В. В.** Visual C# 2012 на примерах / В. В. Зиборов. – Санкт-Петербург : БХВ-Петербург, 2013. – 480 с.

6 Объектно-ориентированное программирование и проектирование : методические рекомендации к лабораторным работам для студентов специальности 1-53 01 02 «Автоматизированные системы обработки информации» дневной формы обучения / Сост. Н. К. Борисов. – Могилёв : Белорус.-Рос. ун-т, 2016. – Ч. 1. – 46 с.

7 Объектно-ориентированное программирование : лабораторный практикум / Сост. А. В. Щербаков. – Минск : БНТУ, 2014. – 38 с.

Электронная библиотека Белорусско-Российского университета
http://e.biblio.bru.by/