

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Набережночелнинский институт (филиал)
Экономическое отделение



Утверждаю

Заместитель директора
по образовательной деятельности
НЧИ КФУ Н.Д.Ахметов



« _____ » _____ 20__ г.

подписано электронно-цифровой подписью

Программа дисциплины

Объектно-ориентированное программирование

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: Прикладная информатика в экономике

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
 - 6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения
 - 6.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания
 - 6.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы
- 6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Лысанов Д.М. (Кафедра бизнес-информатики и математических методов в экономике, Экономическое отделение), DMLysanov@kpfu.ru

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-2	Способен разрабатывать и адаптировать прикладное программное обеспечение

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

ПК-2: знать способы и методы разработки и адаптации прикладного программного обеспечения.

Должен уметь:

ПК-2: уметь разрабатывать и адаптировать прикладное программное обеспечение

Должен владеть:

ПК-2: владеть способностью разрабатывать и адаптировать прикладное программное обеспечение

Должен демонстрировать способность и готовность:

ПК-2: демонстрировать способность и готовность разрабатывать и адаптировать прикладное программное обеспечение

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.В.03 Дисциплины (модули)" основной профессиональной образовательной программы 09.03.03 "Прикладная информатика (Прикладная информатика в экономике)" и относится к вариативной части.

Осваивается на 3 курсе в 5, 6 семестрах.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 9 зачетных(ые) единиц(ы) на 324 часа(ов).

Контактная работа - 102 часа(ов), в том числе лекции - 34 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 68 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 150 часа(ов).

Контроль (зачёт / экзамен) - 72 часа(ов).

Форма промежуточного контроля дисциплины: экзамен в 5 семестре; экзамен в 6 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Методы и классы	5	4	0	8	18
2.	Тема 2. Перегрузка операторов	5	4	0	7	18
3.	Тема 3. Индексаторы и свойства	5	4	0	7	18
4.	Тема 4. Наследование	5	3	0	7	18
5.	Тема 5. Интерфейсы, структуры и перечисления	5	3	0	7	18

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
6.	Тема 6. Обработка исключительных ситуаций	6	4	0	7	12
7.	Тема 7. Применение средств ввода-вывода	6	3	0	7	12
8.	Тема 8. Делегаты, события и лямбда-выражения	6	3	0	6	12
9.	Тема 9. Пространства имен, препроцессор и сборки	6	3	0	6	12
10.	Тема 10. Динамическая идентификация типов, рефлексия и атрибуты	6	3	0	6	12
	Итого		34	0	68	150

4.2 Содержание дисциплины (модуля)

Тема 1. Методы и классы

Управление доступом к членам класса, Модификаторы доступа, Организация закрытого и открытого доступа, Практический пример организации управления доступом, Передача объектов методам по ссылке, Способы передачи аргументов методу, Использование модификаторов параметров ref и out, Использование модификатора параметра ref, Использование модификатора параметра out, Использование модификаторов ref и out для ссылок на объекты, Использование переменного числа аргументов, Возврат объектов из методов, Возврат массива из метода, Перегрузка методов, Перегрузка конструкторов, Вызов перегружаемого конструктора с помощью ключевого слова this, Инициализаторы объектов, Необязательные аргументы, Необязательные аргументы и перегрузка методов, Необязательные аргументы и неоднозначность, Практический пример использования необязательных аргументов, Именованные аргументы, Метод Main(), Возврат значений из метода Main(), Передача аргументов методу Main(), Рекурсия, Применение ключевого слова static, Статические конструкторы, Статические классы

Тема 2. Перегрузка операторов

Основы перегрузки операторов, Перегрузка бинарных операторов, Перегрузка унарных операторов, Выполнение операций со встроенными в C# типами данных, Перегрузка операторов отношения, Перегрузка операторов true и false, Перегрузка логических операторов, Простой способ перегрузки логических операторов, Как сделать укороченные логические операторы доступными для применения, Операторы преобразования, Рекомендации и ограничения по перегрузке операторов, Пример перегрузки операторов

Тема 3. Индексаторы и свойства

Индексаторы, Создание одномерных индексаторов, Перегрузка индексаторов, Индексаторы без базового массива, Многомерные индексаторы, Свойства, Автоматически реализуемые свойства, Применение инициализаторов объектов в свойствах, Ограничения, присущие свойствам, Применение модификаторов доступа в аксессуарах, Применение индексаторов и свойств

Тема 4. Наследование

Основы наследования, Доступ к членам класса и наследование, Организация защищенного доступа, Конструкторы и наследование, Вызов конструкторов базового класса, Наследование и сокрытие имен, Применение ключевого слова base для доступа к скрытому имени, Создание многоуровневой иерархии классов, Порядок вызова конструкторов, Ссылки на базовый класс и объекты производных классов, Виртуальные методы и их переопределение, Что дает переопределение методов, Применение виртуальных методов, Применение абстрактных классов, Предотвращение наследования с помощью ключевого слова sealed, Класс object, Упаковка и распаковка, Класс object как универсальный тип данных

Тема 5. Интерфейсы, структуры и перечисления

Интерфейсы, Реализация интерфейсов, Применение интерфейсных ссылок, Интерфейсные свойства, Интерфейсные индексаторы, Наследование интерфейсов, Сокрытие имен при наследовании интерфейсов, Явные реализации, Выбор между интерфейсом и абстрактным классом, Стандартные интерфейсы для среды .NET Framework, Структуры, О назначении структур, Перечисления, Инициализация перечисления, Указание базового типа перечисления, Применение перечислений

Тема 6. Обработка исключительных ситуаций

Класс System.Exception, Основы обработки исключительных ситуаций, Применение пары ключевых слов try и catch, Пример обработки исключительной ситуации, Последствия перехвата исключений, Обработка исключительных ситуаций, Применение нескольких операторов catch, Перехват всех исключений, Вложение блоков try, Генерирование исключений вручную, Повторное генерирование исключений, Использование блока finally, Подробное рассмотрение класса Exception, Наиболее часто используемые исключения, Получение производных классов исключений, Перехват исключений производных классов, Применение ключевых слов checked и unchecked

Тема 7. Применение средств ввода-вывода

Организация системы ввода-вывода в C# на потоках, Байтовые и символьные потоки, Встроенные потоки, Классы потоков, Класс Stream, Классы байтовых потоков, Классы-оболочки символьных потоков, Двоичные потоки, Консольный ввод-вывод, Чтение данных из потока ввода с консоли, Применение метода ReadKey(), Запись данных в поток вывода на консоль, Класс FileStream и байтовый ввод-вывод в файл, Открытие и закрытие файла, Чтение байтов из потока файлового ввода-вывода, Запись в файл, Использование класса FileStream для копирования файла, Символьный ввод-вывод в файл, Применение класса StreamWriter, Применение класса StreamReader, Переадресация стандартных потоков, Чтение и запись двоичных данных, Класс BinaryWriter, Класс BinaryReader, Демонстрирование двоичного ввода-вывода, Файлы с произвольным доступом, Применение класса MemoryStream, Применение классов StringReader и StringWriter, Класс File, Копирование файлов с помощью метода Copy(), Применение методов Exists() и GetLastAccessTime(), Преобразование числовых строк в их внутреннее представление

Тема 8. Делегаты, события и лямбда-выражения

Делегаты, Групповое преобразование делегируемых методов, Применение методов экземпляра в качестве делегатов, Групповая адресация, Ковариантность и контравариантность, Класс System.Delegate, Назначение делегатов, Анонимные функции, Анонимные методы, Передача аргументов анонимному методу, Возврат значения из анонимного метода, Применение внешних переменных в анонимных методах, Лямбда-выражения, Лямбда-оператор, Одиночные лямбда-выражения, Блочные лямбда-выражения, События, Пример групповой адресации события, Методы экземпляра в сравнении со статическими методами в качестве обработчиков событий, Применение аксессоров событий, Разнообразные возможности событий, Применение анонимных методов и лямбда-выражений вместе с событиями, Рекомендации по обработке событий в среде .NET Framework, Применение делегатов EventHandler<TEventArgs> и EventHandler, Практический пример обработки событий

Тема 9. Пространства имен, препроцессор и сборки

Пространства имен, Объявление пространства имен, Предотвращение конфликтов имен с помощью пространств имен, Директива using, Вторая форма директивы using, Аддитивный характер пространств имен, Вложенные пространства имен, Глобальное пространство имен, Применение описателя псевдонима пространства имен ::, Препроцессор, Директива #define, Директивы #if и #endif, Директивы #else и #elif, Директива #undef Директива #error, Директива #warning, Директива #line, Директивы #region и #endregion, Директива #pragma, Сборки и модификатор доступа internal, Модификатор доступа internal

Тема 10. Динамическая идентификация типов, рефлексия и атрибуты

Динамическая идентификация типов, Проверка типа с помощью оператора is, Применение оператора as, Применение оператора typeof, Рефлексия, Класс System.Type - ядро подсистемы рефлексии, Применение рефлексии, Получение сведений о методах, Вызов методов с помощью рефлексии, Получение конструкторов конкретного типа, Получение типов данных из сборок, Полностью автоматизированное обнаружение типов, Атрибуты, Основы применения атрибутов, Сравнение позиционных и именованных параметров, Встроенные атрибуты, Атрибут AttributeUsage, Атрибут Conditional, Атрибут Obsolete

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

6.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения

Этап	Форма контроля	Оцениваемые компетенции	Темы (разделы) дисциплины
Семестр 5			
	<i>Текущий контроль</i>		
1	Лабораторные работы	ПК-2	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления
2	Устный опрос	ПК-2	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления
3	Тестирование	ПК-2	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления
	<i>Экзамен</i>	ПК-2	
Семестр 6			
	<i>Текущий контроль</i>		
1	Лабораторные работы	ПК-2	6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
2	Устный опрос	ПК-2	6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
3	Тестирование	ПК-2	6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
	<i>Экзамен</i>	ПК-2	

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

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Семестр 5					
Текущий контроль					

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Лабораторные работы	Оборудование и методы использованы правильно. Проявлена превосходная теоретическая подготовка. Необходимые навыки и умения полностью освоены. Результат лабораторной работы полностью соответствует её целям.	Оборудование и методы использованы в основном правильно. Проявлена хорошая теоретическая подготовка. Необходимые навыки и умения в основном освоены. Результат лабораторной работы в основном соответствует её целям.	Оборудование и методы частично использованы правильно. Проявлена удовлетворительная теоретическая подготовка. Необходимые навыки и умения частично освоены. Результат лабораторной работы частично соответствует её целям.	Оборудование и методы использованы неправильно. Проявлена неудовлетворительная теоретическая подготовка. Необходимые навыки и умения не освоены. Результат лабораторной работы не соответствует её целям.	1
Устный опрос	В ответе качественно раскрыты содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продемонстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.	Основные вопросы темы раскрыты. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продемонстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема частично раскрыта. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема не раскрыта. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.	2
Тестирование	86% правильных ответов и более.	От 71% до 85 % правильных ответов.	От 56% до 70% правильных ответов.	55% правильных ответов и менее.	3
Экзамен	Обучающийся обнаружил всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой дисциплины, усвоил взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявил творческие способности в понимании, изложении и использовании учебно-программного материала.	Обучающийся обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные программой задания, усвоил основную литературу, рекомендованную программой дисциплины, показал систематический характер знаний по дисциплине и способен к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.	Обучающийся обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справился с выполнением заданий, предусмотренных программой, знаком с основной литературой, рекомендованной программой дисциплины, допустил погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладает необходимыми знаниями для их устранения под руководством преподавателя.	Обучающийся обнаружил значительные пробелы в знаниях основного учебно-программного материала, допустил принципиальные ошибки в выполнении предусмотренных программой заданий и не способен продолжить обучение или приступить по окончании университета к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине.	
Семестр 6					
Текущий контроль					

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Лабораторные работы	Оборудование и методы использованы правильно. Проявлена превосходная теоретическая подготовка. Необходимые навыки и умения полностью освоены. Результат лабораторной работы полностью соответствует её целям.	Оборудование и методы использованы в основном правильно. Проявлена хорошая теоретическая подготовка. Необходимые навыки и умения в основном освоены. Результат лабораторной работы в основном соответствует её целям.	Оборудование и методы частично использованы правильно. Проявлена удовлетворительная теоретическая подготовка. Необходимые навыки и умения частично освоены. Результат лабораторной работы частично соответствует её целям.	Оборудование и методы использованы неправильно. Проявлена неудовлетворительная теоретическая подготовка. Необходимые навыки и умения не освоены. Результат лабораторной работы не соответствует её целям.	1
Устный опрос	В ответе качественно раскрыты содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продemonстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.	Основные вопросы темы раскрыты. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продemonстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема частично раскрыта. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема не раскрыта. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.	2
Тестирование	86% правильных ответов и более.	От 71% до 85 % правильных ответов.	От 56% до 70% правильных ответов.	55% правильных ответов и менее.	3
Экзамен	Обучающийся обнаружил всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой дисциплины, усвоил взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявил творческие способности в понимании, изложении и использовании учебно-программного материала.	Обучающийся обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные программой задания, усвоил основную литературу, рекомендованную программой дисциплины, показал систематический характер знаний по дисциплине и способен к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.	Обучающийся обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справился с выполнением заданий, предусмотренных программой, знаком с основной литературой, рекомендованной программой дисциплины, допустил погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладает необходимыми знаниями для их устранения под руководством преподавателя.	Обучающийся обнаружил значительные пробелы в знаниях основного учебно-программного материала, допустил принципиальные ошибки в выполнении предусмотренных программой заданий и не способен продолжить обучение или приступить по окончании университета к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине.	

6.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Приложение. Развёрнутое содержание оценочных средств - в прикреплённом файле
[F_1031248456/B1.V.03._Obektno_orientirovannoe_programmirovaniye.pdf](https://shelly.kpfu.ru/pls/student/docs/F_1031248456/B1.V.03._Obektno_orientirovannoe_programmirovaniye.pdf)

Семестр 5

Текущий контроль

1. Лабораторные работы

Темы 1, 2, 3, 4, 5

Тема 1. Методы и классы.

Модификаторы public и private, Алгоритм предотвращения сбоев, Передача методам объектов в качестве параметров, Аргументы простых типов передаются по значению, Передача объектов по ссылке, Модификатор ref для передачи методу значения обычного типа по ссылке, Замена местами значения двух переменных, Модификатор параметра out, Модификатор ref при вызове по ссылке, Модификатор params, Обычный параметр и параметр переменной длины, Возврат объектов, Возврат объектов, определенных пользователем, Перегрузка метода, Автоматическое преобразование типов аргументов, Перегрузка конструктора, Один объект используется для инициализации другого, Ключевое слово this для вызова одного конструктора из другого, Отображение аргументов, заданных в командной строке, Простой автоматизированный телефонный справочник, Рекурсия для вычисления факториала числа, Модификатор static.

Тема 2. Перегрузка операторов.

Перегруженные операторы, Перегрузка унарного ?, Перегрузка унарного ++, Перегрузка бинарного оператора +, Перегрузка операторов "<" и ">".

Тема 3. Индексаторы и свойства.

Индексатор для доступа к элементам массива, Индексаторы, работающие не с массивами, Двухмерный индексатор, Простое свойство, Свойство Length.

Тема 4. Наследование.

Простая иерархия классов, Новая характеристика в классе ? наследнике, Свойства для доступа к закрытым членам класса, Модификатор protected, Конструктор в классе ? наследнике, Конструктор в базовом классе, Скрытие переменной при наследовании, Ключевое слово base для доступа к скрытой переменной, Вызов скрытого метода с помощью ключевого слова base, Многоуровневая иерархия классов, Порядок вызова конструкторов, Ссылка переменной наследуемого класса на объект наследующего класса, Передача ссылочной переменной наследуемого класса ссылки на объект наследующего класса, Виртуальный метод, Виртуальные методы и полиморфизм, Абстрактный класс, Переопределение метода ToString(), Упаковка и распаковка переменной, Упаковка при передаче методу значений в качестве аргументов.

Тема 5. Интерфейсы, структуры и перечисления.

Реализация интерфейса, Интерфейсные ссылки, Свойства в интерфейсе, Индексатор в интерфейсе, Наследование интерфейсов, Явная реализация члена интерфейса, Явная реализация с целью устранения неоднозначности, Возможности структуры, Возможности перечисления.

https://shelly.kpfu.ru/pls/student/docs/F_1031248456/B1.V.03._Obektno_orientirovannoe_programmirovaniye.pdf

2. Устный опрос

Темы 1, 2, 3, 4, 5

Темы 1, 2, 3, 4

1. Модификаторы доступа
2. Организация закрытого и открытого доступа
3. Способы передачи аргументов методу
4. Использование модификатора параметра ref
5. Использование модификатора параметра out
6. Использование модификаторов ref и out для ссылок на объекты
7. Возврат массива из метода
8. Вызов перегружаемого конструктора с помощью ключевого слова this
9. Необязательные аргументы и перегрузка методов
10. Необязательные аргументы и неоднозначность
11. Возврат значений из метода Main()
12. Передача аргументов методу Main()
13. Статические конструкторы
14. Статические классы
15. Перегрузка бинарных операторов

16. Перегрузка унарных операторов
17. Простой способ перегрузки логических операторов
18. Создание одномерных индексаторов
19. Перегрузка индексаторов
20. Индексаторы без базового массива
21. Многомерные индексаторы
22. Автоматически реализуемые свойства
23. Применение инициализаторов объектов в свойствах
24. Ограничения, присущие свойствам
25. Организация защищенного доступа
26. Вызов конструкторов базового класса
27. Применение ключевого слова base для доступа к скрытому имени
28. Что дает переопределение методов
29. Применение виртуальных методов
30. Упаковка и распаковка
31. Класс object как универсальный тип данных
32. Реализация интерфейсов
33. Назначение структур
34. Инициализация перечисления
35. Указание базового типа перечисления
36. Применение перечислений

3. Тестирование

Темы 1, 2, 3, 4, 5

1. Исходный модуль содержит код:

```
internal class Point
{
    public double x,y;
    public Point(double x, double y) { this.x = x; this.y = y; } public static void Move(double d){ x+=d; y+=d;}
}
class Program {
    static void Main() {
        Point p = new Point(1,1); Point.Move(2);
    }
}
```

Метод Main разрабатывался для перемещения точки по обеим координатам на 2. Трансляция программы завершена неудачно. Из приведенных высказываний укажите высказывание, которое раскрывает причину некорректности программы:

тип Point не доступен из класса Program

значение полей x и y не определено

метод Move не может обращаться к полям объекта

в классе Point не определен конструктор без параметров

метод Move должен вызываться как метод объекта, а не метод класса

2. Исходный модуль содержит определения:

```
struct Лампочка {
    public int мощность;
    public Лампочка(int мощность){this.мощность=мощность;} } class Прибор {
    public Лампочка[] лам;
    public Прибор(Лампочка[] лам) { this.лам = лам;} public int Мощность
    { get {
        int s=0;
        for (int i = 0; i < лам.Length; i++) s += лам[i].мощность; return s; }
    }
}
```

В классе Program определен метод Main, в теле которого находится код: Лампочка л1 = new Лампочка(50);

Лампочка л2 = new Лампочка(100); Лампочка[] лам = new Лампочка[] { л1, л2 }; Прибор пр1=new Прибор(лам);

л2 = л1;

Прибор пр2=new Прибор(лам); System.Console.WriteLine(пр1.Мощность + пр2.Мощность);

Укажите результат вывода на консоль после выполнения метода Main():

300

250

400

100

200

3. Проанализируйте приведенный код.

```

delegate void MyEventHandler(ref int a);
class MyEvent
{ public event MyEventHandler SomeEvent; public void OnSomeEvent(ref int d)
  { if (SomeEvent != null) SomeEvent(ref d); } } class EventTest
{ static void Handler1(ref int b) { b = 2 * b; }
static void Handler2(ref int c) { c = c + 12; Handler1(ref c);} public static void Main()
{ MyEvent evt = new MyEvent();
int a = int.Parse(System.Console.ReadLine());
if (a > 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler1);
else if (a < 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler2); evt.OnSomeEvent(ref a);
System.Console.WriteLine(a);
}}

```

Пользователь собирается после запуска программы ввести число -5. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

программа завершится аварийно

7

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

14

-5

4. Приведенному ниже коду доступно пространство имен System.

```

class ExceptA : ApplicationException { }
class ExceptB : ExceptA { }

public static void Main()
{ string a = Console.ReadLine(); string b = Console.ReadLine(); try { if (a == "end") throw new ExceptA();
else if (b == "exit") throw new ExceptB(); else throw new Exception();
}
catch (ExceptB) { Console.Write(" Исключение-1"); } catch (ExceptA) { Console.Write(" Исключение-2"); } catch
(Exception){ Console.Write(" Исключение-3"); }
}

```

Пользователь собирается запустить программу и ввести cont и exit. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Исключение-1 Исключение-3

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

Исключение-1

Исключение-3

Исключение-1 Исключение-2

5. Программа предназначена для присвоения переменной t класса B значения 8:

```

class A
{ static public B b = null; public A()
{ B b = new B(); } public class B
{ static public int r; public int t;
} }
class Test
{ public static void Main() { A a = new A();
_____ }
}

```

Среди приведенных ниже кодов укажите код, которым необходимо заменить знаки подчеркивания для получения корректной программы, решающей поставленную задачу:

B.t = 8;

A.b.t = 8;

a.B.t = 8;

a.b.t = 8;

A.B.t = 8;

6. Проанализируйте приведенный код.

```
struct Auto { public string марка;
public Auto(string m) { марка = m; }
}
class Водитель
{ private string фамилия; private Auto марка;
public Водитель(string фам, Auto м) { фамилия = фам; марка = м; } public override string ToString()
{ return string.Format("{0} имеет {1}",фамилия, марка.марка) ; } }
class Program
{ static void Main(string[] args)
{ Водитель[] водители = new Водитель[2]; Auto auto1 = new Auto("Мерседес");
Auto auto2 = new Auto("Тойота"); auto2 = auto1; водители[0] = new Водитель("Иванов", auto1); водители[1] = new
Водитель("Петров", auto2); System.Console.WriteLine(водители[1]);
} }

```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Иванов имеет Мерседес

Петров имеет Форд

Петров имеет Мерседес

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

программа завершится аварийно

7. Исходный модуль содержит определение типов Газета, Журнал и Program:

```
struct Газета
{ public int Страницы; } struct Журнал
{
public int Страницы;
public override string ToString() { return Страницы.ToString(); } }
В классе Program определен метод Main, в теле которого находится код: Газета a; Журнал b;
a.Страницы=20; b.Страницы =30; System.Console.Write(a.ToString() + " " + b.ToString());

```

Укажите результат вывода на консоль после выполнения метода Main():

Газета 30

при запуске программы фиксируется ошибка: в структуре невозможно переопределить метод ToString()

Газета Журнал

20 Журнал

50

8. Программа предназначена для вычисления и вывода квадрата числа 3:

```
public delegate double F( double x);
class A { public int F(int x) { return x * x; } }
class B { public static double F(double x) { return x * x; } } class Program
{
static void Main() {
F f = new F( _____ ); System.Console.WriteLine(f(3));
} }

```

Среди перечисленных ниже кодов укажите код, подстановка которого вместо знаков подчеркивания позволяет получить корректную программу, решающую эту задачу:

A.F

B.F

new A().F

B.F()

new B().F

9. Исходный модуль содержит код:

```
class Room
{
private int nom1; public int nom2;
public Room(int a, int b){ nom1 = a; nom2 = b; } public int Nom

```

```

{
get { return nom1 * 100 + nom2; }
set { if (value<5555) {nom1 = value/100; nom2 = value%100; } }
}
class Program {
static void Main() {
Room r = new Room(12,45); r.Nom = 4431;
System.Console.WriteLine("{0} ", r.Nom); }
}

```

Укажите результат вывода на консоль после выполнения метода Main():

1245
4431
4445
1231

10. Исходный модуль содержит определения:

```

interface ISмена { void Сменить(); }
class Переключатель: ISмена {
bool s;
public Переключатель(bool s){this.s = s;}
public void Сменить() { s = !s; System.Console.WriteLine(" " + s); } }
class Счетчик : ISмена {
int s;
public Счетчик(int s) { this.s = s; }
public void Сменить() { s--; System.Console.WriteLine(" " + s); } }

```

В этом же модуле определен класс Program с методом Main(), в котором содержится следующий код:

```
ISмена a = new Переключатель(true); ISмена b = new Счетчик(9),c;
```

```
c = b; a.Сменить(); b.Сменить(); b = a; b.Сменить(); b = c;
```

```
b.Сменить(); Укажите результат вывода на консоль после выполнения метода Main():
```

при запуске фиксируется синтаксическая ошибка, указывающая на несовместимость типов при объявлении переменных в методе Main()

False 8 7 6
8 7 6 False
True 8 False 7
False 8 True 7

11. Проанализируйте приведенный код.

```

struct Auto { public string марка;
public Auto(string m) { марка = m; } }
class Водитель
{ private string фамилия; private Auto марка;
public Водитель(string фам, Auto м) { фамилия = фам; марка = м; }
public override string ToString()
{ return string.Format("{0}у нравится {1}",фамилия, марка.марка) ; } }
class Program
{ static void Main(string[] args)
{ Водитель[] водители = new Водитель[2]; Auto auto1 = new Auto("Мерседес"); Auto auto2 = new Auto("Тойота");
водители[0] = new Водитель("Иванов", auto1);
водители[1] = new Водитель("Петров", auto2); auto2.марка = "Форд"; System.Console.WriteLine(водители[1]);
} }

```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

программа завершится аварийно

Петрову нравится Форд

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

Петрову нравится Тойота

Иванову нравится Мерседес

12. Исходный модуль содержит код:

```
class Сотрудник
```

```

{
private int id;
private float зарплата;
public Сотрудник(int i, float z) { id = i; зарплата = z; }
public float Зарплата(int номер) { return номер < id ? зарплата : -1; } public void Зарплата(int номер, ref float зарпл)
{ if (номер < id)
{ float зрпл = зарплата;
зарплата = зарпл; зарпл = зрпл; } } }
class Program
{ static void Main()
{ Сотрудник Иванов = new Сотрудник(12, 40000.0f); float z = 45000.0f; Иванов.Зарплата(10, ref z);
System.Console.WriteLine("Зарплата Иванова = " + z);
} }

```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

```

40000
45000
программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
20000
-1

```

13. Проанализируйте приведенный код.

```

delegate void MyEventHandler(ref int a);
class MyEvent
{ public event MyEventHandler SomeEvent; public void OnSomeEvent(ref int d)
{ if (SomeEvent != null) SomeEvent(ref d); } } class EventTest
{ static void Handler1(ref int b) { b = 2 * b; }
static void Handler2(ref int c) { c = c + 12; Handler1(ref c); } public static void Main()
{ MyEvent evt = new MyEvent();
int a = int.Parse(System.Console.ReadLine());
if (a > 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler1);
else if (a < 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler2); evt.OnSomeEvent(ref a);
System.Console.WriteLine(a); } }

```

Пользователь собирается после запуска программы ввести цифру 0. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

```

0
программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
программа завершится аварийно
24
12

```

14. Программа предназначена для присвоения переменной r класса B значения 4 : class A

```

{ public B b = null; public A()
{ B b = new B(); } public class B
{ static public int r; public int t;
} }
class Test
{ public static void Main() { A a = new A();
_____ }
}

```

Среди приведенных ниже кодов укажите код, которым необходимо заменить знаки подчеркивания для получения корректной программы, решающей поставленную задачу:

```

B.r = 4;
a.B.r = 4;
A.b.r = 4;
A.B.r = 4;
a.b.r = 4;

```

15. В классе Test определен приведенный ниже код:

```
public delegate int CallBackMethod(int a1, int a2);
public static void Main() { int s = 0;
s = CallMethod(new CallBackMethod(CBMethod));
System.Console.WriteLine(s);
}
public static int CallMethod(CallBackMethod cbMethod)
{ return cbMethod(11, 22); }
public static int CBMethod(int a1, int a2)
{ return a1 + a2; }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

программа завершится аварийно
 программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
 33
 Test.CallMethod
 0;

16. Приведенному ниже коду доступно пространство имен System.

```
class ExceptA : ApplicationException { }
class ExceptB : ExceptA { } public static void Main()
{ string a = Console.ReadLine(); string b = Console.ReadLine(); try { if (a == "end") throw new ExceptA();
else if (b == "exit") throw new ExceptB(); else throw new Exception();
}
catch (ExceptB) { Console.Write(" Исключение-1"); } catch (ExceptA) { Console.Write(" Исключение-2"); } catch
(Exception){ Console.Write(" Исключение-3"); }
}
```

Пользователь собирается после запуска программы ввести end и cont. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Исключение-2 Исключение-3
 Исключение-3
 программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
 Исключение-2
 Исключение-1 Исключение-2

17. Сборка Assembly2.cs, приведенная ниже:

```
class Test3
{ static void Main()
{ ClassUser user = new ClassUser();
int count = user.count; System.Console.Write("count=" + count); }
}
```

ссылается на подключенную сборку Assembly1.cs, которая имеет следующий код: internal class ClassUser
 { public int count = 10; }

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

count=0
 программа не выполнится, так как сборка Assembly2.cs создает объект user, в то время как класс ClassUser не принадлежит сборке Assembly2.cs
 count=10
 будет выдано сообщение об ошибке компиляции, так как класс ClassUser недоступен в соответствии с его модификатором доступа

18. Исходный модуль содержит код классов ЧЕЛОВЕК,СОТРУДНИК и Program: class ЧЕЛОВЕК {
 string фамилия;
 public ЧЕЛОВЕК(string фамилия) {this.фамилия = фамилия;} public string ФАМ { get {return фамилия; } } }
 class СОТРУДНИК:ЧЕЛОВЕК { string должность;
 public СОТРУДНИК(string фамилия, string должность):base(фамилия) { this.должность = должность; }
 public string ДОЛ() { return должность; } }

В классе Program определен метод Main, в теле которого находится код: СОТРУДНИК P = new
 СОТРУДНИК("Петров","Кассир"); System.Console.Write("{0} {1}", _____);

Требуется вывести фамилию и должность сотрудника. Среди перечисленных ниже кодов укажите код, подстановка которого вместо знаков подчеркивания позволяет получить программу, решающую поставленную задачу:

Р.ФАМ(), Р.ДОЛ
 Р.фамилия, Р.ДОЛ()
 Р.ФАМ, ((ЧЕЛОВЕК)Р).ДОЛ()
 Р.ФАМ, Р.ДОЛ()

19. Проанализируйте приведенный код.

```
abstract class Rectangle
{ public double x=10, y=20;
  public Rectangle(double x, double y) { this.x = x; this.y = y; } abstract public int Zoom(int d);
  public double Zoom(double d) { x += d; y += d; return x * y; } }
class Параллелепипед : Rectangle { public double z;
  public Параллелепипед(double x, double y, double z) : base (x, y) { this.z = z; } }
class Program
{ static void Main()
{ Параллелепипед парал = new Параллелепипед(6, 8, 10); парал.Zoom(-5);
  System.Console.WriteLine("{0} {1} {2}", парал.x, парал.y, парал.z); } }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

будет обнаружена синтаксическая ошибка, связанная с тем, что класс Параллелепипед не переопределяет абстрактный метод Zoom

1 3 10
 1 3 5
 5 15 10

программа завершится аварийно

20. Приведенному ниже коду доступны пространства имен System и System.Collections.

```
public static void Main()
{ ArrayList al = new ArrayList(); al.Add("Иванов"); al.Add("Петров"); al.Add(3.1415f); al.Add("Ларин");
  foreach (string str in al) Console.WriteLine(str + " ");
}
```

21. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Иванов Петров Ларин

программа завершится аварийно в момент добавления в коллекцию числа вместо фамилии
 программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
 программа завершится аварийно в цикле foreach

Иванов Петров 3,1415 Ларин

22. Пусть процедура ItWorks имеет следующий заголовок: public void ItWorks(int [] A). Какой массив может быть передан ей в качестве аргумента?

string[] s
 Person[] p; (Person ? класс, определенный пользователем).
 byte[] y
 int[] x

23. Перечисление ? это

абстрактный класс
 класс без собственных полей
 класс без собственных методов

24. Правда ли, что:

обработчику события передаются входные аргументы, характеризующие событие, а он может передать объекту, создавшему событие, выходные аргументы
 класс, обрабатывающий события, может отсоединить обработчики событий других классов, чтобы они не мешали его работе
 делегат, задающий тип события, может иметь произвольную сигнатуру

25. Отметьте истинные высказывания

поля структуры не могут быть ссылками
структуры и перечисления ? частные случаи классов
структура наследует методы класса object
перечисление наследует методы класса object

Экзамен

Вопросы к экзамену:

1. Управление доступом к членам класса
2. Передача объектов методам по ссылке
3. Использование модификаторов параметров ref и out
4. Использование переменного числа аргументов
5. Возврат объектов из методов
6. Перегрузка методов
7. Перегрузка конструкторов
8. Инициализаторы объектов
9. Необязательные аргументы
10. Именованные аргументы
11. Метод Main()
12. Применение ключевого слова static
13. Основы перегрузки операторов
14. Выполнение операций со встроенными в C# типами данных
15. Перегрузка операторов отношения
16. Перегрузка операторов true и false
17. Перегрузка логических операторов
18. Операторы преобразования
19. Индексаторы
20. Свойства
21. Применение модификаторов доступа в аксессорах
22. Применение индексаторов и свойств
23. Основы наследования
24. Доступ к членам класса и наследование
25. Конструкторы и наследование
26. Наследование и сокрытие имен
27. Создание многоуровневой иерархии классов
28. Порядок вызова конструкторов
29. Ссылки на базовый класс и объекты производных классов
30. Виртуальные методы и их переопределение
31. Применение абстрактных классов
32. Предотвращение наследования с помощью ключевого слова sealed
33. Класс object
34. Интерфейсы
35. Применение интерфейсных ссылок
36. Интерфейсные свойства
37. Интерфейсные индексаторы
38. Наследование интерфейсов
39. Сокрытие имен при наследовании интерфейсов
40. Явные реализации
41. Выбор между интерфейсом и абстрактным классом
42. Стандартные интерфейсы для среды .NET Framework
43. Структуры
44. Перечисления

Семестр 6

Текущий контроль

1. Лабораторные работы

Темы 6, 7, 8, 9, 10

Тема 6. Обработка исключительных ситуаций.

Обработка исключений, Генерация исключения в одном методе, а перехват его другим методом, Обработка ошибок системой выполнения C#, Обработка ошибок и продолжение выполнения программы, Несколько операторов catch, Оператор catch, перехватывающий все ошибки, Вложенный блок try, Генерирование исключения оператором throw, Повторное генерирование исключения, Блок finally, Члены класса Exception, Заказное исключение, Исключения наследующего класса должны следовать перед исключениями наследуемого класса, Использование ключевых слов checked и unchecked, Использование ключевых слов checked и unchecked при работе с блоками операторов.

Тема 7. Применение средства ввода-вывода.

Ввод с консоли методом ReadLine(), Поток Console.In для чтения массива байтов с клавиатуры, Запись данных в потоки Console.Out и Console.Error, Отображение содержимого файла, Копирование данных из файла в файл, Класс StreamWriter, Открытие файла с помощью класса StreamWriter, Возможности класса StreamReader, Перенаправление Console.Out, Запись и считывание двоичных данных, Произвольный доступ к содержимому файла, Расчет среднего значения введенных чисел.

Тема 8. Делегаты, события и лямбда выражения.

Объявление делегата, Делегаты ссылаются на методы экземпляра, Многоадресность делегата, Использование события, Широковещательное событие, Реагирование обработчиков каждого из объектов класса на событие.

Тема 9. Пространства имен, процессор и сборки.

Объявление пространства имен, Директива using, Использование псевдонима, Аддитивность пространств имен, Вложенное пространство имен, Оператор преобразования, Оператор явного преобразования.

Тема 10. Динамическая идентификация типов, рефлексия и атрибуты.

Директивы препроцессора #if, #endif и #define, Символьное выражение, Директива препроцессора #else, Директива препроцессора #elif, Условный атрибут Conditional, Атрибут Obsolete, Указатель и ключевое слово unsafe,

2. Устный опрос

Темы 6, 7, 8, 9, 10

Темы 5, 6, 7, 8, 9, 10

1. Применение ключевых слов try и catch
2. Обработка исключительной ситуации
3. Повторное генерирование исключений
4. Наиболее часто используемые исключения
5. Байтовые и символьные потоки
6. Встроенные потоки
7. Класс Stream
8. Классы байтовых потоков
9. Двоичные потоки
10. Чтение данных из потока ввода с консоли
11. Запись данных в поток вывода на консоль
12. Открытие и закрытие файла
13. Чтение байтов из потока файлового ввода-вывода
14. Запись в файл
15. Групповое преобразование делегируемых методов
16. Применение методов экземпляра в качестве делегатов
17. Групповая адресация
18. Ковариантность и контравариантность
19. Назначение делегатов
20. Передача аргументов анонимному методу
21. Возврат значения из анонимного метода
22. Применение внешних переменных в анонимных методах
23. Лямбда-оператор
24. Применение аксессоров событий
25. Разнообразные возможности событий
26. Объявление пространства имен
27. Директива using
28. Аддитивный характер пространств имен
29. Вложенные пространства имен
30. Глобальное пространство имен
31. Применение описателя псевдонима пространства имен ::
32. Директива #define, Директивы #if и #endif, Директивы #else и #elif

33. Проверка типа с помощью оператора is
34. Применение оператора as
35. Применение оператора typeof
36. Получение сведений о методах
37. Вызов методов с помощью рефлексии
38. Получение конструкторов конкретного типа
39. Получение типов данных из сборок
40. Основы применения атрибутов
41. Сравнение позиционных и именованных параметров
42. Атрибут AttributeUsage, Атрибут Conditional, Атрибут Obsolete

3. Тестирование

Темы 6, 7, 8, 9, 10

1. Исходный модуль содержит определения:

```
struct Лампочка {
public int мощность;
public Лампочка(int мощность){this.мощность=мощность;} }
class Прибор {
public Лампочка[] лам;
public Прибор(Лампочка[] лам) { this.лам = лам;}
public int Мощность
{ get {
int s=0;
for (int i = 0; i < лам.Length; i++) s += лам[i].мощность;
return s; }
}}

```

В классе Program определен метод Main, в теле которого находится код:

```
Лампочка л1 = new Лампочка(50);
Лампочка л2 = new Лампочка(100);
Лампочка[] лам = new Лампочка[] { л1, л2 };
Прибор пр1=new Прибор(лам);
л2 = л1;
Прибор пр2=new Прибор(лам);
System.Console.WriteLine(пр1.Мощность + пр2.Мощность);

```

Укажите результат вывода на консоль после выполнения метода Main():

- 300
- 250
- 400
- 100
- 200

2. Проанализируйте приведенный код.

```
delegate void MyEventHandler(ref int a);
class MyEvent
{ public event MyEventHandler SomeEvent;
public void OnSomeEvent(ref int d)
{ if (SomeEvent != null) SomeEvent(ref d); } }
class EventTest
{ static void Handler1(ref int b) { b = 2 * b; }
static void Handler2(ref int c) { c = c + 12; Handler1(ref c);}
public static void Main()
{ MyEvent evt = new MyEvent();
int a = int.Parse(System.Console.ReadLine());
if (a > 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler1);
else if (a < 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler2);
evt.OnSomeEvent(ref a); System.Console.WriteLine(a);
}}

```

Пользователь собирается после запуска программы ввести число -5. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

программа завершится аварийно

7

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

14

-5

3. Приведенному ниже коду доступно пространство имен System.

```
class ExceptA : ApplicationException { }
class ExceptB : ExceptA { }
public static void Main()
{ string a = Console.ReadLine(); string b = Console.ReadLine();
try { if (a == "end") throw new ExceptA();
else if (b == "exit") throw new ExceptB();
else throw new Exception();
}
catch (ExceptB) { Console.Write(" Исключение-1"); }
catch (ExceptA) { Console.Write(" Исключение-2"); }
catch (Exception){ Console.Write(" Исключение-3"); }
}
```

Пользователь собирается запустить программу и ввести cont и exit. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Исключение-1 Исключение-3

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

Исключение-1

Исключение-3

Исключение-1 Исключение-2

4. Программа предназначена для присвоения переменной t класса B значения 8 :

```
class A
{ static public B = null;
public A()
{ B = new B(); }
public class B
{ static public int r;
public int t;
} }
class Test
{ public static void Main()
{ A = new A();
}
}
```

Среди приведенных ниже кодов укажите код, которым необходимо заменить знаки подчеркивания для получения корректной программы, решающей поставленную задачу:

B.t = 8

A.b.t = 8

a.B.t = 8

a.b.t = 8

A.B.t = 8

5. Проанализируйте приведенный код.

```
struct Auto { public string марка;
public Auto(string m) { марка = m; }
}
class Водитель
{ private string фамилия; private Auto марка;
public Водитель(string фам, Auto м) { фамилия = фам; марка = м; }
public override string ToString()
{ return string.Format("{0} имеет {1}",фамилия, марка.марка) ; }
}
class Program
```

```

{ static void Main(string[] args)
{ Водитель[] водители = new Водитель[2];
Auto auto1 = new Auto("Мерседес");
Auto auto2 = new Auto("Тойота"); auto2 = auto1;
водители[0] = new Водитель("Иванов", auto1);
водители[1] = new Водитель("Петров", auto2);
System.Console.WriteLine(водители[1]);
} }

```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Иванов имеет Мерседес

Петров имеет Форд

Петров имеет Мерседес

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

программа завершится аварийно

6. Программа предназначена для вычисления и вывода квадрата числа 3:

```

public delegate double F( double x);
class A { public int F(int x) { return x * x; } }
class B { public static double F(double x) { return x * x; } }
class Program
{
static void Main()
{
F = new F( _____ );
System.Console.WriteLine(f(3));
}
}

```

Среди перечисленных ниже кодов укажите код, подстановка которого вместо знаков подчеркивания позволяет получить корректную программу, решающую эту задачу:

A.F

B.F

new A().F

B.F()

new B().F

7. Исходный модуль содержит код:

```

class Room
{
private int nom1;
public int nom2;
public Room(int a, int b){ nom1 = a; nom2 = b; }
public int Nom
{
get { return nom1 * 100 + nom2; }
set { if (value<5555) {nom1 = value/100; nom2 = value%100; }}
}
}
class Program
{
static void Main()
{
Room r = new Room(12,45);
r.Nom = 4431;
System.Console.Write("{0} ", r.Nom);
}
}

```

Укажите результат вывода на консоль после выполнения метода Main():

1245

4431

4445

1231

8. Исходный модуль содержит определения:

```
interface IСмена { void Сменить(); }
class Переключатель: IСмена
{
bool s;
public Переключатель(bool s){this.s = s;}
public void Сменить() { s = !s; System.Console.Write(" " + s); }
}
class Счетчик : IСмена
{
int s;
public Счетчик(int s) { this.s = s; }
public void Сменить() { s--; System.Console.Write(" " + s); }
}
```

В этом же модуле определен класс Program с методом Main(), в котором содержится следующий код:

```
IСмена a = new Переключатель(true);
IСмена b = new Счетчик(9),c;
c = b; a.Сменить(); b.Сменить(); b = a; b.Сменить(); b = c; b.Сменить();
```

Укажите результат вывода на консоль после выполнения метода Main():

при запуске фиксируется синтаксическая ошибка, указывающая на несовместимость типов при объявлении переменных в методе Main()

False 8 7 6

8 7 6 False

True 8 False 7

False 8 True 7

9. Проанализируйте приведенный код.

```
struct Auto { public string марка;
public Auto(string m) { марка = m; }
}
class Водитель
{ private string фамилия; private Auto марка;
public Водитель(string фам, Auto м) { фамилия = фам; марка = м; }
public override string ToString()
{ return string.Format("{0}у нравится {1}",фамилия, марка.марка) ; }
}
class Program
{ static void Main(string[] args)
{ Водитель[] водители = new Водитель[2];
Auto auto1 = new Auto("Мерседес");
Auto auto2 = new Auto("Тойота");
водители[0] = new Водитель("Иванов", auto1);
водители[1] = new Водитель("Петров", auto2); auto2.марка = "Форд";
System.Console.WriteLine(водители[1]);
} }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

программа завершится аварийно

Петрову нравится Форд

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

Петрову нравится Тойота

Иванову нравится Мерседес

10. Исходный модуль содержит код:

```
class Сотрудник
{
private int id;
private float зарплата;
public Сотрудник(int i, float z) { id = i; зарплата = z; }
public float Зарплата(int номер) { return номер < id ? зарплата : -1; }
public void Зарплата(int номер, ref float зарпл)
```

```
{ if (номер < id)
{ float зрпл = зарплата;
зарплата = зарпл; зарпл = зрпл;
}} }
class Program
{ static void Main()
{ Сотрудник Иванов = new Сотрудник(12, 40000.0f);
float z = 45000.0f; Иванов.Зарплата(10, ref z);
System.Console.WriteLine("Зарплата Иванова = " + z);
}} }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

40000
45000
программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
20000
-1

11. Проанализируйте приведенный код.

```
delegate void MyEventHandler(ref int a);
class MyEvent
{ public event MyEventHandler SomeEvent;
public void OnSomeEvent(ref int d)
{ if (SomeEvent != null) SomeEvent(ref d); } }
class EventTest
{ static void Handler1(ref int b) { b = 2 * b; }
static void Handler2(ref int c) { c = c + 12; Handler1(ref c);}
public static void Main()
{ MyEvent evt = new MyEvent();
int a = int.Parse(System.Console.ReadLine());
if (a > 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler1);
else if (a < 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler2);
evt.OnSomeEvent(ref a); System.Console.WriteLine(a); } }
```

Пользователь собирается после запуска программы ввести цифру 0. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

0
программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок
программа завершится аварийно
24
12

12. Программа предназначена для присвоения переменной r класса B значения 4 :

```
class A
{ public B = null;
public A()
{ B = new B(); }
public class B
{ static public int r;
public int t;
} }
class Test
{ public static void Main()
{ A = new A();
}
}
```

Среди приведенных ниже кодов укажите код, которым необходимо заменить знаки подчеркивания для получения корректной программы, решающей поставленную задачу:

B.r = 4
a.B.r = 4

A.b.r = 4
 A.B.r = 4
 a.b.r = 4

13. Приведенному ниже коду доступно пространство имен System.

```
class ExceptA : ApplicationException { }
class ExceptB : ExceptA { }
public static void Main()
{ string a = Console.ReadLine(); string b = Console.ReadLine();
try { if (a == "end") throw new ExceptA();
else if (b == "exit") throw new ExceptB();
else throw new Exception();
}
catch (ExceptB) { Console.Write(" Исключение-1"); }
catch (ExceptA) { Console.Write(" Исключение-2"); }
catch (Exception){ Console.Write(" Исключение-3"); }
}
```

Пользователь собирается после запуска программы ввести end и cont. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Исключение-2 Исключение-3
 Исключение-3

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

Исключение-2

Исключение-1 Исключение-2

14. Сборка Assembly2.cs, приведенная ниже:

```
class Test3
{ static void Main()
{ ClassUser user = new ClassUser();
int count = user.count; System.Console.Write("count=" + count);
}
}
```

ссылается на подключенную сборку Assembly1.cs, которая имеет следующий код:

```
internal class ClassUser
{ public int count = 10; }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

count=0

программа не выполнится, так как сборка Assembly2.cs создает объект user, в то время как класс ClassUser не принадлежит сборке Assembly2.cs

count=10

будет выдано сообщение об ошибке компиляции, так как класс ClassUser недоступен в соответствии с его модификатором доступа

15. Исходный модуль содержит код классов ЧЕЛОВЕК,СОТРУДНИК и Program:

```
class ЧЕЛОВЕК {
string фамилия;
public ЧЕЛОВЕК(string фамилия) {this.фамилия = фамилия;}
public string ФАМ { get {return фамилия; } } }
class СОТРУДНИК:ЧЕЛОВЕК {
string должность;
public СОТРУДНИК(string фамилия, string должность):base(фамилия)
{ this.должность = должность; }
public string ДОЛ() { return должность; } }
```

В классе Program определен метод Main, в теле которого находится код:

```
СОТРУДНИК P = new СОТРУДНИК("Петров","Кассир");
System.Console.Write("{0} {1}", _____ );
```

Требуется вывести фамилию и должность сотрудника. Среди перечисленных ниже кодов укажите код, подстановка которого вместо знаков подчеркивания позволяет получить программу, решающую поставленную задачу:

P.ФАМ(), P.ДОЛ

P.фамилия, P.ДОЛ()

P.ФАМ, ((ЧЕЛОВЕК)P).ДОЛ()
P.ФАМ, P.ДОЛ()

16. Проанализируйте приведенный код.

```
abstract class Rectangle
{ public double x=10, y=20;
  public Rectangle(double x, double y) { this.x = x; this.y = y; }
  abstract public int Zoom(int d);
  public double Zoom(double d) { x += d; y += d; return x * y; }
}
class Параллелепипед : Rectangle
{ public double z;
  public Параллелепипед(double x, double y, double z) : base (x, y) { this.z = z; }
}
class Program
{ static void Main()
{ Параллелепипед парал = new Параллелепипед(6, 8, 10);
  парал.Zoom(-5);
  System.Console.WriteLine("{0} {1} {2}", парал.x, парал.y, парал.z);
} }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение: будет обнаружена синтаксическая ошибка, связанная с тем, что класс Параллелепипед не переопределяет абстрактный метод Zoom

1 3 10

1 3 5

5 15 10

программа завершится аварийно

17. Исходный модуль содержит код:

```
internal class Rectangle
{ public double width, height;
  public void Zoom(double d)
  { width += d; height += d; }
  public int Rectangle(double width, double height)
  { this.width = width; this.height = height; return width * height; }
}
class Program
{ static void Main()
{ Rectangle p = new Rectangle(10, 20);
  p.Zoom(-5);
  System.Console.WriteLine("{0} {1}", p.width, p.height);
} }
```

Метод Main разрабатывался для изменения размера прямоугольника.

Трансляция программы завершена неудачно. Из приведенных высказываний укажите высказывание, которое раскрывает причину некорректности программы:

тип значения, возвращаемого методом Rectangle, должен быть double

конструктор не может возвращать значение

конструктор не может следовать за методами класса

метод Zoom должен вызываться как метод класса, а не метод объекта

статический метод Main не может обращаться к полям объекта

18. Исходный модуль содержит классы A, B и Program:

```
class A
{
  public virtual string F1(){ return "1"; }
  public virtual string F2(){ return "2"; }
}
class B:A
{
  public new string F1() { return "3"; }
  public override string F2() { return "4"; }
}
```

```

}
В классе Program определен метод Main, в теле которого находится код:
В p = new B();
System.Console.Write(p.F1()+ p.F2());
Укажите результат вывода на консоль после выполнения метода Main():
23
13
12
34
14

```

19. Модуль содержит ссылку на пространство имен System и определения классов:

```

class A {public virtual void Show(){Console.Write("SOS! ");}}
class B:A {public new void Show(){Console.Write("BCEM! ");} }
class C:A {public override void Show(){Console.Write("YPA! ");} }
В классе Program определен метод Main, в теле которого находится код:
ArrayList a = new ArrayList();
a.Add(new A()); a.Add(new B()); a.Add(new C());
for (int i=0; i<a.Count; i++)
if(a[i] is A)
((A)a[i]).Show();

```

Укажите результат вывода после запуска программы на компиляцию и выполнение:

```

SOS! SOS! YPA!
SOS! BCEM! YPA!
SOS! BCEM! SOS!

```

генерируется исключение, так как невозможно преобразовать типы ссылок
SOS! SOS! SOS!

20. Исходный модуль содержит код:

```

class Студент
{ string фамилия;
public string Фамилия { get { return фамилия; } }
public Студент(string ф) { фамилия = ф; }
}
class Магистр : Студент
{ int курс;
public int Курс { get { return курс; } }
public Магистр(string фам, int к) : base(фам) { this.курс = к; }
}
class Test
{ static void Main()
{ Студент Иванов = new Магистр("Петров",4);
System.Console.Write("{0} {1}", _____);
}
}

```

Требуется вывести фамилию и номер курса студента в виде: Петров 4. Среди перечисленных ниже кодов укажите код, подстановка которого вместо знаков подчеркивания позволяет получить программу, решающую поставленную задачу:

```

Иванов.Фамилия, ((Магистр) Иванов).Курс
Петров.Фамилия, ((Магистр)Петров).Курс
Петров.Фамилия, (Магистр)Петров.Курс
Иванов.Фамилия, ((Студент)Петров).Курс
Петров.Фамилия, Петров.Курс

```

21. Проанализируйте приведенный код.

```

interface Интерфейс1 { string Метод(); }
interface Интерфейс2 { string Свойство { get; } }
class Класс1 : Интерфейс1
{ public string Метод() { return "Класс1: есть Интерфейс1 "; }
}
class Класс2 : Класс1, Интерфейс2

```

```

{ public string Метод() { return "Класс2: есть Интерфейс1 "; }
public string Свойство { get { return "Класс2: есть Интерфейс2 "; } }
}
class Test
{ static string GetValue(object ob)
{ string str = "Найден интерфейс ";
if (ob is Интерфейс1) str = ((Интерфейс1)ob).Метод();
if (ob is Интерфейс2) { str += ((Интерфейс2)ob).Свойство; }
return str;
}
public static void Main()
{ Интерфейс1 объект1 = new Класс2();
System.Console.WriteLine(GetValue(объект1));
}}

```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Класс1: есть Интерфейс1 Класс2: есть Интерфейс2

Класс2: есть Интерфейс2.

Найден интерфейс

программа не выполняется, так как обнаружена синтаксическая ошибка, связанная с тем, что посредством интерфейсной ссылки объект1 типа Интерфейс1 нельзя вызвать Свойство, определенное в Интерфейс2

Класс2: есть Интерфейс1. Класс2: есть Интерфейс2

22. Проанализируйте приведенный код.

```

delegate void MyEventHandler(ref int a);
class MyEvent
{ public event MyEventHandler SomeEvent;
public void OnSomeEvent(ref int d)
{ if (SomeEvent != null) SomeEvent(ref d); } }
class EventTest
{ static void Handler1(ref int b) { b = 2 * b; }
static void Handler2(ref int c) { c = c + 12; }
public static void Main()
{ MyEvent evt = new MyEvent();
int a = int.Parse(System.Console.ReadLine());
if (a > 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler1);
else if (a < 0) evt.SomeEvent += new MyEventHandler(EventTest.Handler2);
evt.OnSomeEvent(ref a); System.Console.WriteLine(a);
}}

```

Пользователь собирается после запуска программы ввести цифру 5. Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

22

программа не выполняется, так как обнаружена одна или несколько синтаксических ошибок

10

15

программа завершится аварийно

23. Исходный модуль содержит код:

```

class ЧЕЛОВЕК
{
class ПАСПОРТ {}
class ДОЛЖНОСТЬ {}
}
class Program
{
static void Main() { _____ P; }
}

```

В методе Main требуется объявить ссылку P на тип ДОЛЖНОСТЬ. Среди приведенных ниже вариантов ответов укажите истинный вариант:

вместо знаков подчеркивания укажите ЧЕЛОВЕК

вместо знаков подчеркивания укажите ДОЛЖНОСТЬ

вместо знаков подчеркивания укажите ЧЕЛОВЕК.ПАСПОРТ.ДОЛЖНОСТЬ

вместо знаков подчеркивания укажите ЧЕЛОВЕК.ДОЛЖНОСТЬ
объявить требуемую ссылку в методе Main невозможно

24. Проанализируйте приведенный код.

```
interface Интерфейс1 { string Метод(); }
interface Интерфейс2 { string Свойство {get;} }
class Класс1 : Интерфейс1
{ public string Метод() { return "Есть Интерфейс1. "; } }
class Класс2 : Интерфейс1, Интерфейс2
{ public string Метод() { return "Есть Интерфейс1. "; }
  public string Свойство { get { return "Есть Интерфейс2. "; } }
}
class Test
{ static string GetValue(object ob)
  { string str = "Найден интерфейс ";
  Интерфейс1 I1 = ob as Интерфейс1;
  if (I1 != null) str = I1.Метод();
  Интерфейс2 I2 = ob as Интерфейс2;
  if (I2 != null) { str += I2.Свойство; }
  return str;
}
public static void Main()
{ Класс2 объект2 = new Класс2();
  System.Console.WriteLine( GetValue(объект2));
} }
```

Укажите результат вывода на консоль после попытки запустить программу на компиляцию и выполнение:

Есть Интерфейс1. Есть Интерфейс2.

Есть Интерфейс1.

Есть Интерфейс2.

Найден интерфейс

программа не выполняется, так как обнаружена синтаксическая ошибка, связанная с тем, что интерфейс не может определять свойство

25. Исходный модуль содержит классы Class и Program:

```
class
{
  private int x, y;
  public Class(int x, int y) { this.x = x; this.y = y; }
  public static Class operator ++(Class a) { return new Class(a.x, a.y + 1);}
  public void View(){System.Console.Write("{0} {1} ",x,y);}
}
```

В классе Program определен метод Main, в теле которого находится код:

Class a = new Class(2,3),b;

b = ++a;

b.View(); a.View();

Укажите результат вывода на консоль после выполнения метода Main():

2 3 3 4

2 3 2 4

3 4 3 4

2 4 2 4

Экзамен

Вопросы к экзамену:

1. Класс System.Exception
2. Основы обработки исключительных ситуаций
3. Последствия перехвата исключений
4. Обработка исключительных ситуаций
5. Применение нескольких операторов catch
6. Перехват всех исключений
7. Вложение блоков try
8. Генерирование исключений вручную
9. Использование блока finally

10. Подробное рассмотрение класса Exception
11. Получение производных классов исключений
12. Перехват исключений производных классов
13. Применение ключевых слов checked и unchecked
14. Организация системы ввода, вывода в C# на потоках
15. Классы потоков
16. Консольный ввод, вывод
17. Класс FileStream и байтовый ввод, вывод в файл
18. Символьный ввод, вывод в файл
19. Переадресация стандартных потоков
20. Чтение и запись двоичных данных
21. Файлы с произвольным доступом
22. Применение класса MemoryStream
23. Применение классов StringReader и StringWriter
24. Класс File
25. Преобразование числовых строк в их внутреннее представление
26. Делегаты
27. Анонимные функции
28. Анонимные методы
29. Лямбда выражения
30. События
31. Применение анонимных методов и лямбда выражений с событиями
32. Пространства имен
33. Препроцессор
34. Сборки и модификатор доступа internal
35. Модификатор доступа internal
36. Динамическая идентификация типов
37. Рефлексия
38. Применение рефлексии
39. Атрибуты
40. Встроенные атрибуты

6.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

В КФУ действует балльно-рейтинговая система оценки знаний обучающихся. Суммарно по дисциплине (модулю) можно получить максимум 100 баллов за семестр, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов.

Для зачёта:

56 баллов и более - "зачтено".

55 баллов и менее - "не зачтено".

Для экзамена:

86 баллов и более - "отлично".

71-85 баллов - "хорошо".

56-70 баллов - "удовлетворительно".

55 баллов и менее - "неудовлетворительно".

Форма контроля	Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций	Этап	Количество баллов
Семестр 5			
Текущий контроль			
Лабораторные работы	В аудитории, оснащённой соответствующим оборудованием, обучающиеся проводят учебные эксперименты и тренируются в применении практико-ориентированных технологий. Оцениваются знание материала и умение применять его на практике, умения и навыки по работе с оборудованием в соответствующей предметной области.	1	20
Устный опрос	Устный опрос проводится на практических занятиях. Обучающиеся выступают с докладами, сообщениями, дополнениями, участвуют в дискуссии, отвечают на вопросы преподавателя. Оценивается уровень домашней подготовки по теме, способность системно и логично излагать материал, анализировать, формулировать собственную позицию, отвечать на дополнительные вопросы.	2	20

Форма контроля	Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций	Этап	Количество баллов
Тестирование	Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий.	3	10
Экзамен	Экзамен нацелен на комплексную проверку освоения дисциплины. Экзамен проводится в устной или письменной форме по билетам, в которых содержатся вопросы (задания) по всем темам курса. Обучающемуся даётся время на подготовку. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.		50
Семестр 6			
Текущий контроль			
Лабораторные работы	В аудитории, оснащённой соответствующим оборудованием, обучающиеся проводят учебные эксперименты и тренируются в применении практико-ориентированных технологий. Оцениваются знание материала и умение применять его на практике, умения и навыки по работе с оборудованием в соответствующей предметной области.	1	20
Устный опрос	Устный опрос проводится на практических занятиях. Обучающиеся выступают с докладами, сообщениями, дополнениями, участвуют в дискуссии, отвечают на вопросы преподавателя. Оценивается уровень домашней подготовки по теме, способность системно и логично излагать материал, анализировать, формулировать собственную позицию, отвечать на дополнительные вопросы.	2	20
Тестирование	Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий.	3	10
Экзамен	Экзамен нацелен на комплексную проверку освоения дисциплины. Экзамен проводится в устной или письменной форме по билетам, в которых содержатся вопросы (задания) по всем темам курса. Обучающемуся даётся время на подготовку. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.		50

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке им. Н.И. Лобачевского. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра (для обучающихся по ФГОС 3++ - не менее 0,25 экземпляра) каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Microsoft - <https://www.microsoft.com/ru-ru>

Программирование - <http://professorweb.ru/>

Программирование - <https://metanit.com/>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	<p>Лекция - устное изложение информации, выстроенное по строго определенной логической структуре, подчиненной задаче максимально глубоко и понятно раскрыть заданную тематику. Основное назначение лекции: помощь в освоении фундаментальных аспектов; упрощение процесса понимания научно-популярных проблем; распространение сведений о новых достижениях современной науки. Функции лекционной подачи материала: информационная; стимулирующая; воспитательная; развивающая; ориентирующая; поясняющая; убеждающая. Лекция позволяет раскрыть основные понятия и проблематику изучаемой области науки, дать учащимся представление о сути предмета, продемонстрировать взаимосвязь с другими смежными дисциплинами.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>
лабораторные работы	<p>Лабораторная работа требует от студентов не только наличия знаний, но еще и умений применять эти знания в новых ситуациях, сообразительности. Лабораторная работа активизирует познавательную деятельность студентов, выполняется на компьютере. Используется лабораторная работа для закрепления определенных навыков с программными средствами, когда кроме алгоритмических предписаний в задании студентов может получать консультации преподавателя. Подготовка к лабораторной работе включает 2 этапа: организационный; закрепление и углубление теоретических знаний. В процессе этой работы студент должен стремиться понять и запомнить основные положения материала, примеры, поясняющие его, а также разобраться в дополнительном материале.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>
самостоятельная работа	<p>В процессе самостоятельной деятельности студент должен научиться выделять познавательные задачи, выбирать способы их решения, выполнять операции контроля за правильностью решения поставленной задачи, совершенствовать навыки реализации теоретических знаний. Формирование умений и навыков самостоятельной работы студентов может протекать как на сознательной, так и на интуитивной основе. Самостоятельная работа - деятельность обучающегося направленная на самостоятельное выполнение заданий. К самостоятельной работе относятся: подготовка к практическим, лабораторным и семинарским занятиям, подготовка к устному опросу, подготовка к тестированию. Самостоятельная работа творческого характера позволяет развивать творческие способности студентов.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>
устный опрос	<p>Устный опрос как контроль знаний студентов осуществляется в виде фронтальной и индивидуальной проверки. При фронтальном опросе за короткое время проверяется состояние знаний студентов всей группы по определенному вопросу или группе вопросов. Эта форма проверки используется для: выяснения готовности группы к изучению нового материала; определения сформированности понятий; проверки домашних заданий; поэтапной или окончательной проверки учебного материала, только что разобранный на занятии; при подготовке к выполнению практических и лабораторных работ. Индивидуальный устный опрос позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления, культуру речи студентов.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>
тестирование	<p>Тест представляет собой кратковременное технически сравнительно просто составленное испытание, проводимое в равных для всех условиях. Различают следующие виды тестов. Избирательный тест состоит из системы заданий, к каждому из которых прилагаются как верные, так и неверные ответы. Из них студент выбирает тот, который считает верным для данного вопроса. При этом неверные ответы содержат такую ошибку, которую студент может допустить, имея определенные пробелы в знаниях. Закрытые тесты не содержат вариантов ответов. Студенты предлагают свой вариант ответа. Имеются тесты перекрестного выбора, в которых требуется установить соответствие между элементами множества ответов. Встречаются также тесты идентификации, в которых в качестве ответов приводятся графики, схемы, чертежи.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>

Вид работ	Методические рекомендации
экзамен	<p>Экзаменом заканчивается изучение предмета или его основного раздела. Наиболее распространённые формы экзамена: экзамен по билетам, экзамен-беседа, письменный экзамен, коллоквиум, тестовый контроль, программированный контроль. Успешно сдает экзамен тот, кто умеет периодически повторяет материал, который был прослушан на лекциях, законспектирован и закреплён на самостоятельных занятиях. Повторение предполагает обобщение, углубление, а также расширение усвоенных знаний. При подготовке к экзамену необходимо опираться на лекции, а также на источники, которые разбирались на практических, семинарских и лабораторных занятиях.</p> <p>Использование дистанционных технологий в образовательном процессе.</p>

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

Компьютер и принтер для распечатки раздаточных материалов.

Мультимедийная аудитория.

Компьютерный класс.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи;
- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 09.03.03 "Прикладная информатика" и профилю подготовки "Прикладная информатика в экономике".

Приложение 2
к рабочей программе дисциплины (модуля)
Б1.В.03 Объектно-ориентированное программирование

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: Прикладная информатика в экономике

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Основная литература:

1. Хорев П. Б. Объектно-ориентированное программирование с примерами на С# : учебное пособие / П.Б. Хорев. - Москва : ФОРУМ : ИНФРА-М, 2020. - 200 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - URL: <https://znanium.com/catalog/product/1069921> (дата обращения: 06.10.2020). - Текст : электронный.
2. Шакин В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.NET : учебное пособие / В.Н. Шакин, А.В. Загвоздкина, Г.К. Сосновиков. - Москва : ФОРУМ : ИНФРА-М, 2019. - 398 с. - (Высшее образование. Бакалавриат). - ISBN 978-5-00091-048-1. - URL: <https://znanium.com/catalog/product/1010028> (дата обращения: 06.10.2020). - Текст : электронный.
3. Корнеев В. И. Программирование графики на С++. Теория и примеры : учебное пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. - Москва : ИД 'ФОРУМ' : ИНФРА-М, 2019. - 517 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0837-2. - URL: <https://znanium.com/catalog/product/1018909> (дата обращения: 06.10.2020). - Текст : электронный.

Дополнительная литература:

1. Медведев М.А. Программирование на СИ#: учебное пособие / М.А. Медведев, А.Н. Медведев. - 2-е изд., стер. - Москва : Флинта, Изд-во Урал. ун-та, 2017. - 64 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-9765-3169-7. - URL: <http://znanium.com/catalog/product/948428> (дата обращения: 06.10.2020). - Текст : электронный.
2. Кучунова Е.В. Программирование. Процедурное программирование: учебное пособие / Е.В. Кучунова, Б.В. Олейников, О.М. Чередниченко. - Красноярск: СФУ, 2016. - 92 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-7638-3555-7 - URL: <http://znanium.com/catalog/product/978627> (дата обращения: 06.10.2020). - Текст : электронный.
3. Бунаков П. Ю. Практикум по решению задач на ЭВМ в среде Delphi : учебное пособие / П.Ю. Бунаков, А.К. Лопатин. - Москва : ФОРУМ : ИНФРА-М, 2018. - 304 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-00091-481-6. - URL: <https://znanium.com/catalog/product/939812> (дата обращения: 06.10.2020). - Текст : электронный.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.В.03 Объектно-ориентированное программирование

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 09.03.03 - Прикладная информатика

Профиль подготовки: Прикладная информатика в экономике

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7 Профессиональная или Windows XP (Volume License)

Пакет офисного программного обеспечения Microsoft Office 365 или Microsoft Office Professional plus 2010

Браузер Mozilla Firefox

Браузер Google Chrome

Adobe Reader XI или Adobe Acrobat Reader DC

Kaspersky Endpoint Security для Windows