

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
Воронежский государственный технический университет
(ФГБОУ ВО «ВГТУ», ВГТУ)

УТВЕРЖДАЮ

Декан факультета

«Экономики, менеджмента и
информационных технологий»

С.А. Баркалов

« 04 » сентября 2017 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Объектно-ориентированное программирование»

Направление подготовки (специальность) 09.03.02 «Информационные
системы и технологии»

Профиль Информационные системы и технологии в строительстве

Квалификация (степень) выпускника

бакалавр

Нормативный срок обучения

4 года

Форма обучения

очная

Автор программы



канд. техн. наук, доцент Сысоев Д.В.

Программа обсуждена на заседании кафедры «Информационных технологий
и автоматизированного проектирования в строительстве»

«31» августа 2017 года

Протокол № 1

Зав. кафедрой



А.В. Смольянинов

Воронеж 2017

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цель дисциплины

Целью данной дисциплины является необходимость формирования у обучающихся понимания идеологии и ключевых аспектов объектно-ориентированного программирования (ООП) на языке C#, достаточного для практического использования в процессе дальнейшего обучения и в профессиональной сфере.

1.2. Задачи освоения дисциплины:

Задачей изучения дисциплины является изучение современных объектно-ориентированных подходов и технологий разработки ПО.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Дисциплина «Объектно-ориентированное программирование» относится к вариативной части обязательных дисциплин блока «Дисциплины (модули)» учебного плана.

Данная учебная дисциплина взаимосвязана с другими дисциплинами, а именно базируется на дисциплинах: «Основы программной реализации информационных технологий», «Основы программирования и алгоритмизации», «Теоретические основы информатики и численные методы».

Курс базируется на самых различных отраслях знаний и научных выводах математики и информатики. Изучением дисциплины достигается формирование у специалистов представления о взаимосвязи основных нечетких моделей с алгоритмами и методами реализации программ в современных информационных системах.

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины «Объектно-ориентированное программирование» направлен на формирование следующих компетенций:

- владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (ОПК-1);
- способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи (ОПК-6);
- способностью проводить рабочее проектирование (ПК-3);
- способностью обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений (ПК-24).

В результате изучения дисциплины студент должен:

Знать:

основные принципы объектно-ориентированного программирования и их приложения.

Уметь:

разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п.

Владеть:

методикой разработки объектно-ориентированных программ.

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины «Объектно-ориентированное программирование» составляет 8 зачетных единиц.

Вид учебной работы	Всего часов	Семестры	
		4	5
Аудиторные занятия (всего)	104/-	68/-	36/-
В том числе:			
Лекции	52/-	34/-	18/-
Практические занятия (ПЗ)	/-	-/-	/-
Лабораторные работы (ЛР)	52/-	34/-	18/-
Самостоятельная работа (всего)	148/-	40/-	108/-
В том числе:			
Курсовая работа			36
Подготовка к лабораторным работам		36	36
Контрольная работа			
Вид промежуточной аттестации (зачет, экзамен)	зачет, экз, 36	зачет	экз 36
Общая трудоемкость час зач. ед.	288/-	108/-	180/-
	8/-	3/-	5/-

Примечание: здесь и далее числитель – очная/знаменатель – заочная формы обучения.

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1. Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1	Введение в объектно-ориентированное программирование	Объектно-ориентированное программирование. Классы
2	Основные понятия языка С#	Состав языка. Типы данных
3	Переменные, операции и выражения	Переменные. Именованные константы. Операции и выражения. Линейные программы.
4	Операторы	Выражения, блоки и пустые операторы. Операторы ветвления. Операторы цикла. Обработка исключительных ситуаций. Операторы checked и unchecked.
5	Классы: основные понятия	Присваивание и сравнение объектов. Данные: поля и константы. Методы. Ключевое слово this. Конструкторы. Свойства.

№ п/п	Наименование раздела дисциплины	Содержание раздела
6	Массивы и строки	Массивы. Оператор foreach. Массивы объектов. Символы и строки. Класс Random.
7	Классы: подробности	Перезагрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Деструкторы. Вложенные типы.
8	Иерархии классов	Наследование. Виртуальные методы. Абстрактные классы. Бесплодные классы. Класс object.
9	Интерфейсы и структурные типы	Синтаксис интерфейса. Реализация интерфейса. Работа с объектами через интерфейсы. Операции is и as. Интерфейсы и наследование. Стандартные интерфейсы .NET. Структуры. Перечисления.
10	Делегаты, события и потоки выполнения	Делегаты. События. Многопоточные приложения
11	Работа с файлами	Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).
12	Сборки, библиотеки, атрибуты, директивы	Сборки. Создание библиотеки. Использование библиотеки. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.
13	Структуры данных, коллекции и классы-прототипы	Абстрактные структуры данных. Пространство имен System.Collections. Классы-прототипы. Чистичные типы. Обнуляемые типы.
14	Введение в программирование под Windows	Событийно-управляемое программирование. Шаблон Windows-приложения. Класс Control. Элементы управления. Замечания о формах. Класс Form. Диалоговые окна. Класс Application. Введение в графику.
15	Дополнительные средства C#	Небезопасный код. Регулярные выражения. Документирование в формате XML.

5.2. Разделы дисциплины и междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами

№ п/ п	Наименование обеспечиваемых (последующих) дисциплин	№ № разделов данной дисциплины, необходимых для изучения обеспечиваемых (последующих) дисциплин														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Технологии программирования	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2	Методы и средства проектирования ИС и Т	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

5.3. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	СРС	Всего час.
1.	Введение в объектно-ориентированное программирование	2			6	8
2.	Основные понятия языка C#	2		1	8	11
3.	Переменные, операции и выражения	2		1	8	11
4.	Операторы	2		8	10	20
5.	Классы: основные понятия	2		2	10	14
6.	Массивы и строки	4		12	12	28
7.	Классы: подробности	6		6	12	24
8.	Иерархии классов	4		4	10	18
9.	Интерфейсы и структурные типы	6		8	10	24
10.	Делегаты, события и потоки выполнения	4			10	14
11.	Работа с файлами	4			12	16
12.	Сборки, библиотеки, атрибуты, директивы	4			10	14
13.	Структуры данных, коллекции и классы-прототипы	4		4	10	18
14.	Введение в программирование под Windows	4		6	10	20
15.	Дополнительные средства C#	2			10	12

5.4. Лабораторный практикум

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час)
1.	2, 3	Линейные программы	2
2.	4	Разветвляющиеся вычислительные процессы	2
3.	4	Организация циклов	4
4.	4, 5	Простейшие классы	4
5.	6	Одномерные массивы	4
6.	6	Двумерные массивы	4
7.	6	Строки	4
8.	7	Классы и операции	6
9.	8	Наследование	4
10.	9	Структуры	6
11.	9, 13	Интерфейсы и параметризованные коллекции	6
12.	14	Создание Windows-приложений	6

5.5. Практические занятия

Практические занятия не предусмотрены учебным планом

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ И КОНТРОЛЬНЫХ РАБОТ

КОМПЛЕКТ №1 - КЛАССЫ

Тема 1. Разработка класса для представления множества целых чисел на основе связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 2. Разработка класса для представления многочлена от одной переменной на основе связанного списка. Операции: сложение многочленов, умножение многочленов, деление многочленов, дифференцирование, интегрирование.

Тема 3. Разработка класса для представления упорядоченного множества вещественных чисел на основе циклического связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 4. Разработка класса для представления многочлена от одной переменной на основе двунаправленного связанного списка. Операции: сложение многочленов, умножение многочленов, деление многочленов, дифференцирование, интегрирование.

Тема 5. Разработка класса для представления упорядоченного множества строк на основе бинарного дерева. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

КОМПЛЕКТ №2 - Использование стандартной библиотеки шаблонов (STL)

Тема 1. Программа моделирования Т-образного сортировочного узла на железной дороге с использованием контейнерного класса `stack` из STL.

Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность ввода исходных данных с клавиатуры и из файла.

Тема 2. Программа моделирования работы автобусного парка.

Сведения о каждом автобусе содержат: номер автобуса, фамилию и инициалы водителя, номер маршрута.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла);
- имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;
- имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;
- вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте.

Для представления необходимых списков использовать контейнерный класс `list`.

Тема 3. Написать программу учета заявок на авиабилеты.

Каждая заявка содержит: пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- добавление заявок в список;
- удаление заявок;
- вывод заявок по заданному номеру рейса и дате вылета;
- вывод всех заявок.

Для хранения данных использовать контейнерный класс `list`.

Тема 4. Написать программу учета книг в библиотеке.

Сведения о книгах содержат: фамилию и инициалы автора, название, год издания, количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- добавление данных о книгах, вновь поступающих в библиотеку;
- удаление данных о списываемых книгах;
- выдача сведений о всех книгах, упорядоченных по фамилиям авторов;
- выдача сведений о всех книгах, упорядоченных по годам издания.

Хранение данных организовать с применением контейнерного класса `multimap`, в качестве ключа использовать «фамилию и инициалы автора».

Тема 5. Программа «Моя записная книжка».

Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям.

Хранение данных организовать с применением контейнерного класса `map` или `multimap`.

КОМПЛЕКТ №3 – Игровые программы

Тема 1. Игра Шахматы. Реализовать класс «Игровое поле», иерархию «Фигура»-«Конкретная фигура». Реализовать метод вывода фигуры на экран. Класс «Фигура» должен содержать виртуальную функцию проверки допустимости указанного хода, а классы для конкретных фигур – ее реализации с учетом правил выполнения ходов для каждого вида фигуры. Для класса «Игровое поле» определить методы запроса хода (поочередно для противников), выполнения хода, «съедания» фигуры противника, вывода сообщения о нахождении фигуры под боем, шахе и мате. Предусмотреть возможность сохранения и загрузки игры с помощью ввода/вывода в поток.

Тема 2. Логическая игра «Крестики-нолики»

Приложение является реализацией известной логической игры «Крестики-нолики». В данной реализации предусматривается игра двух игроков на неограниченном поле. Цель игры: построить непрерывную линию из пяти или более фишек (крестиков или ноликов) по горизонтали, вертикали или диагонали. Возможны два режима:

игры, когда выигрывает тот, кто первым построит линию, или кто больше наберет очков за определенное количество времени.

Во втором случае количество построенных линий неограничено, каждая фишка в линии приносит игроку одно очко.

Приложение должно обеспечивать начало новой игры на чистом поле, а также проверку соответствия действий игроков правилам игры и условия окончания игры. Роль одного из игроков (по выбору пользователя) может выполнять компьютер.

Для исключения возможности образования изолированных игр на одном поле и неоправданного увеличения размера поля следует установить максимально допустимое его состояние (не более пяти) от существующих фишек доновой, размещаемой игроком.

Тема 3. Логическая игра “Цепь”.

В данной реализации предусматривается игра двух игроков на квадратном поле фиксированного размера. Цель игры - построить непрерывную линию, соединяющую горизонтальные или вертикальные (для каждого из игроков соответственно) стороны игрового поля, причем линия считается непрерывной, если фишки граничат друг с другом по горизонтали, вертикали или диагонали. Выигрывает тот игрок, который первым построит свою линию.

Приложение должно обеспечивать начало новой игры на чистом поле, а также проверку соответствия действий игроков правилам игры и условия окончания игры. Роль одного из игроков (по выбору пользователя) может выполнять компьютер.

Тема 4. Логическая игра “Точки”

В данной реализации предусматривается игра двух игроков на квадратном поле фиксированного размера. Цель игры - заполнить максимальное количество клеток игрового поля своими фишками. Игроки выполняют ходы по очереди.

Ход игрока заключается в произвольной установке линии на границе двух ячеек, причем, если какая-либо ячейка оказывается со всех четырех сторон обрамлена линиями, то она отмечается фишкой данного игрока,

а самому игроку предоставляется дополнительный ход, и т.д. Игра заканчивается, когда все игровое поле оказывается заполнено фишками. Выигрывает тот игрок, чья фишек на поле больше на момент окончания игры.

Приложение должно обеспечивать начало новой игры на чистом поле, а также проверку соответствия действий игроков правилам игры и условия окончания игры. Роль одного из игроков (по выбору пользователя) может выполнять компьютер.

Тема 5. Логическая игра “Мозаика”

Разрабатываемое приложение представляет собой программную реализацию популярной игры Мозаика (Puzzle), цель которой заключается в создании игроком полного изображения путем упорядочивания его фрагментов последовательным перемещением и соединением соответствующих сторон. Приложение должно обеспечивать:

просмотр исходного изображения во время игры, а также поиск подходящих друг к другу фрагментов (функция подсказки); возможность перемещать фрагменты за пределы рабочего поля или в другое окно и наоборот для улучшения наглядности.

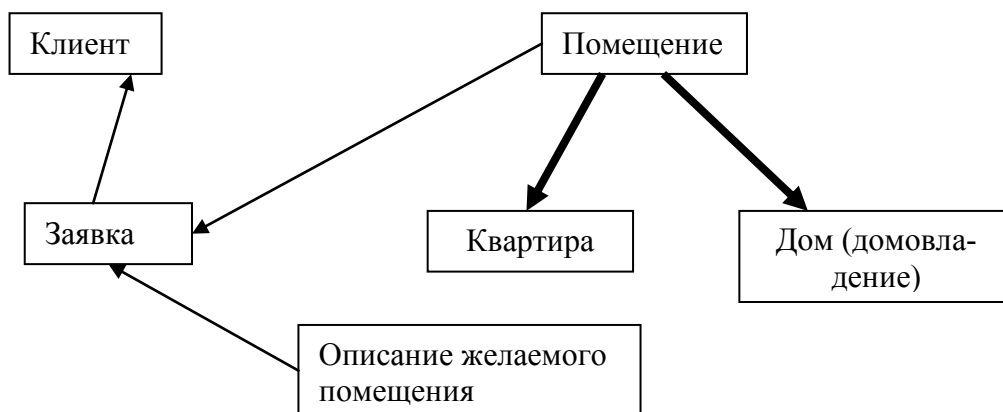
Следует учесть, что при создании новой игры все фрагменты должны быть разьединены и размещены беспорядочно в области рабочего поля или в области, отведенной специально для фрагментов.

При использовании отдельной области для отображения фрагментов может быть применена ее прокрутка и изменение масштаба.

КОМПЛЕКТ №4

Тема 1. Информационная система риэлтерской конторы, занимающейся продажей и обменом жилья.

Использовать следующую примерную объектную модель:

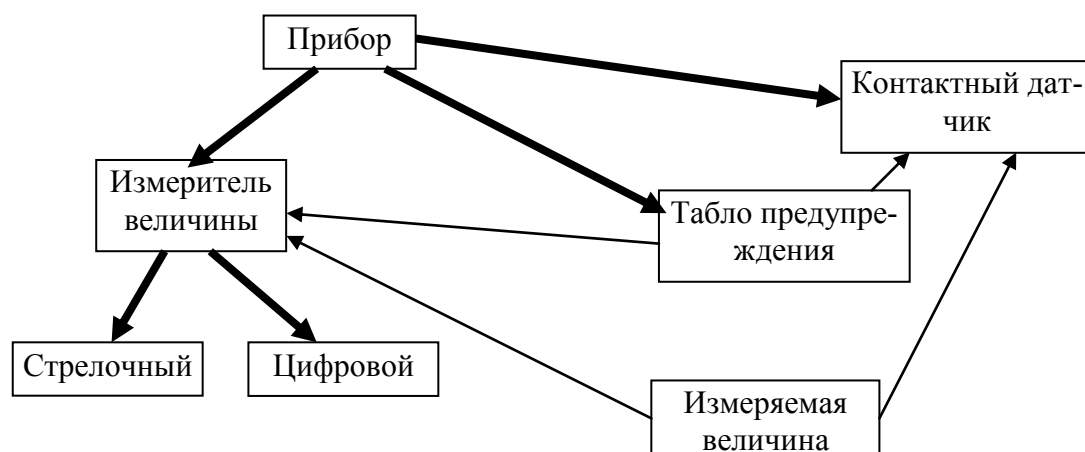


В заявке содержатся: вид операции (продажа, покупка, обмен). Для продажи и обмена указывается имеющееся у клиента дом/квартира. Помещение описывается следующими параметрами: адрес, площадь, кол-во комнат, стоимость, наличие удобств. Для покупки и обмена к заявлению прилагается также описание желаемого помещения, где указывается, например: диапазон для цены, кол-ва комнат, требования к удобствам, требования к району размещения, и.т.п.

Реализовать функции добавления клиентов и заявок, подбора Тематик по заявкам, сохранения/загрузки объектов в файл с использованием перегрузки операторов ввода/вывода в поток.

Тема 2. Виртуальная приборная панель.

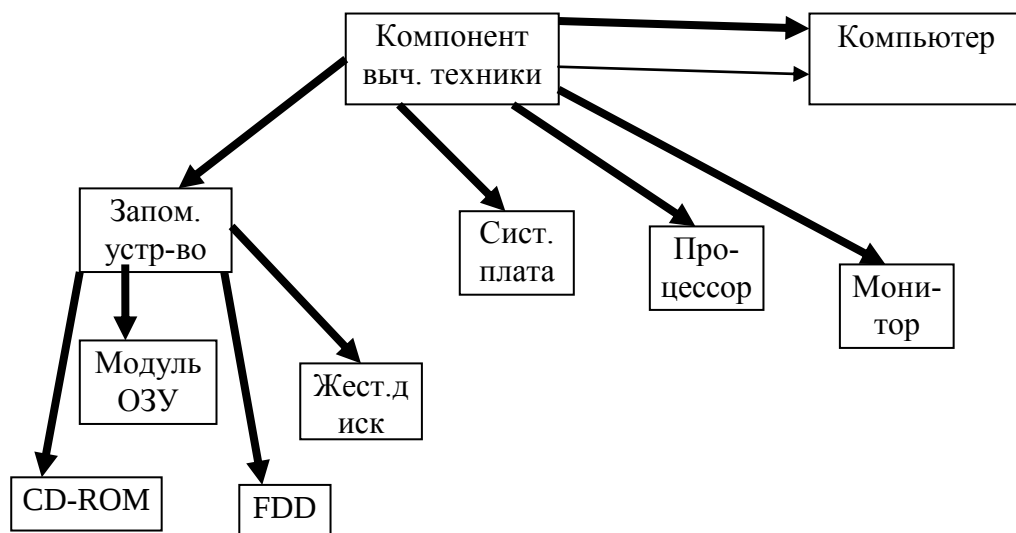
Разработать объектную модель для комплекса приборов наблюдения за какими-либо параметрами:



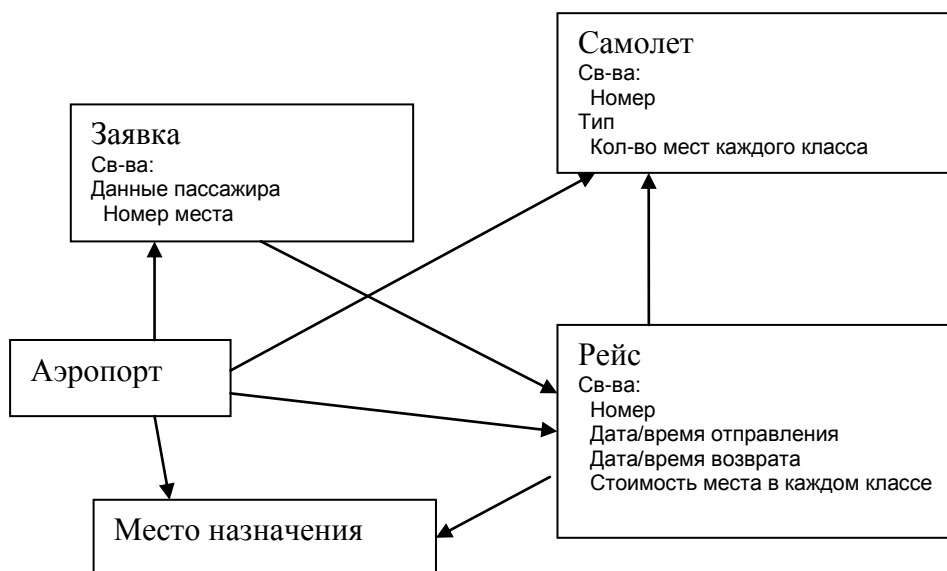
Для каждого класса приборов реализовать методы визуализации, перемещения по рабочему полю (экрану); задать набор параметров, определяющих вид прибора (напр. размер циферблата, пределы измерения, вид шрифта для цифр/текста). Реализовать методы изменения измеряемой величины. Для табло предупреждения реализовать метод, позволяющий задавать несколько диапазонов измеряемой величины и соответствующие им сообщения и вид сообщения (например: информация, предупреждение, тревога). Обеспечить для класса «Прибор» возможность периодического чтения измеряемого показателя из внешнего источника (напр. из файла). С помощью созданных классов смоделировать приборный щиток какого-либо агрегата (напр. приборная панель автомобиля, контрольные приборы газового котла и т.п.)

Тема 3. Справочная система компонентов компьютерной техники. Создать справочную систему, которая позволяла бы хранить сведения о компьютерах организации и об их составляющих (компонентах). Использовать следующую примерную иерархию классов.

Система должна позволять хранить информацию, как об отдельных компонентах, так и о компонентах в составе компьютеров. Реализовать поиск компонентов по: виду, инвентарному номеру, названию производителя, признаку «свободен/находится в составе компьютера», по компонентам, принадлежащим одному компьютеру. Реализовать методы включения свободного компонента в состав компьютера; передачи компонента от одного компьютера к другому; изъятия компонента из состава компьютера и включения в состав свободных компонентов. Реализовать выгрузку в файл/ загрузку из файла всех объектов и связей между ними с помощью перегрузки операторов ввода/вывода в поток.



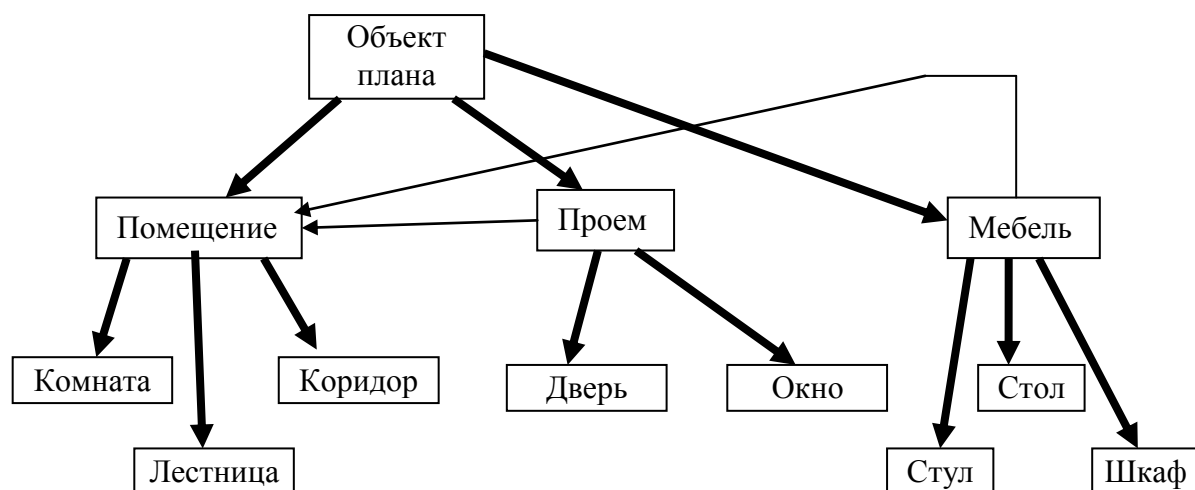
Тема 4. Информационная система «Аэропорт». Использовать приблизительно следующую иерархию:



Реализовать: назначение самолета на рейс с контролем того, чтобы он успел вернуться из предыдущего рейса и провел некоторое время на техническое обслуживание; поиск свободного места на рейсах по месту назначения, диапазону даты и времени отправления, диапазону класса места и цены. Реализовать сохранение/загрузку состояния базы в файл с использованием перегрузки операторов ввода/вывода в поток для классов.

Тема 5. Планировщик помещений.

Разработать визуальный конструктор (редактор), позволяющий создавать планы помещений. Использовать приблизительно следующую иерархию классов для объектов, из которых может состоять план:



Реализовать методы, с помощью которых пользователь может перемещать объекты на плане, изменять их размеры. Реализовать функцию сохранения/загрузки плана в файл с помощью перегрузки операторов ввода/вывода в поток для созданных классов.

7. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Компетенция (общекультурная – ОК; общепрофессиональная – ОПК; профессиональная – ПК)	Форма контроля	Семестр
1.	владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (ОПК-1)	Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз)	4, 5
2.	способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи (ОПК-6)	Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз)	4, 5
3.	способностью проводить рабочее проектирование (ПК-3)	Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз)	4, 5
4.	способностью обосновывать правильность	Индивидуальный опрос (ИО),	4, 5

	выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений (ПК-24)	защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз)	
--	--	--	--

7.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Дескриптор компетенции	Показатель оценивания	Форма контроля					
		ИО	ЗЛР	КР	Т	КурсП	Экз
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	+	+	+	+	+	+
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)	+	+	+	+	+	+
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)	+	+	+	+	+	+

7.2.1. Этап текущего контроля знаний

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по пятибалльной шкале с оценками:

- «отлично»;
- «хорошо»;
- «удовлетворительно»;
- «неудовлетворительно»;
- «не аттестован».

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	отлично	Полное или частичное посещение лекционных, лабораторных занятий. Защита лабораторных работ. Выполненные КР и курсовая работа на оценки «отлично».
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	хорошо	Полное или частичное посещение лекционных, лабораторных занятий. Защита лабораторных работ. Выпол-
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		ненные КР и курсовая работа на оценки «хорошо».
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	удовлетворительно	Полное или частичное посещение лекционных, лабораторных занятий. Защита лабораторных работ. Выполненные КР и курсовая работа на оценки «удовлетворительно».
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	неудовлетворительно	Частичное посещение лекционных, лабораторных занятий. Защита лабораторных работ. Выполненные КР и курсовая работа на оценки «неудовлетворительно».
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	не аттестован	Непосещение лекционных, лабораторных занятий. Нет защищенных лабораторных работ. Невыполненные КР и курсовая работа.
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		

7.2.2. Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний (зачет в четвертом семестре) оцениваются по двухбальной шкале:

- «зачтено»;
- «не зачтено».

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1,	зачтено	Студент демонстрирует полное понимание заданий. Все требования, предъявляемые к

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
	ОПК-6, ПК-3, ПК-24)		
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		заданию выполнены Студент демонстрирует значительное понимание заданий. Все требования, предъявляемые к заданию выполнены Студент демонстрирует частичное понимание заданий. Большинство требований, предъявляемых к заданию выполнены
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)	Не зачтено	1. Студент демонстрирует небольшое понимание заданий. Многие требования, предъявляемые к заданию не выполнены. 2. Студент демонстрирует непонимание заданий. 3. У студента нет ответа. Не было попытки выполнить задание
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		

Результаты промежуточного контроля знаний (экзамен в пятом семестре) оцениваются по четырехбалльной шкале с оценками:

- «отлично»;
- «хорошо»;
- «удовлетворительно»;
- «неудовлетворительно».

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)	отлично	Студент демонстрирует полное понимание заданий. Все требования, предъявляемые к заданию выполнены
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Умеет	разрабатывать программы на языке программирования С#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)	хорошо	Студент демонстрирует значительное понимание заданий. Все требования, предъявляемые к заданию выполнены
Владеет	методикой разработки объектно-		

Дескриптор компетенции	Показатель оценивания	Оценка	Критерий оценивания
	ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	удовлетворительно	Студент демонстрирует частичное понимание заданий. Большинство требований, предъявляемых к заданию выполнены
Умеет	разрабатывать программы на языке программирования C#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Знает	основные принципы объектно-ориентированного программирования и их приложения (ОПК-1, ОПК-6, ПК-3, ПК-24)	неудовлетворительно	1. Студент демонстрирует небольшое понимание заданий. Многие требования, предъявляемые к заданию не выполнены. 2. Студент демонстрирует непонимание заданий. 3. У студента нет ответа. Не было попытки выполнить задание
Умеет	разрабатывать программы на языке программирования C#, с использованием классов, их взаимодействия и т.п. (ОПК-1, ОПК-6, ПК-3, ПК-24)		
Владеет	методикой разработки объектно-ориентированных программ (ОПК-1, ОПК-6, ПК-3, ПК-24)		

7.3. Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

Текущий контроль успеваемости осуществляется на лабораторных занятиях: в виде опроса теоретического материала и самостоятельного выполнения лабораторных заданий под контролем преподавателя, в виде тестирования по отдельным темам.

Промежуточный контроль осуществляется проведением контрольных работ по отдельным разделам дисциплины, тестирования по разделам дисциплины, изученным студентом в период между аттестациями. Контрольные работы проводятся на лабораторных занятиях в рамках самостоятельной работы под контролем преподавателя. Варианты контрольных и курсовых работ выдаются каждому студенту индивидуально.

7.3.1. Примерная тематика и содержание КР

1. Введение в объектно-ориентированное программирование. Объектно-ориентированное программирование. Классы
2. Основные понятия языка C#. Состав языка. Типы данных
3. Переменные, операции и выражения. Переменные. Именованные константы. Операции и выражения. Линейные программы.
4. Операторы. Выражения, блоки и пустые операторы. Операторы ветвления. Операторы цикла. Обработка исключительных ситуаций. Операторы checked и unchecked.
5. Классы: основные понятия. Присваивание и сравнение объектов. Данные: поля и константы. Методы. Ключевое слово this. Конструкторы. Свойства.
6. Массивы и строки. Массивы. Оператор foreach. Массивы объектов. Символы и строки. Класс Random.
7. Классы: подробности. Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Деструкторы. Вложенные типы.
8. Иерархии классов. Наследование. Виртуальные методы. Абстрактные классы. Бесплодные классы. Класс object.
9. Интерфейсы и структурные типы. Синтаксис интерфейса. Реализация интерфейса. Работа с объектами через интерфейсы. Операции is и as. Интерфейсы и наследование. Стандартные интерфейсы .NET. Структуры. Перечисления.
10. Делегаты, события и потоки выполнения. Делегаты. События. Многопоточные приложения
11. Работа с файлами. Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).
12. Сборки, библиотеки, атрибуты, директивы. Сборки. Создание библиотеки. Использование библиотеки. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.
13. Структуры данных, коллекции и классы-прототипы. Абстрактные структуры данных. Пространство имен System.Collections. Классы-прототипы. Чистичные типы. Обнуляемые типы.
14. Введение в программирование под Windows. Событийно-управляемое программирование. Шаблон Windows-приложения. Класс Control. Элементы управления. Замечания о формах. Класс Form. Диалоговые окна. Класс Application. Введение в графику.
15. Дополнительные средства C#. Небезопасный код. Регулярные выражения. Документирование в формате XML.

7.3.2. Примерные задания для тестирования

1. Можно ли в C# в разных файлах проекта использовать одинаковые имена переменных?
 - а. Да

- b. Только в разных пространствах имен
 - c. Нет
2. Что возвратит результат выражения `1 isObject`?
- a. Произойдет ошибка компиляции
 - b. `True`
 - c. `False`
3. Как правильно описать заголовок функции `Swap`, меняющей местами значения двух целых переменных?
- a. `void Swap(int a, int b);`
 - b. `void Swap(ref int a, ref int b);`
 - c. `void Swap(out int a, out int b);`
4. Какой модификатор по умолчанию используется для классов?
- a. `public`
 - b. `private`
 - c. `internal`
5. Какой модификатор по умолчанию используется для членов?
- a. `public`
 - b. `private`
 - c. `internal`
 - d. `protected`
6. При наследовании конструктор базового класса вызывается
- a. До конструктора производного класса
 - b. Может вообще не вызываться, если мы его явно не вызовем
 - c. Вызывается в тот момент, когда мы явно укажем его вызов в коде конструктора подкласса
7. Сколько интерфейсов может реализовывать класс?
- a. Не более одного
 - b. Множество
 - c. Множество, но в этом случае он не может наследоваться от другого класса
8. Если в классе `MyClass` метод интерфейса `MyInt {void Menu();}` реализован внешним и внутренним образом, то при вызове `MyClass m = new MyClass(); ((MyInt)m).Menu();`
- a. Вызовется внешняя реализация интерфейса
 - b. Вызовется внутренняя реализация интерфейса
 - c. Произойдет ошибка компиляции
9. Что произойдет при XML сериализации объекта `widget`?
- ```
public interface IWidget
```

```

{
 int Index
 { get;
 set; }
}
[Serializable]
public class Widget : IWidget
{
 private int index;
 public int Index
 { get { return index; }
 set { index = value; }
 }
}
// Метод Main()
IWidget widget = new Widget();
widget.Index = 10;
using (StreamWriter writer = new StreamWriter("data.xml"))
{
 XmlSerializer serializer = new XmlSerializer(typeof(IWidget));
 serializer.Serialize(writer, widget);
}

```

- a. Ошибка времени исполнения: класс Widget должен быть помечен атрибутом XmlSerializable, а не Serializable
- b. Ошибка времени компиляции: для сериализации нужно использовать не StreamWriter, а только XmlWriter
- c. Объект widget будет успешно сериализован
- d. Ошибка времени исполнения: интерфейс IWidget должен быть помечен атрибутом Serializable
- e. Будет вызвано исключение NotSupportedException с сообщением, что нельзя сериализовать интерфейс IWidget

10. Выберите средства, которые предоставляет C# для условной компиляции.

- a. Директива #endif
- b. Директива #typedef
- c. Директива #else
- d. Директива #elseif
- e. Атрибут Conditional
- f. Директива #define
- g. Директива #if

11. Чтобы вывести на экран число типа float 10.56f с количеством цифр после запятой равным 3 и шириной равной 10 символам нужно написать:

- a. Console.WriteLine("{3:10F0}", 10.56f);
- b. Console.WriteLine("{10:0:3F}", 10.56f);
- c. Console.WriteLine("{0, 3F:10}", 10.56f);
- d. Console.WriteLine("{0, 10:F3}", 10.56f);

12. Чем отличаются константы и доступные только для чтения поля?

- a. Ничем не отличаются

- b. Доступные только для чтения поля инициализируются во время компиляции, константы - во время выполнения
- c. Константы можно изменять, а доступные только для чтения поля нет
- d. Константы инициализируются во время компиляции, доступные только для чтения поля - во время выполнения

13. Выберите операторы, которые необходимо переопределять попарно.

- a. и <
- b. != и ==
- c. ++ и --
- и /
- d. >= и <=
- e. + и -

14. Какие операторы нужно переопределить у класса Digit, чтобы были возможными преобразования:

```
Digit digit = new Digit(100);
decimal decimalDigit = digit;
byte byteDigit = (byte)digit;
```

- a. public static implicit operator byte(Digit digit) { }
- b. public static explicit operator byte(Digit digit) { }
- c. Нужные операторы не приведены
- d. public static implicit operator decimal(Digit digit) { }

15. Когда вызываются статические конструкторы классов в C#?

- a. После каждого обращения к статическим полям, методам и свойствам
- b. Строгий порядок вызова не определен
- c. Статических конструкторов в C# нет
- d. Один раз при первом создании экземпляра класса или при первом обращении к статическим членам класса

16. По какой причине данный код не скомпилируется?

```
public delegate void EventHandler();
public class Control
{
 public event EventHandler Invalidate;
}
public class Button : Control
{
 public void Draw()
 {
 if (Invalidate != null)
 {
 Invalidate();
 }
 }
}
```

- a. Вызывать событие напрямую может только класс, в котором они объявлены
- b. Нельзя сравнивать событие с null
- c. Событие всегда равно null
- d. У события отсутствуют обязательные части add { } и remove { }

17. Выберите правильный вариант переопределенного метода GetHashCode() у класса Point (будем считать, что Equals() уже переопределено правильно)

```
public class Point
{
```

```
 public int x;
 public int y;
}
```

- `public override int GetHashCode() { return x | y; }`
- `public override int GetHashCode() { return x * y; }`
- Нет правильного ответа
- `public override int GetHashCode() { return x & y; }`
- `public override int GetHashCode() { return x ^ y; }`

18. Что произойдет при Бинарной сериализации объекта widget?

```
public interface IWidget
{
```

```
 int Index
 {
 get;
 set;
 }
}
```

```
[Serializable]
```

```
public class Widget : IWidget
{
```

```
 private int index;
 public int Index
 {
 get { return index; }
 set { index = value; }
 }
}
```

```
// Метод Main()
```

```
IWidget widget = new Widget();
```

```
widget.Index = 10;
```

```
using (FileStream writer = new FileStream("data.dat", FileMode.Truncate))
```

```
{
```

```
 BinaryFormatter serializer = new BinaryFormatter ();
 serializer.Serialize(writer, widget);
}
```

- Ошибка времени компиляции: для сериализации нужно использовать не StreamWriter, а только BinaryWriter
- Ошибка времени исполнения: интерфейс IWidget должен быть помечен атрибутом Serializable
- Ошибка времени исполнения: класс Widget должен быть помечен атрибутом BinarySerializable, а не Serializable
- Будет вызвано исключение NotSupportedException с сообщением, что нельзя сериализовать интерфейс IWidget
- Объект widget будет успешно сериализован

19. Чем опасны подобные конструкции?

```
public class Widget
```

```
{
```

```
 private int width;
 public int IncrementWidth
 {
```

```

 {
 get { return ++width; }
 }
 }
}
class Program
{
 static void Main(string[] args)
 {
 Widget widget = new Widget();
 int width = widget.IncrementWidth;
 }
}

```

- a. Неправильно сделан инкремент поля width, нужно ++width заменить на width++
- b. Ничем, код полностью корректен
- c. При отладке кода в режиме Debug при просмотре свойства IncrementWidth объекта widget будет увеличиваться значение width, что повлечет сложности при отладке
- d. Нет блока set {} в свойстве IncrementWidth

20. Что произойдет при компиляции проекта, где используется класс, структура, интерфейс или перечисление, помеченное атрибутом Obsolete?

- a. Что произойдет при компиляции проекта, где используется класс, структура, интерфейс или перечисление, помеченное атрибутом Obsolete?
- b. Будет выведено предупреждение о том, что данный тип устарел, но сборка будет создана
- c. Сборка будет создана, но при запуске произойдет ошибка времени выполнения
- d. Атрибут Obsolete никак не влияет на компиляцию
- e. Нет нужного варианта ответа

### 7.3.2. Примерный перечень вопросов к зачетам и экзаменам

#### Зачет

1. Объектно-ориентированное программирование.
2. Классы
3. Состав языка.
4. Типы данных
5. Переменные.
6. Именованные константы.
7. Операции и выражения.
8. Линейные программы.
9. Выражения, блоки и пустые операторы.
10. Операторы ветвления.
11. Операторы цикла.
12. Обработка исключительных ситуаций.
13. Операторы checked и unchecked.
14. Присваивание и сравнение объектов.
15. Данные: поля и константы.

16. Методы.
17. Ключевое слово `this`.
18. Конструкторы.
19. Свойства.
20. Массивы.
21. Оператор `foreach`.
22. Массивы объектов.
23. Символы и строки.
24. Класс `Random`.
25. Перегрузка методов.
26. Рекурсивные методы.
27. Методы с переменным количеством аргументов.
28. Метод `Main`.
29. Индексаторы.
30. Операции класса.
31. Деструкторы.
32. Вложенные типы.
33. Наследование.
34. Виртуальные методы.
35. Абстрактные классы.
36. Бесплодные классы.
37. Класс `object`.
38. Синтаксис интерфейса.
39. Реализация интерфейса.
40. Работа с объектами через интерфейсы.
41. Операции `is` и `as`.
42. Интерфейсы и наследование.
43. Стандартные интерфейсы `.NET`.
44. Структуры.
45. Перечисления.
46. Делегаты.
47. События.
48. Многопоточные приложения

### **Экзамен**

49. Потоки байтов.
50. Асинхронный ввод-вывод.
51. Потоки символов.
52. Двоичные потоки.
53. Ввод-вывод.
54. Работа с каталогами и файлами.
55. Сохранение объектов (сериализация).
56. Собрки.
57. Создание библиотеки.
58. Использование библиотеки.

59. Рефлексия.
60. Атрибуты.
61. Пространства имен.
62. Директивы препроцессора.
63. Абстрактные структуры данных.
64. Пространство имен System.Collections.
65. Классы-прототипы.
66. Чистичные типы.
67. Обнуляемые типы.
68. Событийно-управляемое программирование.
69. Шаблон Windows-приложения.
70. Класс Control.
71. Элементы управления.
72. Замечания о формах.
73. Класс Form.
74. Диалоговые окна.
75. Класс Application.
76. Введение в графику.
77. Небезопасный код.
78. Регулярные выражения.
79. Документирование в формате XML.

### 7.3.3. Паспорт фонда оценочных средств

| № п/п | Контролируемые разделы (темы) дисциплины             | Код контролируемой компетенции или ее части | Наименование оценочного средства                                                                                                                                                    |
|-------|------------------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.    | Введение в объектно-ориентированное программирование | ОПК-1, ОПК-6, ПК-3, ПК-24                   | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 2.    | Основные понятия языка C#                            | ОПК-1, ОПК-6, ПК-3, ПК-24                   | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 3.    | Переменные, операции и выражения                     | ОПК-1, ОПК-6, ПК-3, ПК-24                   | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 4.    | Операторы                                            | ОПК-1, ОПК-6, ПК-3, ПК-24                   | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 5.    | Классы: основные понятия                             | ОПК-1, ОПК-6, ПК-3, ПК-24                   | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка                                                                                                                |

| №<br>п/п | Контролируемые<br>разделы (темы)<br>дисциплины    | Код контролируемой<br>компетенции или ее<br>части | Наименование оценочного средства                                                                                                                                                    |
|----------|---------------------------------------------------|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|          |                                                   |                                                   | выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз)                                                                      |
| 6.       | Массивы и строки                                  | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 7.       | Классы: подробности                               | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 8.       | Иерархии классов                                  | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 9.       | Интерфейсы и<br>структурные типы                  | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 10.      | Делегаты, события<br>и потоки выполнения          | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 11.      | Работа с файлами                                  | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 12.      | Сборки, библиотеки,<br>атрибуты, директивы        | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 13.      | Структуры данных,<br>коллекции и классы-прототипы | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 14.      | Введение в программирование под<br>Windows        | ОПК-1, ОПК-6, ПК-3,<br>ПК-24                      | Индивидуальный опрос (ИО), защита лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |
| 15.      | Дополнительные                                    | ОПК-1, ОПК-6, ПК-3,                               | Индивидуальный опрос (ИО), защита                                                                                                                                                   |

| № п/п | Контролируемые разделы (темы) дисциплины | Код контролируемой компетенции или ее части | Наименование оценочного средства                                                                                                                  |
|-------|------------------------------------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
|       | средства С#                              | ПК-24                                       | лабораторных работ (ЗЛР), проверка выполнения СР по дисциплине, контрольная работа (КР), тестирование (Т), курсовой проект (КурсП), экзамен (Экз) |

## 8. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ(МОДУЛЮ)

| № п/п | Наименование издания | Вид издания (учебник, учебное пособие, методические указания, компьютерная программа) | Автор (авторы) | Год издания | Место хранения и количество |
|-------|----------------------|---------------------------------------------------------------------------------------|----------------|-------------|-----------------------------|
|       |                      |                                                                                       |                |             |                             |

## 9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

| Вид учебных занятий         | Деятельность студента                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Лекция                      | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, лабораторном или практическом занятии                                                                                                                                                                                                                                                                                                                                     |
| Лабораторные занятия        | Подготовка к лабораторным работам, оформление отчетов и защита лабораторных работ включает: проработку и анализ теоретического материала, выполнение заданий, решение задач, описание проделанной экспериментальной работы, а также контроль знаний по теме лабораторной работы с помощью приведенных контрольных вопросов и заданий.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Контрольная/курсовая работа | Знакомство с основной и дополнительной литературой, включая справочные издания, зарубежные источники, конспект основных положений, терминов, сведений, требующих для запоминания и являющихся основополагающими в этой теме. Составление аннотаций к прочитанным литературным источникам. Предусматривает закрепление навыков программирования на языке С# в инструментальной среде разработки MICROSOFT VISUAL STUDIO в виде самостоятельного написания программного приложения по заданной теме. В процессе выполнения курсового проекта необходимо: составить развернутое описание поставленной задачи; выполнить анализ задания, обосновать структуру входных и выходных данных; разработать программу, отладить ее и подготовить документацию к ней. По результатам выполнения работы оформляется пояснительная записка, включающая введение; разделы основной части; заключение, список использованных источников, приложения. |
| Подготовка к заче-          | При подготовке к зачету/экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и решение задач на лабораторных заняти-                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| Вид учебных занятий | Деятельность студента |
|---------------------|-----------------------|
| ту/экзамену         | ях.                   |

## **10. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **10.1.Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**

#### **10.1.1. Основная литература:**

1. Хорев П.Б.Объектно-ориентированное программирование [Текст]: учеб.пособие: рек. УМО. – 3-е изд., испр. –М.: Академия, 2011 (Тверь: ОАО «Тверской полиграф. Комбинат», 2011). – 446 с.

2. Казанский А.А. Объектно-ориентированное программирование на языке MicrosoftVisual C# в среде разработки MicrosoftVisualStudio 2008 и .NET Framework. 4.3: Учебное пособие и практикум / Казанский А. А. – М.: Московский государственный строительный университет, ЭБС АСВ, 2013. – 180 с. – Режим доступа: <http://www.iprbookshop.ru/19258>

#### **10.1.2. Дополнительная литература:**

1. Подбельский В.В. Язык Си#. Базовый курс [Электронный ресурс]: учебное пособие/ Подбельский В.В. – Электрон.текстовые данные. – М.: Финансы и статистика, 2011. – 384 с. – Режим доступа: <http://www.iprbookshop.ru/18866>

2. Приемы объектно-ориентированного проектирования. Паттерны проектирования [Электронный ресурс]/ Гамма Эрих и др. – Электрон.текстовые данные. – М.: ДМК Пресс, 2007. – 368 с. – Режим доступа: <http://www.iprbookshop.ru/6905>.

3. Астахова И.Ф. Объектно-ориентированное проектирование информационных систем [Текст]: учебное пособие для вузов / Воронеж.гос. ун-т. – Воронеж: Издательско-полиграфический центр Воронежского государственного университета, 2013 (Воронеж:ИПЦ ВГУ, 2013). – 51 с.

### **10.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине(модулю), включая перечень программного обеспечения и информационных справочных систем:**

- Персональные компьютеры с ОС Windows 7\*;
- Microsoft Office;
- Microsoft Visual Studio 2010ивыше.

### **10.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины(модуля):**

- [www.citforum.ru](http://www.citforum.ru)
- <http://www.lastmile.su/>

- <http://www.connect.ru>
- [www.ieee.org](http://www.ieee.org)
- <http://www.intuit.ru>

## **11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА:**

Компьютерный класс.

## **12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ (образовательные технологии)**

Традиционная лекция имеет несколько ограниченные возможности формирования в сознании студентов ярких представлений элементов изучаемого материала, несущих смысловую нагрузку. Поэтому компьютерная демонстрация лекционного материала является одним из решений изложенной выше проблемы. Лекция должна побуждать к познанию и творческому поиску, а также служить примером использования современных технологий. При представлении электронных презентаций подача информации преподносится модулями на «зрительном», «графическом» и «звуковом» уровнях, что является важным фактором для улучшения восприятия лекционного материала студентами.

Для сопровождения всего лекционного занятия или отдельной его части: этапа мотивации, изучения нового материала, контроля за усвоением используются слайды, созданные с помощью программы графических презентаций PowerPoint. Состав информационных объектов определяется особенностями конкретной темы и целевым назначением занятия. В качестве демонстрируемых фрагментов могут быть использованы текстовые материалы, статические и динамические изображения, контрольные задания и т. п. Для эффективного предъявления учебного материала применяются мультимедийные средства отображения информации.

На визуализированной лекции удобно осуществлять обратную связь. Для этого можно на завершающем этапе лекции предложить студентам выбрать правильные из имеющихся вариантов ответов на несколько простых вопросов по всему изученному на занятии материалу. Форма контроля определяется уровнем подготовленности студентов, содержанием учебного материала.

Таким образом, используя современные программно-технические средства, преподаватель имеет возможность проводить более наглядные и информационно насыщенные занятия, иллюстрировать каждое новое понятие и его связи с соответствующими задачами практики; и тем самым улучшить процесс восприятия и усвоения материала. Система контрольных мероприятий должна обеспечивать объективную оценку знаний и навыков студентов, способствовать повышению эффективности всех видов учебных занятий. Для освоения всех разделов дисциплины эффективно использование обучающих и контролирую-

щих компьютерных программ. При освоении всех разделов дисциплины необходимо сочетание различных форм учебной деятельности: изучение лекционного материала, выполнение заданий на практических занятиях, как с использованием компьютера, так и без него, самостоятельная работа с рекомендуемой литературой и использование методических указаний, консультации преподавателей при выполнении дополнительных заданий.

При реализации различных видов учебной работы используются следующие образовательные технологии:

1. Лекционные занятия проводятся с широким использованием активных и интерактивных форм, в том числе мультимедийных технологий.
2. На лабораторных и практических занятиях используются интерактивные формы проведения занятий.
3. Внеаудиторная работа широко использует возможности Интернет и другие информационные источники, с целью самостоятельного формирования и развития профессиональных навыков обучающихся.

По завершении тем, для закрепления материала рекомендуется выдача самостоятельных заданий по изученным темам. Рекомендуется практиковать написание и заслушивание кратких докладов студентов по изучаемым темам. При изучении дисциплины целесообразно использовать материалы интернет-ресурсов образовательной, аналитической направленности.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 «Информационные системы и технологии».

**Руководитель основной образовательной программы**

канд. техн. наук, доцент  
кафедры информационных технологий  
и автоматизированного  
проектирования в  
строительстве

 /О.В. Курипта /

Рабочая программа одобрена учебно-методической комиссией факультета  
«Экономики, менеджмента и информационных технологий»

«07» сентября 2017г., протокол № 3

Председатель доктор техн. наук, профессор  Курочка П.Н.  
учёная степень и звание, подпись инициалы, фамилия

**Эксперт**

ВТУИТ к.ф.м.н. доцент Курочкин и Ю. Курочкова  
(место работы) (занимаемая должность) (подпись) (инициалы, фамилия)

