Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 03.10.2023 12:52:22
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» ____02_____ 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Объектно-ориентированное программирование»

Направление подготовки/специальность
09.03.02 Информационные системы и технологии

Профиль/специализация

Программное обеспечение игровой компьютерной индустрии

Квалификация
Бакалавр

Формы обучения
Очная

Москва, 2023 г.

Разработчик(и):
Доцент кафедры
«Информатика и информационные технологии»  / П. С. Новиков /

Согласовано:
Заведующий кафедрой
«Информатика и информационные технологии»,
к.т.н.  / Е.В. Булатников /

Содержание

1. Цели, задачи и планируемые результаты обучения по дисциплине 4
2. Место дисциплины в структуре образовательной программы 4
3. Структура и содержание дисциплины 4
3.1. Виды учебной работы и трудоемкость (по формам обучения) 5
3.2. Тематический план изучения дисциплины 5
3.3. Содержание дисциплины 8
3.4. Тематика семинарских/практических и лабораторных занятий 12
3.5. Тематика курсовых проектов (курсовых работ) 14
4. Учебно-методическое и информационное обеспечение 14
4.1. Нормативные документы и ГОСТы 14
4.2. Основная литература 14
4.3. Дополнительная литература 15
4.4. Электронные образовательные ресурсы 15
4.5. Лицензионное и свободно распространяемое программное обеспечение 15
4.6. Современные профессиональные базы данных и информационные справочные системы. 15
5. Материально-техническое обеспечение 16
6. Методические рекомендации 16
6.1. Методические рекомендации для преподавателя по организации обучения 16
6.2. Методические указания для обучающихся по освоению дисциплины 16
7. Фонд оценочных средств 17
7.1. Методы контроля и оценивания результатов обучения 17
7.2. Шкала и критерии оценивания результатов обучения 17
7.3. Оценочные средства 17

1. Цели, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины «Объектно-ориентированное программирование» является формирование понимания идеологии и ключевых аспектов объектно-ориентированного программирования (ООП) на языке C#, достаточного для практического использования в процессе дальнейшего обучения и в профессиональной сфере.

К основным **задачам** освоения дисциплины следует отнести:

- изучение языка C# для проектирования объектной структуры программы
- изучение средств языка C# для создания объектной структуры программы
- получение знаний и практических навыков в области проектирования и разработки объектно-ориентированных программ.

Обучение по дисциплине «Объектно-ориентированное программирование» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-2 Способность понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использование их при решении задач профессиональной деятельности	Знать: Принципы построения объектно-ориентированных программ на языке C# Уметь: выполнять практические работы при помощи объектно-ориентированных средств языка C#; Владеть: Навыками применения языка C# при создания объектно-ориентированных программ.
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	Знать: Основные конструкции объектно-ориентированного языка C# и способы их применения при разработке алгоритмов и программ; Уметь: создавать алгоритмы и программы на языке C#; Владеть: навыками составления алгоритмов и программ с использованием объектно-ориентированных конструкций языка C#

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к модулю «Базовые программирование» обязательной части Блока 1. Дисциплины (модули) учебного плана программы бакалавриата.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Backend-разработка;
- Шаблоны проектирования;
- Функциональное программирование;
- Производственная практика (преддипломная);
- Выполнение и защита выпускной квалификационной работы.

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

3.1. Виды учебной работы и трудоемкость (по формам обучения)

3.1.1. Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры
			3
1	**Аудиторные занятия**	**72**	**72**
	В том числе:		
1.1	Лекции	36	36
1.2	Семинарские/практические занятия	36	36
1.3	Лабораторные занятия	36	36
2	**Самостоятельная работа**	**72**	**72**
	В том числе:		
2.1	Подготовка и выполнение лабораторных работ	72	72
3	**Курсовое проектирование**		**КП**
4	**Промежуточная аттестация**		
	Экзамен	экзамен	экзамен
	Итого:	**144**	**144**

3.2. Тематический план изучения дисциплины (по формам обучения)

3.2.1. Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1.1	Тема №1 «Введение в объектно-ориентированное программирование»	2	0,5		1		0,5
2.1	Тема №2 «Классы»	4,5	2		1		1,5
2.1	Лабораторная работа № 1 «Создание программы с использованием классов»	3,5			2		1,5
3.1	Тема №3 «Структуры»	2,5	1				1,5
3.2	Лабораторная работа № 2 «Добавление в программу из п.п. 2.1 структур. Изучение разницы в поведении типов значений и ссылочных типов. Изучение работы сборщика мусора»	3,5			2		1,5
4.1	Тема №4 «Типы значений и ссылочные типы»	2,5	1				1,5
5.1	Тема №5 «Типы значений и ссылочные типы»	2,5	1				1,5
6.1	Тема №6 «Область видимости переменных и констант»	2,5	1				1,5
7.1	Тема №7 «Пространства имен»	2,5	1				1,5
8.1	Тема №8 «Модификаторы доступа»	2,5	1				1,5

8.2	Лабораторная работа № 3 «Работа с пространством имен и модификаторами доступа. Изучение работы программы при разных сочетаниях модификатор»	3,5			2		1,5
9.1	Тема №9 «Свойства»	2,5	1				1,5
9.2	Лабораторная работа № 4 «Создание свойств. Написание свойств разными способами (по умолчания, развернутых, вычисляемых, с короткой записью). Применение к свойствам модификатор доступа. Создание перезагружаемых методов класса»	3,5			2		1,5
10.1	Тема №10 «Перегрузка методов»	2,5	1				1,5
11.1	Тема №11 «Статические члены и модификатор static»	2,5	1				1,5
11.2	Лабораторная работа № 5 «Добавление в программу различных статических членов. Изучение поведение членов с модификатором static»	3,5			2		1,5
12.1	Тема №12 «Константы, поля и структуры для чтения»	2,5	1				1,5
13.1	Тема №13 «Null в ссылочных и значимых типах»	2,5	1				1,5
13.2	Лабораторная работа № 6 «Изучение поведения объектов, допускающих значение null. Изучение взаимодействие между объектами, допускающих значение null»	3,5			2		1,5
14.1	Тема №14 «Псевдонимы типов и статический импорт»	2,5	1				1,5
15.1	Тема №15 «Наследование»	3,5	2				1,5
16.1	Тема №16 «Преобразование типов»	2,5	1				1,5
16.2	Лабораторная работа № 7 «Создание программы с наследованием элементов. Изучение реализации взаимодействия между базовыми элементами и их наследниками. Изучение преобразование типов в наследовании»	3,5			2		1,5
17.1	Тема №17 «Виртуальные методы и свойства, скрытие методов и свойств, Различие переопределения и скрытия методов»	2,5	1				1,5
18.1	Тема №18 «Абстрактные классы и члены классов»	3,5	2				1,5
18.2	Лабораторная работа № 8 «Создание программы с использованием абстрактных классов»	3,5			2		1,5
19.1	Тема №19 «Класс System.Object»	2,5	1				1,5

20.1	Тема №20 «Обобщения, ограничение обобщений, наследование обобщенных типов»	2,5	1				1,5
20.2	Лабораторная работа № 9 «Создание программы с классами предусматривающие обобщения. Установление ограничений на обобщения. Изучение наследования таких классов, ковариантности и котравариантности»	3,5			2		1,5
21.1	Тема №21 «Обработка исключений»	2,5	1				1,5
21.2	Лабораторная работа № 10 «Обработка исключений в программе. Изучение использования директив throw. Написание собственных классов исключений»	3,5			2		1,5
22.1	Тема №22 «Делегаты»	3,5	2				1,5
22.2	Лабораторная работа № 11 «Использование делегатов. Написание программы с разными параметрами и способами применения делегатов»	3,5			2		1,5
23.1	Тема №23 «Лямбды»	2,5	1				1,5
23.2	Лабораторная работа № 12 «Изучение применения лямбда-выражений, с разными параметрами, возвращаемыми значениями и применениями. Изучение замыканий»	3,5			2		1,5
24.1	Тема №24 «События»	2,5	1				1,5
24.2	Лабораторная работа № 13 «Применение событий. Изучение разных способов использования событий»	3,5			2		1,5
25.1	Тема №25 «Ковариантность и контравариантность»	2,5	1				1,5
26.1	Тема №26 «Замыкания»	2,5	1				1,5
27.1	Тема №27 «Интерфейсы»	8	2,5		4		1,5
27.2	Лабораторная работа № 14 «Применение интерфейсов. Написание программы с использованием интерфейсов. Изучение поведения членов, реализованных от интерфейсов»	2,5			1		1,5
28.1	Тема №28 «Определение операторов, перезагрузка операций и преобразования типов»	2,5	1				1,5
28.2	Лабораторная работа № 15 «Изучение работы программ с использованием перезагрузки операторов»	2,5			1		1,5
29.1	Тема №29 «Индексаторы»	2,5	1				1,5
30.1	Тема №30 «Частные классы и методы, анонимные типы»	2,5	1				1,5
30.2	Лабораторная работа № 16 «Написание программы с применением анонимных классов и типов»	2,5			1		1,5

31.1	Тема №31 «Кортежи»	2	0,5				1,5
32.1	Тема №32 «Records»	1,5	0,5				1
32.2	Лабораторная работа № 17 «Применение индексаторов, кортежей и не изменяемых типов»	2,5			1		1,5
Итого		**144**	**36**		**36**		**72**

3.3. Содержание дисциплины

Тема 1. ООП

- Введение
- Методология разработки объектно-ориентированного программного обеспечения
- Основные понятия и терминология объектно-ориентированного анализа и проектирования
- Инкапсуляция
- Наследование
- Полиморфизм

Тема 2. Классы

- Классы
- Объекты
- Поля и методы класса
- Создание объекта класса
- Конструктор по умолчанию
- Обращения к функциональности класса

Тема 3. Конструкторы, инициализаторы и деконструкторы

- Создание конструкторов
- Ключевое слово this
- Цепочка вызовов конструкторов
- Инициализаторы объектов
- Диконструкторы

Тема 4. Структуры

- Определение структуры
- Создание объекта структуры
- Инициация полей по умолчанию
- Конструкторы структуры
- Копирование структуры

Тема 5. Типы значений и ссылочные типы

- Типы значений
- Ссылочные типы
- Хранение типов в памяти, стек и куча, общие сведения о очистке мусора
- Составные типы
- Копирование значений
- Ссылочные типы внутри типов значений
- Объекты как параметры методов

Тема 6. Область видимости переменных и констант

Тема 7. Пространства имен

- Класс Program
- Метод Main
- Программы верхнего уровня
- Пространства имен, подключение пространства имен

- Вложенные пространства имен
- Пространства имен уровня файла
- Глобальное пространство имен
- Подключение пространства имен по умолчанию

Тема 8. Модификаторы доступа

- Виды модификаторов доступы
- Модификаторы доступа в рамка проекта
- Модификаторы доступа в рамках сборки
- Файл как область видимости

Тема 9. Свойства

- Определение свойств
- Свойства для чтения и записи
- Вычисляемые свойства
- Модификаторы доступа в свойствах
- Автоматические свойства
- Блок init
- Сокращенная запись свойств
- Модификатор required

Тема 10. Перегрузка методов

Тема 11. Статические члены и модификатор static

- Модификатор static
- Статические поля
- Статический свойства
- Статические методы
- Статические конструкторы
- Статические классы

Тема 12. Константы, поля и структуры для чтения

- Константы класса
- Поля для чтения и модификатор readonly
- Сравнение констант
- Структуры для чтения

Тема 13. Null в ссылочных и значимых типах

- Null и ссылочные типы
- Оператор ! (null-forgiving operator)
- Исключение кода из nullable-контекста
- Null и значимые типы
- Преобразование значимых nullable-типов
- Операции с nullable-типами
- Проверка на null
- Null guard
- Оператор ??
- Оператор условного null

Тема 14. Псевдонимы типов и статический импорт

- Псевдонимы
- Статический импорт

Тема 15. Наследование

- Наследование
- Доступ к членам базового класса из класса-наследника
- Ключевое слово base
- Конструкторы в производных классах

- Порядок вызова конструкторов

Тема 16. Преобразование типов

Общие сведения о преобразовании типов

- Восходящие преобразования. Upcasting
- Нисходящие преобразования. Downcasting
- Способы преобразований

Тема 17. Виртуальные методы и свойства, скрытие методов и свойств, Различие переопределения и скрытия методов

- Общее сведения о виртуализации
- Ключевое слово base
- Преобразование свойств
- Запрет переопределения методов
- Скрытие свойств
- Скрытие методов
- Скрытие переменных и констант
- Переопределение
- Различие скрытия и переопределения

Тема 18. Абстрактные классы и члены классов

- Общие сведения об абстрактных классах
- Абстрактные члены класса
- Абстрактные методы
- Абстрактные свойства
- Отказ от реализации абстрактных членов

Тема 19. Класс System.Object

- Метод ToString
- Метод GetHashCode
- Получение типа объекта и метод GetType
- Метод Equals
- Различия в сравнения и методики сравнения
- Переопределение стандартных методов сравнения

Тема 20. Обобщения, ограничение обобщений, наследование обобщенных типов

- Обобщения
- Статические поля обобщенных классов
- Использование универсальных параметров
- Обобщение методов
- Ограничение обобщений
- Ограничение обобщенных методов
- Ограничение обобщений в типах
- Типы ограничений
- Стандартные ограничения
- Наследование обобщенных типов

Тема 21. Обработка исключений

- Обработка исключений
- Конструкция try..catch..finally
- Блок catch и фильтры исключений
- Типы исключений. Класс Exception
- Генерация исключения и оператор throw
- Создание классов исключений

- Поиск блока catch при обработке исключений

Тема 22. Делегаты

- Место определения делегата
- Параметры и результат делегата
- Присвоение ссылки на метод
- Соответствие методов делегату
- Добавление методов в делегат
- Объединение делегатов
- Вызов делегата
- Обобщенные делегаты
- Делегаты как параметры методов
- Возвращение делегатов из метода
- Применение делегатов
- Добавление и удаление методов в делегате
- Анонимные методы
- Делегаты Action, Predicate и Func

Тема 22. Лямбды

- Общие сведения о лямбда-выражениях
- Параметры лямбды
- Возвращение результата
- Добавление и удаление действий в лямбда-выражении
- Лямбда-выражение как результат метода

Тема 24. События

- Общие сведения о событиях
- Определение и вызов события
- Добавления обработчика события
- Удаление обработчика события
- Управление обработчиками событий
- Передача данных событию

Тема 25. Ковариантность и контравариантность

- Ковариантность
- Контравариантность
- Ковариантность и контравариантность в обобщенных делегатах
- Совмещение ковариантности и контравариантности

Темя 26. Замыкания

- Общие сведения о замыканиях
- Реализация с помощью лямбда-выражений
- Применение параметров

Темя 27. Интерфейсы

- Определение интерфейсов
- Применение интерфейсов
- Явная реализация интерфейсов
- Реализация интерфейсов в базовых и производных классах
- Наследование интерфейсов
- Интерфейсы в обобщениях
- Ковариантность и контравариантность обобщенных интерфейсов

Тема 28. Определение операторов, перезагрузка операций и преобразования типов

- Определение операторов

- Определение инкремента и декремента
- Определение операций true и false
- Перегрузка операций преобразования типов

Тема 29. Индексаторы

- Общие сведения
- Индексы
- Применение нескольких параметров
- Блоки get и set
- Перегрузка индексаторов
- Переменная-ссылка
- Ссылка как результат функции

Тема 30. Частные классы и методы, анонимные типы

- Частичные методы
- Частные классы
- Анонимные типы

Тема 31. Кортежи

- Общие сведения о кортежах
- Кортеж как результат метода
- Кортеж как параметр метода

Тема 32. Records

- Неизменяемый тип и ключевое слово records
- Сравнение на равенство
- Оператор with
- Позиционные records
- Позиционные структуры для чтения
- Наследование

3.4. Тематика семинарских/практических и лабораторных занятий

3.4.1. Семинарские/практические занятия

Семинарские и практические занятия не предусмотрены.

3.4.2. Лабораторные занятия

Лабораторная работа № 1 «Создание программы с использованием классов»

В данной работе рассмотрено, как создать простейшую объектно-ориентированную программу с использованием простых классов, методы определения в классах полей и методов, способ создания объектов на основе определенных классов, взаимодействия объектов в программе.

Лабораторная работа № 2 «Добавление в программу из п.п. 2.1 структур. Изучение разницы в поведении типов значений и ссылочных типов. Изучение работы сборщика мусора»

В данной работе рассмотрено, как добавлять в объектно-ориентированную программу структур, изучается чем структуры отличаются от классов, изучается на примерах различие в поведениях этих двух типов объектов.

Лабораторная работа № 3 «Работа с пространством имен и модификаторами доступа. Изучение работы программы при разных сочетаниях модификатор»

В данной работе изучается работа с пространством имен разных типов, влияние модификаторов доступа на сущности, проводится анализ целостности данных при разных условиях описания членов программы. Вырабатываются навыки правильного написания приложения с применением принципов SOLID.

Лабораторная работа № 4 «Создание свойств. Написание свойств разными способами (по умолчания, развернутых, вычисляемых, с короткой записью). Применение к свойствам модификатор доступа. Создание перезагружаемых методов класса»

В данной работе изучается создание свойств структур и классов, вырабатываются навыки их определения, изучаются примеры возможного их применения и использования.

Лабораторная работа № 5 «Добавление в программу различных статических членов. Изучение поведение членов с модификатором static»

В данной работе изучается определение статических методов, их правильное применение, проверяется на практике различие в поведение статических и не статических сущностей программы.

Лабораторная работа № 6 «Изучение поведения объектов, допускающих значение null. Изучение взаимодействие между объектами, допускающих значение null»

В данной работе изучается использование объектов имеющих возможность иметь тип null, способы добавления возможность добавления возможности иметь тип null к «no null» типам и особенности поведения объектов, имеющие возможность применять тип null. Нарабатываются навыки работы с такими объектами.

Лабораторная работа № 7 «Создание программы с наследованием элементов. Изучение реализации взаимодействия между базовыми элементами и их наследниками. Изучение преобразование типов в наследовании»

В данной работе изучается механизм наследования объектно-ориентированных программ, способы его применения и особенности использования.

Лабораторная работа № 8 «Создание программы с использованием абстрактных классов»

В данной работе изучается механизм применения абстрактных классов, вырабатываются навыки использования таких конструкций. Нарабатывается способы написания программ с использованием методики SOLID.

Лабораторная работа № 9 «Создание программы с классами предусматривающие обобщения. Установление ограничений на обобщения. Изучение наследования таких классов, ковариантности и котравариантности»

В данной работе изучается работа с дженериками, способы их применения, возможности ограничивать их значения. Нарабатываются навыки применения обобщений.

Лабораторная работа № 10 «Обработка исключений в программе. Изучение использования директив throw. Написание собственных классов исключений»

В данной работе рассматривается правильная работа с исключениями, способы и возможности обработки ошибок выполнения, устранения их последствий.

Лабораторная работа № 11 «Использование делегатов. Написание программы с разными параметрами и способами применения делегатов»

В данной работе рассматривается применение делегатов в объектно-ориентированном программировании, способы их использования, области возможного применения.

Лабораторная работа № 12 «Изучение применения лямбда-выражений, с разными параметрами, возвращаемыми значениями и применениями. Изучение замыканий»

В данной работе изучается способы определения и область применения лямбда-выражений, особенности использования, возможный спектр применения.

Лабораторная работа № 13 «Применение событий. Изучение разных способов использования событий»

В данной работе изучается способы обработки событий и круг их использования в объектно-ориентированном программировании.

Лабораторная работа № 14 «Применение интерфейсов. Написание программы с использованием интерфейсов. Изучение поведения членов, реализованных от интерфейсов»

В данной работе рассматривается применение интерфейсов, реализация сущностей на основе интерфейсов, способы наследования самих интерфейсов и объектов, созданных с их реализацией.

Лабораторная работа № 15 «Изучение работы программ с использованием перезагрузки операторов»

В данной работе рассматриваются способы перезагрузке операторов.

Лабораторная работа № 16 «Написание программы с применением анонимных классов и типов»

В данной работе изучаются создание, применение и использования анонимных классов и типов, круг их применения.

Лабораторная работа № 17 «Применение индексаторов, кортежей и не изменяемых типов»

В данной работе рассматриваются индексаторы, кортежи и не изменяемые типы. Способы их определения, поведенческие особенности, способы использования в объектно-ориентированном программировании.

3.5. Тематика курсовых проектов (курсовых работ)

Курсовой проект предусматривает написание объектно-ориентированной программы, по выбранной тематике, с использованием методик изучаемых в курсе, а также создание пояснительной записки, с объяснением выбора технологий и обоснования способов реализации.

4. Учебно-методическое и информационное обеспечение

4.1. Нормативные документы и ГОСТы

Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);

Федеральный государственный образовательный стандарт высшего образования (уровень магистратуры) по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденный приказом Министерства науки и высшего образования Российской Федерации от 19 сентября 2017 г. N 926 (в редакции приказа от 26 ноября 2020 г. №1456);

Приказ Министерства образования и науки РФ от 05 апреля 2017 г. № 301 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры;

4.2. Основная литература

1. Документация по C#
2. C# Sharp 10 и .NET 6. Современная кросс-платформенная разработка : 16+ / Марк Прайс ; [перевел с английского С. Черников]. - 6-е изд. - Санкт-Петербург [и др.] : Питер, 2023. - 844, [1] с. : ил., портр., табл.; 23 см. - (Серия "Для профессионалов").; ISBN 978-5-4461-2249-3 : 500 экз.

3. Язык программирования C# 9 и платформа .NET 5: основные принципы и практики программирования / Эндрю Троелсен, Филипп Джепикс; перевод с английского и редакция Ю. Н. Артеменко. - 10-е изд. - Москва : Диалектика ; Санкт-Петербург : Диалектика, 2022. - 1391 с. : ил., табл.; 24 см. - (Профессионалам от профессионалов).; ISBN 978-5-907458-67-3 : 500 экз.

4. C# 9.0. Справочник : полное описание языка / Албахари Джозеф; перевод с английского и редакция Ю. Н. Артеменко. - Москва : Диалектика ; Санкт-Петербург : Диалектика, 2021. - 1056 с. : ил., табл.; 25 см.; ISBN 978-5-907365-81-0 : 500 экз.

5. Лебедева, Т. Н. Теория и практика объектно-ориентированного программирования : учебное пособие / Т. Н. Лебедева. — 2-е изд. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 221 c. — ISBN 978-5-4486-0663-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/81498.html (дата обращения: 01.10.2023)

4.3. Дополнительная литература

1. Программирование на C#. Основные сведения [Текст] : [12+] / Васильев А. Н. - Москва : Эксмо, 2018. - 582, [1] с. : табл.; 24 см. - (Российский компьютерный бестселлер).; ISBN 978-5-04-092519-3 : 2000 экз.

2. C#. Программирование [Текст] : учебное пособие : [в 3 ч.] / Н. А. Тюкачев, В. Г. Хлебостроев ; М-во образования и науки Российской Федерации, Федеральное гос. бюджетное образовательное учреждение высш. проф. образования "Воронежский гос. ун-т". - Воронеж : Изд.-полиграфический центр Воронежского гос. ун-та, 2013. - 21 см. - (Учебник Воронежского государственного университета).

3. Бабушкина, И. А. Практикум по объектно-ориентированному программированию / И. А. Бабушкина, С. М. Окулов. — 5-е изд. — Москва : Лаборатория знаний, 2020. — 367 c. — ISBN 978-5-00101-780-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/12254.html (дата обращения: 01.10.2023)

4.4. Электронные образовательные ресурсы

Шурыгин В.Н. ЭОР «Объектно-ориентированное программирование» [Электронный ресурс] Режим доступа - https://online.mospolytech.ru/course/view.php?id=1165

4.5. Лицензионное и свободно распространяемое программное обеспечение

1. Visual Studio
2. Visual Studio Code
3. Модульная платформа .NET

4.6. Современные профессиональные базы данных и информационные справочные системы.

1. https://urait.ru/
2. https://www.iprbookshop.ru/
3. https://e.lanbook.com/

5. Материально-техническое обеспечение

Компьютерные классы со следующей оснащенностью: столы, стулья, аудиторная доска, использование переносного мультимедийного комплекса (переносной проектор, персональный ноутбук). Персональные компьютеры, мониторы, мышки, клавиатуры. Рабочее место преподавателя: стол, стул.

Программное обеспечение: Microsoft Windows или Linux на основе deb-пакетов (Debian, Ubuntu, Astra и т.д.), сервер с системой контроля версий GIT (GitLab)

6. Методические рекомендации

6.1. Методические рекомендации для преподавателя по организации обучения

Методика преподавания дисциплины «Объектно-ориентированное программирование» предусматривает использование онлайн-курса в системе дистанционного обучения Университета, групповых и индивидуальных консультаций обучающихся, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Лабораторные работы по дисциплине «Объектно-ориентированное программирование» осуществляются в форме самостоятельной проработки теоретического материала обучающимися; выполнения практического задания; защиты преподавателю лабораторной работы (знание теоретического материала и выполнение практического задания по теме лабораторной работы).

6.2. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в соответствии с учебным планом.

На занятиях осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на умение применять полученные знания на практике, в том числе при решении реальных задач, отличающихся от проработанных.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, самостоятельно знакомятся с теоретическим материалом, выполняют лабораторные работы, готовятся к текущему контролю и промежуточной аттестации.

Выполнение и сдача лабораторных работ проводится с применением системы контроля версий GIT.

Текущий контроль осуществляется на аудиторных занятиях в виде защиты лабораторных работ. Критериями оценки результатов являются:

– уровень освоения теоретического материала;
– уровень владения практическими навыками (в виде вопросов по процессу выполнения лабораторных работ);
– умения обучающегося использовать теоретические знания при выполнении практических задач (в виде дополнительных заданий);
– сформированность компетенций;
– оформление материала в соответствии с требованиями.

Промежуточный контроль осуществляется на зачете в форме тестирования в системе дистанционного обучения Университета, включающего вопросы на знание практической части языка C#.

7. Фонд оценочных средств

7.1. Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций: **лабораторные работы**, **экзамен**.

7.2. Шкала и критерии оценивания результатов обучения

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Объектно-ориентированное программирование».

7.2.1. Критерии оценки ответа на зачёте

(формирование компетенций — ОПК-2, ОПК-4)

«Зачтено»:

Выполнены все виды учебной работы, предусмотренные учебным планом. Обучающийся демонстрирует прочные теоретические знания, практические навыки, владеет терминами, делает аргументированные выводы и обобщения, приводит примеры, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации, которые обучающийся может исправить при коррекции преподавателем.

«Не зачтено»:

Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Обучающийся демонстрирует незнание теоретических основ предмета, отсутствие практических навыков, не умеет делать аргументированные выводы и приводить примеры, не владеет терминами, проявляет отсутствие логичности и последовательности изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на дополнительные вопросы, допускает значительные ошибки, испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.2.2. Критерии оценки работы обучающегося на лабораторных занятиях:

(формирование компетенций — ОПК-2, ОПК-4)

«5» (отлично): выполнены все практические задания, предусмотренные лабораторными работами, обучающийся четко и без ошибок ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог выполнить дополнительные задания.

«4» (хорошо): выполнены все практические задания, предусмотренные лабораторными работами, обучающийся с корректирующими замечаниями преподавателя ответил на все контрольные вопросы, проявил творческий подход при выполнении заданий, смог частично выполнить дополнительные задания.

«3» (удовлетворительно): выполнены все практические задания, предусмотренные лабораторными работами, с замечаниями преподавателя; обучающийся ответил на все контрольные вопросы с замечаниями, дополнительные задания выполнены с замечаниями.

«2» (неудовлетворительно): обучающийся не выполнил или выполнил неправильно практические задания, предусмотренные лабораторными работами, обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы, дополнительные задания выполнены неверно или не выполнены.

7.3. Оценочные средства

7.3.1. Текущий контроль

Текущий контроль осуществляется на аудиторных занятиях в виде защиты лабораторных работ. Лабораторная работа – средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде демонстрации полученных навыков при решении поставленных практических задач.

Примеры вопросов к защите лабораторных работ (оцениваемые компетенции — ОПК-2, ОПК-4).

Лабораторная работа № 1 «Создайте классы и объявите их. Запустите программу и выведите что-нибудь из класса в консоль.»
Вопросы к защите лабораторной работы:

1. Как определить классы в языке программирования? На каких основных элементах класс состоит?
2. Что такое методы класса? Как они объявляются и используются?
3. Как можно создать объекты классов? Какие операторы или конструкции обычно используются для этого?
4. Что такое консольный вывод? Каким образом можно вывести что-нибудь в консоль в выбранном языке программирования?
5. Как связываются объекты с их методами? Как происходит вызов метода для конкретного объекта?
6. Каким образом можно запустить программу и увидеть вывод в консоль? Какие инструменты или команды нужно использовать?

Лабораторная работа № 1 «Создание программы с использованием классов»
Примеры вопросов:
Дан следующий класс:

```
class Person
{
   public string name = "Ben";
   public int age = 18;
   public string email = "ben@gmail.com";

   public Person(string name)
   {
      this.name = name;
   }
   public Person(string name, int age) : this(name)
   {
      this.age = age;
   }
   public Person(string name, int age, string email) : this("Bob", age)
   {
      this.email = email;
   }
}
```

Какое значения будут иметь поля name, age и email после выполнения следующего кода и почему? В каком порядке будут вызываться конструкторы класса Person?

Лабораторная работа № 2 «Добавление в программу из п.п. 2.1 структур. Изучение разницы в поведении типов значений и ссылочных типов. Изучение работы сборщика мусора»
Примеры вопросов:

Почему не компилируется следующая программа:

```
struct Person
{
   public string name;
```

```
}
class Program
{
   static void Main(string[] args)
   {
      Person person;
      Console.WriteLine(person.name);
      person.name = "Bob";
   }
}
```

Лабораторная работа № 3 «Работа с пространством имен и модификаторами доступа. Изучение работы программы при разных сочетаниях модификатор»

Примеры вопросов:

Что выведет на консоль следующая программа и почему?

```
class Person
{
   int age = 26;
   string name = "Tom";

   public Person(int age, string name)
   {
      this.age = age;
      this.name = name;
   }
}
class Program
{
   static void Main(string[] args)
   {
      Person person = new Person(19, "Bob");
      Console.WriteLine(person.name);

      Console.ReadKey();
   }
}
```

Лабораторная работа № 4 «Создание свойств. Написание свойств разными способами (по умолчания, развернутых, вычисляемых, с короткой записью). Применение к свойствам модификатор доступа. Создание перезагружаемых методов класса»

Примеры вопросов:

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```
class Person
{
   internal string Name { internal get; set; } = "Bob";
}
class Program
{
   static void Main(string[] args)
   {
      Person tom = new Person { Name = "Tom" };
      Console.WriteLine(tom.Name);
```

```
            Console.ReadKey();
        }
    }
```

Лабораторная работа № 5 «Добавление в программу различных статических членов. Изучение поведение членов с модификатором static»

Примеры вопросов:

Дана следующая программа:

```
class Person
{
    public static int retirementAge = 60;
    int _age;
    static Person()
    {
        Console.WriteLine($"Начальный пенсионный возраст: {retirementAge}");
    }
    public Person(int age)
    {
        _age = age;
    }
    public void Display()
    {
        if (_age >= retirementAge) Console.WriteLine("Вы уже на пенсии");
        else Console.WriteLine($"До пенсии осталось {retirementAge - _age} лет");
    }
}
class Program
{
    static void Main(string[] args)
    {
        Person tom;
        tom = new Person(34);
        Person.retirementAge = 65;
        tom.Display();

        Console.ReadKey();
    }
}
```

При выполнении какой строки кода в методе Main будет вызван конструктор класса Person?

Лабораторная работа № 6 «Изучение поведения объектов, допускающих значение null. Изучение взаимодействие между объектами, допускающих значение null»

Примеры вопросов:

Может ли переменная типа int принимать значение null? Если нет, то как изменить ее поведение?

Лабораторная работа № 7 «Создание программы с наследованием элементов. Изучение реализации взаимодействия между базовыми элементами и их наследниками. Изучение преобразование типов в наследовании»

Примеры вопросов:

Что выведет на консоль следующая программа и почему?

```
class Auto  // легковой автомобиль
{
   public int Seats { get; set; } // количество сидений
   public Auto(int seats)
   {
      Seats = seats;
   }
}
class Truck : Auto     // грузовой автомобиль
{
   public decimal Capacity { get; set; } // грузоподъемность
   public Truck(int seats, decimal capacity)
   {
      Seats = seats;
      Capacity = capacity;
   }
}
class Program
{
   static void Main(string[] args)
   {
      Truck truck = new Truck(2, 1.1m);
      Console.WriteLine($"Грузовик с грузоподъемностью {truck.Capacity} тонн");
      Console.ReadKey();
   }
}
```

Лабораторная работа № 8 «Создание программы с использованием абстрактных классов»

Примеры вопросов:

Есть ли в следующем коде ошибка? Если есть, то какая?

```
abstract class Base
{
   public abstract void Display();
}
abstract class Derived : Base
{
   public override void Display()
   {
      Console.WriteLine("This is Derived");
   }
}
```

Лабораторная работа № 9 «Создание программы с классами предусматривающие обобщения. Установление ограничений на обобщения. Изучение наследования таких классов, ковариантности и котравариантности»

Примеры вопросов:

Что такое упаковка (boxing) и распаковка (unboxing)?

Лабораторная работа № 10 «Обработка исключений в программе. Изучение использования директив throw. Написание собственных классов исключений»

Примеры вопросов:

Возможно применение конструкции try{}finally{}?

Лабораторная работа № 11 «Использование делегатов. Написание программы с разными параметрами и способами применения делегатов»

Примеры вопросов:

Что будет выведено на консоль в результате выполнения следующей программы:

```
class Program
{
   delegate void Message();

   static void Main(string[] args)
   {
      Message mes = new Message(Hello);
      mes += Hi;
      mes -= Hello;
      mes -= Hi;
      mes();
      Console.Read();
   }
   private static void Hello() { Console.WriteLine("Hello"); }
   private static void Hi() { Console.WriteLine("Hi"); }
}
```

Лабораторная работа № 12 «Изучение применения лямбда-выражений, с разными параметрами, возвращаемыми значениями и применениями. Изучение замыканий»

Лабораторная работа № 13 «Применение событий. Изучение разных способов использования событий»

Примеры вопросов:

Какой будет консольный вывод при выполнении следующей программы:

```
class Program
{
   delegate void Message();

   static void Main(string[] args)
   {
      Message mes1 = Hello;
      mes1 += HowAreYou;
      mes1 += Hello;
      mes1 += Hello;
      mes1 -= Hello;
      mes1();

      Console.Read();
   }
   private static void Hello() { Console.WriteLine("Hello"); }
   private static void HowAreYou() { Console.WriteLine("How are you?"); }
}
```

Лабораторная работа № 14 «Применение интерфейсов. Написание программы с использованием интерфейсов. Изучение поведения членов, реализованных от интерфейсов»

Примеры вопросов:

Класс Tester реализует интерфейсы IFoo и IBar:

```
interface IFoo
{
   void Execute();
```

```
}
interface IBar
{
   void Execute();
}
class Tester : IFoo, IBar
{
   public void Execute()
   {
      Console.WriteLine("Tester Executes");
   }
}
```

Метод Execute() какого именно интерфейса реализует класс Tester?

Лабораторная работа № 15 «Изучение работы программ с использованием перезагрузки операторов»

Примеры вопросов:

Почему не компилируется следующая программа?

```
class Counter
{
   public int Number { get; set; }

   public static int operator + (int val, Counter counter)
   {
      return counter.Number + val;
   }
}
class Program
{
   static void Main(string[] args)
   {
      Counter counter = new Counter { Number = 45 };
      int x = counter + 6;
      Console.WriteLine(x);  // 51

      Console.ReadKey();
   }
}
```

Лабораторная работа № 16 «Написание программы с применением анонимных классов и типов»

Примеры вопросов:

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```
class Program
{
   static void Main(string[] args)
   {
      var user = new { Name = "Tom", Age = 34 };
      user.Name = "Bob";
      Console.WriteLine(user.Name);
```

```
            Console.Read();
        }
    }
```

Лабораторная работа № 17 «Применение индексаторов, кортежей и не изменяемых типов»

Примеры вопросов:

```
Что выведет на экран программа:
Person person = new Person("Tom", 33);

(_, int age) = person;

Console.WriteLine(age);
```

7.3.2. Промежуточная аттестация

Промежуточная аттестация обучающихся в форме зачёта осуществляется по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине, при этом учитываются результаты текущего контроля успеваемости в течение семестра. Зачёт проводится в форме выполнения практического задания и проведение устного опроса включающей вопросы на знание объектно-ориентированной части языка C#. По итогам промежуточной аттестации по дисциплине выставляется оценка «зачтено» или «не зачтено».

Примеры заданий:

1. В курсе «Введение в программирование» вы создавали структуру employee. Теперь создайте класс employee. Класс должен включать поле типа int для хранения номера сотрудника и поле типа float для хранения величины его оклада. Методы класса должны позволять пользователю вводить и отображать данные класса. Напишите функцию main(), которая запросит пользователя ввести данные для трех сотрудников и выведет полученную информацию на экран.

2. Создайте класс date. Его данные должны размещаться в трех полях типа int: month, day и year. Метод класса getdate() должен принимать значение для объекта в формате 12/31/02, а метод showdate() - выводить данные на экран.

3. Расширьте содержание класса employee из ранее выполненной задачи, включив в него класс date и перечисление etype (см. задачу 2 из ЛР 3 дисциплины «Введение в программирование»). Объект класса date будет использоваться для хранения даты приема сотрудника на работу. Перечисление будет использовано для хранения статуса сотрудника: лаборант, секретарь, менеджер и т. д. Последние два поля данных должны быть закрытыми в определении класса employee, как и номер и оклад сотрудника. Вам будет необходимо разработать методы getemploy() и putemploy(), предназначенные соответственно для ввода и отображения информации о сотруднике. Возможно, при создании методов вам понадобится ветвление switch для работы с перечисляемым типом etype. Напишите функцию main(), которая попросит пользователя ввести данные о трех сотрудниках, а затем выведет эти данные на экран.

4. В морской навигации координаты точки измеряются в градусах и минутах широты и долготы. Например, координаты бухты Панити на о. Таити равны 149 градусов 34.8 минут восточной долготы и 17 градусов 31.5 минут южной широты. Это записывается как 149°34.8' W, 17°31.5' S. Один градус равен 60 минутам (устаревшая система также де-

лила одну минуту на 60 секунд, но сейчас минуту делят на обычные десятичные доли). Долгота измеряется величиной от 0 до 180 градусов восточнее или западнее Гринвича. Широта принимает значения от 0 до 90 градусов севернее или южнее экватора.

5. Создайте класс angle, включающий следующие три поля: типа int для числа градусов, типа float для числа минут и типа char для указания направления (N, S, E или W). Объект этого класса может содержать значение как широты, так и долготы. Создайте метод, позволяющий ввести координату точки, направление, в котором она измеряется, и метод, выводящий на экран значение этой координаты, например 179°59.9' E. Кроме того, напишите конструктор, принимающий три аргумента. Напишите функцию mainQ, которая сначала создает переменную с помощью трехаргументного конструктора и выводит ее значение на экран, а затем циклически запрашивает пользователя ввести значение координаты и отображает введенное значение на экране. Для вывода символа градусов (°) можно воспользоваться символьной константой '\xF8'.

6. Создайте класс, одно из полей которого хранит «порядковый номер» объекта, то есть для первого созданного объекта значение этого поля равно 1, для второго созданного объекта значение равно 2 и т. д.

7. Для того чтобы создать такое поле, вам необходимо иметь еще одно поле, в которое будет записываться количество созданных объектов класса (это означает, что последнее поле должно относиться не к отдельным объектам класса, а ко всему классу в целом. Вспомните, какое ключевое слово необходимо при описании такого поля.). Каждый раз при создании нового объекта конструктор может получить значение этого поля и в соответствии с ним назначить объекту индивидуальный порядковый номер.

8. В класс следует включить метод, который будет выводить на экран порядковый номер объекта. Создайте функцию main(), в которой будут созданы три объекта, и каждый объект выведет на экран свой порядковый номер, например: Мой порядковый номер: 2 и т. п.

9. На основе структуры fraction (см. задачу 3 из ЛР 3 дисциплины «Введение в программирование») создайте класс fraction. Данные класса должны быть представлены двумя полями: числителем и знаменателем. Методы класса должны получать от пользователя значения числителя и знаменателя дроби в форме 3/5 и выводить значение дроби в этом же формате. Кроме того, должен быть разработан метод, складывающий значения двух дробей. Напишите функцию main(), которая циклически запрашивает у пользователя ввод пары дробей, затем складывает их и выводит результат на экран. После каждой такой операции программа должна спрашивать пользователя, следует ли продолжать цикл.