Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ганеев Винер Валиахметович
Должность: Директор
Дата подписания: 02.11.2026 14:16:09
Уникальный программный ключ:
fceab25d7092f3bff743e8ad3f8d57fddc1f5e66

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Уфимский университет науки и технологий»

Бирский филиал

ФАКУЛЬТЕТ ФИЗИКИ И МАТЕМАТИКИ

УТВЕРЖДАЮ:
Декан
подписано ЭЦП Гайсин Ф.Р.
(подпись, инициалы, фамилия)

«26» 02 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование
(наименование дисциплины)

ОПОП ВО программа бакалавриата
09.03.03 Прикладная информатика
шифр и наименование направления подготовки (специальности)

направленность (профиль, специализация)

Прикладная информатика в информационной сфере
наименование направленности (профиля, специализации)

форма обучения
очная
(очная, очно-заочная, заочная)

Бирск 2026 г.

Рабочая программа составлена на основании учебного плана основной профессиональной образовательной программы 09.03.03 Прикладная информатика профиль Прикладная информатика в информационной сфере, одобренного ученым советом Бирского филиала Уфимского университета науки и технологий (протокол №7 от 27.02.2026 г.) и утвержденного директором Бирского филиала 27.02.2026.

Зав.кафедрой кафедры информатики и экономики *(наименование кафедры разработчика программы)*	*подписано ЭЦП*	Стовба Е.В.
Разработчик программы	*подписано ЭЦП*	Тазетдинова Ю.А.
Руководитель образовательной программы	*подписано ЭЦП*	Бигаева Л.А.

1. Цель дисциплины. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы

1.1. Цель дисциплины

Цель изучения дисциплины: формирование у студентов системы знаний в области объектно-ориентированного и функционального программирования, алгоритмов и структур данных, принципам программирования; овладение приемами разработки программного обеспечения для решения прикладных задач.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы

Таблица 1. – Результаты обучения по дисциплине

Планируемые результаты освоения основной профессиональной образовательной программы (компетенции, закрепленные за дисциплиной)		Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной
код компетенции	наименование компетенции	
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знать принципы работы современных информационных технологий технологий и программных средств, в том числе отечественного производства, и их методы использования для решения задач профессиональной деятельности
		ОПК-2.2. Уметь описывать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их для решения задач профессиональной деятельности
		ОПК-2.3. Владеть навыками использования принципов работы современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Демонстрирует знания основных языков программирования и подходы в разработке алгоритмов, современных программных сред разработки информационных систем и технологий.
		ОПК-7.2. Демонстрирует навыки разработки алгоритмов и программ, пригодных для практического применения.
		ОПК-7.3. Разрабатывает алгоритмы и программы, пригодные для практического применения.

2. Структура и трудоемкость дисциплины

Общая трудоемкость (объем) дисциплины составляет 9 зачетные единицы (з.е.), 324 академических часов.

Таблица 2 – Объем дисциплины

Виды учебной работы	Всего, часов	Количество часов в семестре
Общая трудоемкость дисциплины	324	2 семестр - 180 3 семестр - 144
Контактная работа обучающихся с преподавателем по видам учебных занятий (всего)	166	2 семестр - 84 3 семестр - 82
в том числе:		
лекции	36	2 семестр - 18 3 семестр - 18
лабораторные занятия	96	2 семестр - 48 3 семестр - 48
практические занятия	34	2 семестр - 18 3 семестр - 16
Другие виды работ в соответствии с УП:		
контрольная работа	1	2 семестр - 1 3 семестр - 1
консультации	2	2 семестр - 1 3 семестр - 1
Самостоятельная работа обучающихся (всего)	158	2 семестр - 96 3 семестр - 62
Из них:		
контроль	69.6	
ФКР:		
зачет	0	
зачет с оценкой	0	
курсовая работа (проект)	2	3 семестр - 1
экзамен	2.4	2 семестр - 1 3 семестр - 1

3 Содержание дисциплины

Таблица 3 – Содержание дисциплины

№ п/п	Раздел (тема) дисциплины	Виды деятельности							Форма текущего контроля успеваемости
		Лек, час.	Лаб, час.	П, час.	Эк, час.	КоР, час.	КуР, час.	С	
1 курс / 2 семестр									
1	Visual Studio .NET как среда визуального объектно-ориентированного программирования								
1.1	Основные понятия объектно-ориентированного программирования (ООП).	2						6	Гестирование

	Основные понятия объектно-ориентированного программирования (ООП). Синтаксические особенности объектно-ориентированных языков программирования. Технология и среды визуального программирования. Microsoft Visual Studio.NET как единая визуальная среда для создания .NET приложений.								
1.2	Разработка Windows-приложений в Visual Studio .NET. Разработка Windows- приложений в Visual Studio .NET. Стандартные элементы управления и простейшие проекты. Флажки, переключатели, контейнеры. Списки. Свойства, события и методы списков. Организация ввода-вывода данных в Windows-приложениях. Вычислительные проекты.	2	6	2				6	Лабораторная работа
2	Строки и массивы как объекты.								
2.1	Массивы. Массивы как объекты. Коллекции. Работа с коллекциями объектов. Ссылка на объекты в коллекции. Использование цикла for each ... next для работы с коллекциями. Элемент управления DataGridView. Ввод и вывод одномерных и двумерных массивов в визуальных средах. Примеры проектов.	2	6	4				8	Лабораторная работа
2.2	Строки. Ссылочные типы данных. Строки. Методы модуля Strings. Строка как объект. Свойства и методы класса String.	2	6	2				6	Тестирование, Лабораторная работа
3	Графика и анимация								
3.1	Компьютерная графика. Компьютерная графика. Методы и свойства класса Graphics. Элемент управления PictureBox. Создание графических композиций.	2	6	2				8	Лабораторная работа
3.2	Анимация.	2	6	2				6	Лабораторная

	Создание простейшей анимации. Элемент управления Timer.							работа
3.3	Построение графиков функций. Элемент управления Chart. Построение графиков функций. Рекурсия. Примеры рекурсивных методов с использованием графики.	2	6	2				4 Лабораторная работа, Тестирование
4	Текстовые файлы, диалоговые окна. Типизированные файлы.							
4.1	Работа с файлами в C#. Работа с файлами в C#. Класс File. Потоки. Класс FileStream.	2	6	2				8 Лабораторная работа
4.2	Текстовые файлы. Текстовые файлы. Классы StreamReader и StreamWriter. Примеры чтения из файла и записи в файл. Диалоговые окна открытия и сохранения файла.	2	6	2				7 Тестирование, Лабораторная работа
5	Контрольная работа					1		0
6	Экзамен				1			3
Итого по 1 курсу 2 семестру		18	48	18	1	1		9
2 курс / 3 семестр								
1	Реализация объектно-ориентированного подхода на языке C#							
1.1	Реализация объектно-ориентированного подхода на языке C# Определение, область применения и основные понятия объектно-ориентированного программирования. Описание класса. Статические члены класса, и члены экземпляра. Методы: виды, описание, вызов. Способы передачи параметров в методы. Типы параметров: значения, ссылки (ref), выходные параметры (out). Инкапсуляция. Модификаторы доступа.	10	24	8				8 Тестирование, Лабораторная работа

	Конструкторы. Ключевое слово this. Свойства, индексаторы. Наследование. Раннее и позднее связывание. Виртуальные методы. Абстрактные и бесплодные (запечатанные) классы. Класс object: основные элементы. Интерфейсы. Сравнение и клонирование объектов.							
2	Востребованные возможности языка C#, дополняющие объектно-ориентированную парадигму							
2.1	Востребованные возможности языка C#, дополняющие объектно-ориентированную парадигму Структуры. Отличия структур от классов. Обнуляемые типы. Перечисления (enum).Обработка исключительных ситуаций. Операторы throw, try-catch-finally. Класс Exception и его производные.Работа с неуправляемыми ресурсами. Операторы try-finally, using. Интерфейс IDisposable.Делегаты. Библиотечные делегаты Func и Action. Лямбда-выражения. События. Автоматический вывод типа. Анонимные типы. Расширяющие методы.Коллекции и итераторы. Интерфейсы IEnumerable, IEnumerator. Цикл foreach. Оператор yield return.Библиотечные коллекции и интерфейсы.	8	24	8				1Лабораторная работа, Тестирование
3	Контрольная работа					1		0
4	Курсовая работа						1	2
5	Экзамен				1			3
Итого по 2 курсу 3 семестру		18	48	16	1	1	1	6
Итого по дисциплине		36	96	34	2	2	1	1

Таблица 4 – Лабораторные работы

№	Наименование лабораторных работ	Объем, час.

п/п		

Таблица 5 – Практические (семинарские) занятия

№ п/п	Наименование семинарских и практических работ	Объем, час.

4. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Критериями оценивания являются баллы, которые выставляются за виды деятельности (оценочные средства) по итогам изучения модулей (разделов дисциплины), перечисленных в рейтинг-плане дисциплины. Баллы, выставляемые за конкретные виды деятельности представлены ниже.

Контрольные задания для проведения текущего контроля успеваемости

Тестовые задания

Описание тестовых заданий: тестовые задания включают тесты закрытого типа (с одним правильным ответом), тесты на установлении последовательности и на установление соответствия. Оценка за выполнение тестовых заданий выставляется на основании процента заданий, выполненных студентами в процессе прохождения промежуточного и рубежного контроля знаний
Среда разработки Microsoft Visual Studio является обязательной для разработки приложений на платформах Microsoft.NET, .Net Core. Нет Да
Среда разработки Microsoft Visual Studio позволяет разрабатывать приложения только на языках C# и Visual Basic Нет Да Microsoft Visual Studio: Контейнер для одного или нескольких связанных проектов вместе с информацией о сборке, параметрами окна Visual Studio и любыми прочими файлами, которые не относятся к какому-либо конкретному проекту называется решением сборкой проектом файлом конфигурации группой Microsoft Visual Studio: Контейнер для всех файлов исходного кода, значков, изображений, файлов данных и прочих элементов, которые будут скомпилированы в исполняемую программу, библиотеку или веб-сайт называется решением сборкой проектом файлом конфигурации группой
Git: Каталог файловой системы, в котором находятся: файлы конфигурации, файлы журналов операций, индекс расположения файлов и хранилище, содержащее сами контролируемые файлы называется репозиторием репозиторий Git: Виды репозиториев Локальный Удалённый Глобальный Серверный Git: ... — это параллельная версия репозитория. Она включена в этот репозиторий, но не влияет на главную версию, тем самым позволяя свободно работать в параллельной. Когда вы внесли нужные изменения, то вы можете объединить их с главной версией.
Git: ... — фиксация изменений или запись изменений в репозиторий. Коммит происходит на локальной машине.
Git: ... — получение последних изменений с удалённого сервера репозитория.
Git: ... — отправка всех неотправленных коммитов на удалённый сервер репозитория.
Дан фрагмент программы class Account { public string name; public int level; public static int Counter = 0; } ... Account p = new Account();Допустимое обращение к level: p.level Account.level level.p Account[level] p[level]
Дан фрагмент программы class Account { public string name; public int level; public static int counter = 0; } ... Account p = new Account();Допустимое обращение к counter: Account.counter p.counter counter.p Account[counter] p[counter]
Статические методы Не могут быть виртуальными Могут быть вызваны только для объекта класса, в котором описаны
Чтобы получить ссылку на экземпляр, для которого вызван метод, в теле этого метода используется ключевое слово ...
Модификатор доступа, который обозначает член класса или структуры, доступный всюду за пределами класса ...

Верные утверждения о конструкторах объектов Конструктор - это метод Конструктор вызывается только при создании объекта В заголовке конструктора указывается тип результата void В одном классе может быть только один конструктор, остальные наследуются
Элемент класса, объединяющий в себе одно из двух или оба специальных метода get и set, и использующий их при доступе к нему как к переменной свойство поле конструктор оператор индексатор
При переопределении невиртуального метода необходимо использовать ключевое слово class TheBest { public ??? MyResultType TheMethod(...) { ... } }
Дан фрагмент программы class Control { } class WinControl : Control { } class CustomControl : WinControl { } ... Control t = new CustomControl(); var m = (CustomControl)t;Выражение **(CustomControl)t** в последней строке вернёт Ссылку на объект класса CustomControl Ссылку на объект класса Control null Исключительную ситуацию

Методические материалы, определяющие процедуру оценивания выполнения тестовых заданий

Описание методики оценивания выполнения тестовых заданий: оценка за выполнение тестовых заданий ставится на основании подсчета процента правильно выполненных тестовых заданий.
Критерии оценки (в баллах)*:*
- **9-10** баллов выставляется студенту, если процент правильно выполненных тестовых заданий составляет 81 – 100 %;
- **7-8** баллов выставляется студенту, если процент правильно выполненных тестовых заданий составляет 61 – 80 %;
- **4-6** баллов выставляется студенту, если процент правильно выполненных тестовых заданий составляет 41 – 60 %;
- **до 4** баллов выставляется студенту, если процент правильно выполненных тестовых заданий составляет 40 %;

Контрольная работа

Разработать иерархию классов "замки дверные"
1. Разработать класс OrdinaryLock (замок-обычный) Свойство *Locked* (Запертый замок или нет): только для чтения, начальное значение - false. Метод GenerateKey (генерация ключа). Ключём будет целое число. Ключ должен храниться в экземпляре класса и быть доступным только для для него.В начальный момент времени Ключ = 0, что означает, что он не определён. Первый вызов метода GenerateKey генерирует, запоминает (в поле объекта) и возвращает (return) случайное целое число не равное 0. Первым вызовом метода мы моделируем получение ключа. Последующие вызовы метода ничего не делают, и возвращают 0. Таким образом мы моделируем невозможность получить ключ от замка посторонним. Метод *Open* (открыть). Методу передаётся некий ключ, метод проверяет, совпадает ли данный ключ с ключём замка. Если да, то замок открывается, в противном случае - не открывается. Добавить тестовый код:Создание замка. Генерация ключа, вывод его на консоль. Открыть замок неверным ключём. Вывод на экран неверного ключа и состояния (Locked) замка. Открыть замок правильным ключём. Вывод на экран состояния замка.
2. Добавить к замку сигнализацию. Описать свойство - делегат Alarm типа AlarmHandler. Тип делегата AlarmHandler описать самостоятельно. Этот делегат должен вызываться при попытке открыть (в Open) замок неверным ключём.Модификация тестового кода:После создания замка задать его делегат Alarm с помощью лямбда-функции, которая просто выводит на экран сообщение "Тревога! Попытка взлома!". Больше ничего менять не надо, так как уже имеющийся код должен вызвать этот делегат, и результаты будут видимы.3. Мы решили, что одного, обычного замка нам не достаточно, и нам нужен производный от OrdinaryLock класс MagicLock (магический замок). Магический замок кроме ключа требует произнесение заклинания. Добавить к этому классу метод CastSpell (наложить заклинание). Метод применяется к замку однократно. В замке запоминается текст заклинания. Перед открытием (Open) такого замка необходимо произнести заклинание (SaySpell). Сделать метод Open полиморфным (изменения будут и в базовом классе).Тестовый код:Создать набор из 4-х замков. 2 - обычных и два магических. Инициировать их. Отобразить

начальное состояние. Затем в цикле предлагать пользователю открыть замок. Пользователь указывает номер замка (от 1 до 4, 0 - выход из цикла), ключ и заклинание. Производится попытка открыть указанный замок и отображается его состояние.

Методические материалы, определяющие процедуру оценивания контрольной работы

Описание методики оценивания: при оценке выполнения студентом контрольной работы максимальное внимание следует уделять следующим аспектам: насколько полно в теоретическом вопросе раскрыто содержание материала, четко и правильно даны определения, раскрыто содержание понятий; верно использованы научные термины; демонстрируются высокий уровень умения оперировать научными категориями, анализировать информацию, владение навыками практической деятельности; кейс-задание решено на высоком уровне, содержит аргументацию и пояснения.

Критерии оценки (в баллах):

- **9-10** баллов выставляется студенту, если в теоретическом вопросе полно раскрыто содержание материала; четко и правильно даны определения, раскрыто содержание понятий; верно использованы научные термины; демонстрируются высокий уровень умения оперировать научными категориями, анализировать информацию, владение навыками практической деятельности; кейс-задание решено на высоком уровне, содержит пояснения; тестовые задания решены свыше, чем на 80%; уровень знаний, умений, владений – высокий;
- **7-8** баллов выставляется студенту, если В теоретическом вопросе раскрыто основное содержание материала; в основном правильно даны определения понятий и использованы научные термины; ответ самостоятельный; определения понятий неполные, допущены незначительные нарушения в последовательности изложения; небольшие недостатки при использовании научных терминов; кейс-задание решено верно, но решение не доведено до завершающего этапа; тесты решены на 60-80%. Уровень знаний, умений, владений – средний;
- **5-6** баллов выставляется студенту, если в теоретическом вопросе усвоено основное, но непоследовательно; определения понятий недостаточно четкие; не использованы в качестве доказательства выводы и обобщения из наблюдений, практических занятий; уровень умения оперировать научными категориями, анализировать информацию, владения навыками практической деятельности невысокий, наблюдаются пробелы и неточности; в решение кейс-задания верно выполнены некоторые этапы; тесты решены на 40-60%; уровень знаний, умений, владений – удовлетворительный;
- **менее 5** баллов выставляется студенту, если в теоретическом вопросе не изложено основное содержание учебного материала, изложение фрагментарное, не последовательное; определения понятий не четкие; не использованы в качестве доказательства выводы и обобщения из наблюдений, уровень умения оперировать научными категориями, анализировать информацию, владения навыками практической деятельности очень низкий; тесты решены менее, чем на 40 %; уровень знаний, умений, владений – недостаточный.

Решение задач

Решение задач способствует формированию умений и навыков относящихся к конкретной сфере деятельности

1. Описать класс «боксёр» : имя, фамилия, вес, число боёв, число побед. Сформировать список боксёров (имя и фамилия вводятся с клавиатуры, остальное либо вводится с клавиатуры либо генерируется случайным образом). Найти игроков, с процентом побед выше 60, и весом менее 70 кг.
2. Описать класс Угол, который хранит информацию в радианной мере, но позволяет работать с собой и в радианной и в градусной мере. Для этого описать методы GetGradus, SetGradus.
3. Описать класс Угол, который хранит информацию в радианной мере, но позволяет работать с собой и в радианной и в градусной мере. Для этого описать свойство Gradus.
4. Разработать метод, который применяет заданную функцию к каждому элементу коллекции, и возвращает коллекцию из результатов.

Методические материалы, определяющие процедуру оценивания решения задач

Описание методики оценивания выполнения решения задачи: уделяется внимание выбранному алгоритму, рациональному способу решения, правильному применению формул, получению верного ответа.
Критерии оценки
3 балла выставляется студенту, если: составлен правильная компьютерная программа или алгоритм решения задачи, учтены все ограничения и условия задачи, задача решена рациональным способом.
2 балла выставляется студенту, если выполнено хотябы одно из условий: составленая компьютерная программа или алгоритм решения задачи не содержит существенных ошибок, не учтены вспомогательные ограничения (условия) задачи, задача решена нерациональным способом.
1 балл выставляется студенту, если составленая компьютерная программа или алгоритм решения задачи содержит не более 2-х существенные ошибки, не учтены важные ограничения (условия) задачи.
0 баллов выставляется студенту, если если решения нет, либо оно, по большей части, неверно.

Лабораторная работа

Лабораторная работа 1. Классы, объекты, инкапсуляция
1. Разработать класс - стек вещественных чисел, фиксированного размера. Стек это контейнер, работающий по принципу "последний вошел, первый вышел" (last in, first out — LIFO). Реализацию стека выполнить с помощью массива. Размер стека задаётся в конструкторе класса. Свойства: Capacity - Максимальный размер стека, Count - количество элементов в стеке. Методы: Push - записать число на вершину стека, Pop - извлечь число с вершины стека, Peek - вернуть число с вершины стека, без его извлечения, Clear - очистить стек.
При выполнении недопустимой операции со стеком сгенерировать исключительную ситуацию стандартного класса InvalidOperationException. Осуществить инкапсуляцию, то есть обдуманно выделить интефейсную часть класса (public) и его реализацию (private). Продемонстрировать работоспособность класса простейшим способом.
2. Изучить класс Stack из стандартной библиотеки платформы .NET. Продемонстрировать работу с этим классом. В каком пространстве имён находится этот класс? В какой сборке? В чём отличия данного класса от разработанного вами в задаче 1?
3. Разработать класс инкапсулирующий стековый калькулятор.
Стековый калькулятор работает следующим образом. Числа, которыми он оперирует, хранятся в стеке. То есть пользователь калькулятора может помещать числа в стек и извлекать их оттуда. Для выполнения математической операции калькулятор извлекает из стека необходимое количество операндов, выполняет вычисления, и результат записывает в стек.
Осуществить инкапсуляцию.
4. Разработать консольную программку - стековый калькулятор. Воспользоваться классом из задачи 3. Пользователь вводит в одной строчке либо число, либо одну из команд +, -, *, /, sqrt, clear, после чего нажимает "ввод". Калькулятор выполняет вычисления, и отображает количество элементов в стеке, и 2 или меньшее количество чисел у вершины с тека.
При возникновении во время вычислений исключительных ситуаций, программа должна выдавать сообщение "недопустимая операция", и продолжать работу.

Методические материалы, определяющие процедуру оценивания выполнения лабораторных работ

Процедура сдачи лабораторной работы состоит из
1. Теоретической части, которая состоит из 2-х вопросов (низкой или средней сложности), ответы на которые оцениваются на 0-2 балла. Переход к сдаче практической части возможен только если за теоретическую часть полученно не менее 2 баллов.

2. Практической части, которая содержит несколько задач (обычно 3-5), за каждую из которых можно получить от 2 до 5 баллов.
Суммарно по всем частям можно получить до 20 баллов. Эта оценка, на равне с оценками за другие лабораторыне работы, используется при расчёте рейтинга по дисциплине.
Параллельно с этим, для удобства её можно перевести в 4-х балльную оценку, аналогично тому, как это делается для всей дисциплины (неудовлетворительно: < 9 баллов, удовлетворительно: 9 - 13, хорошо: 14 - 15, отлично 16 - 20).

Курсовая работа

Описание курсовой работы: курсовая работа, как правило, включает теоретическую часть — изложение позиций и подходов, сложившихся в науке по данному вопросу, и аналитическую (практическую часть) — содержащую анализ проблемы на примере конкретной ситуации (на примере предприятия, экологической проблемы или иного объекта). Курсовая работа в обязательном порядке содержит оглавление, введение, в котором формулируются цель и задачи, теоретический раздел, практический раздел, иногда проектную часть, в которой студент отражает проект решения рассматриваемой проблемы, заключение, список литературы, и приложения по необходимости. Объем курсовой работы может варьироваться.
Примерные темы курсовых работ:
1. Разработка графического редактора на C#
2. Разработка тренажера для развития памяти.
3. Разработка тренажера устного счета.
4. Разработка логической игры "Лабиринт" на C#.
5. Разработка логической игры "Пазлы" на C#.
6. Разработка логической игры "Мемори" на C#.
7. Разработка логической игры "Пятнашки" на C#.
8. Разработка логической игры "Судоку" на C#.
9. Разработка логической игры "Крестики-нолики" на C#.
10. Разработка обучающей программы "Тренажер системы счисления" на C#.

Методические материалы, определяющие процедуру оценивания выполнения курсовых работ

Критерии оценки (в баллах*):*
- **5 баллов** выставляется студенту, если:
слайды презентации по курсовоой работы оформлены в соответствии стребованиями к презентациям, полностью раскрывают суть работы и полученные результаты.
доклад студентом выполнен в хорошем темпе, без значительных запинок. Студент ответил на дополнительные вопросы.
оформление текста курсовой выполнено в соответствии с требованиями к оформлению курсовых работ.
разработанное приложение обладает необходимым функционалом (в зависимости от темы).
- **4 балла** выставляется студенту, если:
слайды презентации по курсовоой работы оформлены в соответствии стребованиями к презентациям, но одержат незначиельные ошибки. Слайды раскрывают основную суть работы и полученные результаты.
доклад студентом выполнен в хорошем темпе, без значительных запинок. Студент ответил на большинство дополнительных вопросов.
оформление текста курсовой выполнено в соответствии с требованиями к оформлению курсовых работ.
разработанное приложени обладает необходимым функционалом (в зависимости от темы.
- **3 балла** выставляется студенту, если:
слайды презентации по курсовоой работы оформлены не качественно. Слайды не полностью раскрывают основную суть работы и полученные результаты.
доклад студентом выполнен с запинками. Студент не ответил на дополнительные вопросы.

оформление текста курсовой выполнено в соответствии с требованиями к оформлению курсовых работ.
разработанное приложение обладает недостаточным функционалом (в зависимости от темы).
- **2 балла** выставляется студенту, если:
слайды презентации по курсовоой работы оформлены не качественно или их нет.
доклад студентом выполнен с запинками. Студент не ответил на дополнительные вопросы.
оформление текста курсовой выполнено не в соответствии с требованиями к оформлению курсовых работ. Логически содержание не соответсвует тематике.
разработанное приложение обладает недостаточным функционалом (в зависимости от темы) или не доработано.

Экзаменационные билеты

Экзамен (зачет) является оценочным средством для всех этапов освоения компетенций. Структура экзаменационного билета: в билете указывается кафедра в рамках нагрузки которой реализуется данная дисциплина, форма обучения, направление и профиль подготовки, дата утверждения; билет может включать в себя теоретический(ие) вопрос(ы) и практическое задание (кейс-задание).

Примерные вопросы к экзамену, 1 курс / 2 семестр

1. Основные понятия объектно-ориентированного программирования (ООП) .
2. Технология и среды визуального программирования.
3. Microsoft Visual Studio .NET как единая визуальная среда для создания .NET приложений.
4. Разработка Windows-приложений в Visual Studio .NET. Модель объектно-ориентированного событийно - управляемого программирования (основные понятия).
5. Форма и основные элементы управления. Разработка проектов.
6. Организация ввода-вывода в визуальных средах. Примеры .
7. Элементы управления, организующие выбор пользователя (флажки и переключатели). Примеры.
8. Классы и объекты. Динамическое создание объектов.
9. Элемент управления ListBox. Строки в C#.
10. Работа с одномерными массивами в C#. Организация ввода-вывода массивов в windows–приложениях.
11. Работа с двумерными массивами в C#. Элемент управления DataGridView.
12. Элемент управления Chart. Построение графиков функций.
13. Класс Graphics. Создание графических композиций.
14. Создание простейшей анимации. Элемент управления Timer.
15. Процедуры и функции – методы класса. Синтаксис описания методов в C#.
16. Формальные и фактические параметры. Передача параметров по значению и ссылке. Примеры .
17. Рекурсия. Примеры рекурсивных методов.
18. Работа с файлами в C#. Класс File. Потоки. Класс FileStream.
19. Классы StreamReader и StreamWriter. Примеры чтения из файла и записи в файл.
20. Диалоговые окна открытия и сохранения файла.

Примерные вопросы к экзамену, 2 курс / 3 семестр

1. Объектно-ориентированное программирование (ООП): определение, область применения. Основные понятия ООП - инкапсуляция, наследование, полиморфизм.
2. Классы. Виды элементов класса. Модификаторы доступа. Статические поля, константы.
3. Сокрытие данных. Инкапсуляция.
4. Типы-значения и ссылочные типы. Автоматическое управление памятью.
5. Методы: виды, описание, вызов. Ключевое слово this. Виды параметров.

6. Конструкторы. Вызов из конструктора другого конструктора. Вызов из конструктора конструктора базового класса.
7. Свойства. Автоматически реализуемые свойства. Индексаторы.
8. Наследование. Раннее и позднее связывание. Виртуальные методы. Полиморфизм.
9. Полиморфизм. Операторы as и is, явное приведение типа. Класс Object и его элементы.
10. Абстрактные и запечатанные классы (модификаторы abstract и sealed).
11. Интерфейсы. Реализация интерфейса. Сравнение и клонирование объектов.
12. Исключительные ситуации и их обработка. Операторы throw, try-catch, try-finally.
13. Работа с неуправляемыми ресурсами. Интерфейс IDisposable. Финализаторы. Оператор try-finally. Оператор using.
14. Делегаты. Лямбда-функции. Библиотечные типы делегатов Func и Action. События.
15. Структуры. Обнуляемые типы (типы, допускающие значение null). Перечисления.
16. Коллекции и итераторы. Интерфейсы IEnumerable, IEnumerator. Цикл foreach. Оператор yield return.
17. Инициализаторы объектов. Ключевое слово var. Анонимные типы. Расширяющие методы.
18. Библиотечные коллекции и интерфейсы: массивы переменной длины, неизменяемые массивы, словари, очереди, стеки.

Образец экзаменационного билета

<table>
<tr><td colspan="2">МИНОБРНАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»
БИРСКИЙ ФИЛИАЛ УУНиТ
Кафедра информатики и экономики</td></tr>
<tr><td>Дисциплина: Программирование
очная форма обучения
1 курс 2 семестр</td><td>Курсовые экзамены 20__-20__ г.
Направление 09.03.03 Прикладная информатика
Профиль: Прикладная информатика в информационной сфере</td></tr>
<tr><td colspan="2">Экзаменационный билет № 1
1. Технология и среды визуального программирования.
2. Решить задачу.</td></tr>
<tr><td>Дата утверждения: __.__._____</td><td>Заведующий кафедрой
_____</td></tr>
</table>

Методические материалы, определяющие процедуру оценивания ответа на экзамене

Критериями оценивания являются баллы, которые выставляются за виды деятельности (оценочные средства) по итогам изучения модулей (разделов дисциплины), перечисленных в рейтинг-плане дисциплины: текущий контроль – максимум 40 баллов; рубежный контроль – максимум 30 баллов, поощрительные баллы – максимум 10.

При оценке ответа на экзамене максимальное внимание должно уделяться тому, насколько полно раскрыто содержание материала, четко и правильно даны определения, раскрыто содержание понятий, верно ли использованы научные термины, насколько ответ самостоятельный, использованы ли ранее приобретенные знания, раскрыты ли раскрыты причинно-следственные связи, насколько высокий уровень умения оперирования научными категориями, анализа информации, владения навыками практической деятельности.

Критерии оценки (в баллах):

- 25-30 баллов выставляется студенту, если студент дал полные, развернутые ответы на все теоретические вопросы билета, продемонстрировал знание функциональных возможностей,

терминологии, основных элементов, умение применять теоретические знания при выполнении практических заданий. Студент без затруднений ответил на все дополнительные вопросы. Практическая часть работы выполнена полностью без неточностей и ошибок;
- **17-24 баллов** выставляется студенту, если студент раскрыл в основном теоретические вопросы, однако допущены неточности в определении основных понятий. При ответе на дополнительные вопросы допущены небольшие неточности. При выполнении практической части работы допущены несущественные ошибки;
- **10-16** баллов выставляется студенту, если при ответе на теоретические вопросы студентом допущено несколько существенных ошибок в толковании основных понятий. Логика и полнота ответа страдают заметными изъянами. Заметны пробелы в знании основных методов. Теоретические вопросы в целом изложены достаточно, но с пропусками материала. Имеются принципиальные ошибки в логике построения ответа на вопрос. Студент не решил задачу или при решении допущены грубые ошибки;
- **1-10 баллов** выставляется студенту, если ответ на теоретические вопросы свидетельствует о непонимании и крайне неполном знании основных понятий и методов. Обнаруживается отсутствие навыков применения теоретических знаний при выполнении практических заданий. Студент не смог ответить ни на один дополнительный вопрос.

Перевод оценки из 100-балльной в четырехбалльную производится следующим образом:
- отлично – от 80 до 110 баллов (включая 10 поощрительных баллов);
- хорошо – от 60 до 79 баллов;
- удовлетворительно – от 45 до 59 баллов;
- неудовлетворительно – менее 45 баллов.

Задания для проведения промежуточной аттестации обучающихся

Промежуточная аттестация по дисциплине проводится в форме:

3 семестр - курсовая работа, 2,3 семестр - экзамен.

Проверяемыми на промежуточной аттестации элементами содержания являются темы дисциплины.

Для проверки *знаний* используются вопросы и задания в различных формах.

Умения, навыки (или опыт деятельности) и компетенции проверяются с помощью компетентностно-ориентированных задач (ситуационных, производственных или кейсового характера) и различного вида конструкторов.

Все задачи являются многоходовыми. Некоторые задачи, проверяющие уровень сформированности компетенций, являются многовариантными. Часть умений, навыков и компетенций прямо не отражена в формулировках задач, но они могут быть проявлены обучающимися при их решении.

В каждый вариант включаются задания по каждому проверяемому элементу содержания во всех перечисленных выше формах и разного уровня сложности. Такой формат позволяет объективно определить качество освоения обучающимися основных элементов содержания дисциплины и уровень сформированности компетенций.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине при использовании модульно-рейтинговой системы

Критериями оценивания при модульно-рейтинговой системе являются баллы, которые выставляются преподавателем за виды деятельности (оценочные средства) по итогам изучения модулей (разделов дисциплины), перечисленных в рейтинг-плане дисциплины:

для экзамена: текущий контроль – максимум 40 баллов; рубежный контроль – максимум 30 баллов, поощрительные баллы – максимум 10.

Шкалы оценивания:

для экзамена:

от 45 до 59 баллов – «удовлетворительно»;

от 60 до 79 баллов – «хорошо»;

от 80 баллов – «отлично».

Рейтинг-план дисциплины

Таблица перевода баллов текущего контроля в баллы рейтинга

	5	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
0	0	0	0	0	0	0	0	0	0	0
1	5	3	2	2	1	1	1	1	1	1
2		5	4	3	2	2	2	2	2	1
3			5	4	3	3	3	2	2	2
4				5	4	4	3	3	3	2
5					5	5	4	4	3	3
6						5	5	4	4	3
7							5	5	4	4
8								5	5	4
9									5	5
10										5

Рейтинг-план дисциплины представлен в Приложении 1.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1. Основная учебная литература

1. Павловская , Т. А. C#. Программирование на языке высокого уровня : Учеб. для вузов / Т. А. Павловская .— СПб. : Питер, 2009 .— 432 с. : ил. — (Учебник для вузов) .— ISBN 978-5-91180-174-8
2. Тюкачев, Н.А. C#. Основы программирования : учеб. пособие / Н.А. Тюкачев, В.Г. Хлебостроев. — Электрон. дан. — Санкт-Петербург : Лань, 2018. — 272 с. https://e.lanbook.com/book/104962
3. Биллиг В.А. Основы программирования на C# 3.0: ядро языка / В.А. Биллиг, 2-е изд., испр. М.: Национальный открытый университет “ИНТУИТ”, 2016 . -411 с. URL:- https://biblioclub.ru/index.php?page=book_view_red&book_id=428947

5.2. Дополнительная учебная литература

1. Волкова, Т.И. Программирование в среде PASCAL ABC: учеб. пособие / Т.И. Волкова; Башкирский государственный университет, Бирский филиал. — Бирск: БФ БашГУ, 2013. <URL:https://elib.bashedu.ru/dl/read/VolkovaTI.Program.V Srede.Pascal ABC.UchPos.pdf>.
2. Волкова, Т.И. Лабораторный практикум по визуальному программированию [Электронный ресурс]: учеб. пособие / Т.И. Волкова, А.Р. Набиуллин; Башкирский государственный университет, Бирский филиал. — Бирск: БФ БашГУ, 2014. — Электрон. версия печ. публикации. — <URL:https://elib.bashedu.ru/dl/read/Volkova_Nabiullin_Laboratorny praktikum po vizualnomu programmirovaniyu_Uch.pos_Birsk__2014.pdf>

3. Тазетдинов, Б.И. Лабораторный практикум по базовому курсу программирования на языке C #: [Электронный ресурс]: учебное пособие / Башкирский государственный университет, Бирский филиал; авт.- сост. Б.И. Тазетдинов. — Бирск: РИЦ БашГУ, 2017. <URL:https://elib.bashedu.ru/dl/local/Tazetdinov_avt-sost_Laboratornyj praktikum_Birsk_2017.pdf> .
4. Суханов М.В. Основы Microsoft.NET Framework и языка программирования C#: учеб. пособие [Электронный ресурс]/ М.В. Суханов, И.В. Бачурин, И.С. Майров; Сев. (Арктич.) федер. ун-т им. М.В. Ломоносова.- Архангельск: ИД САФУ, 2014.-97 с. URL:- https://biblioclub.ru/index.php?page=book_view_red&book_id=312313

5.3. Другие учебно-методические материалы

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

Перечень ресурсов информационно-телекоммуникационной сети «Интернет» и программного обеспечения, необходимых для освоения дисциплины

1. Научная электронная библиотека eLIBRARY.RU [Электронный ресурс]. – Режим доступа: https://elibrary.ru/.
2. Электронная библиотечная система «Лань» [Электронный ресурс]. – Режим доступа: https://e.lanbook.com/.
3. Университетская библиотека онлайн biblioclub.ru [Электронный ресурс]. – Режим доступа: http://biblioclub.ru/.
4. Электронная библиотека УУНиТ [Электронный ресурс]. – Режим доступа: https://elib.bashedu.ru/.
5. Российская государственная библиотека [Электронный ресурс]. – Режим доступа: https://www.rsl.ru/.
6. Национальная электронная библиотека [Электронный ресурс]. – Режим доступа: https://xn--90ax2c.xn--p1ai/viewers/.
7. Национальная платформа открытого образования npoed.ru [Электронный ресурс]. – Режим доступа: http://npoed.ru/.
8. Электронное образование Республики Башкортостан [Электронный ресурс]. – Режим доступа: https://edu.bashkortostan.ru/.
9. Информационно-правовой портал Гарант.ру [Электронный ресурс]. – Режим доступа: http://www.garant.ru/.

Программное обеспечение

1. Система дистанционного обучения Moodle - Бесплатная лицензия http://www.gnu.org/licenses/gpl.html
2. Браузер Яндекс - Бесплатная лицензия https://yandex.ru/legal/browser_agreement/index.html
3. Visual Studio Community - Бесплатная лицензия https://visualstudio.microsoft.com/ru/free-developer-offers/
4. Операционная система Astra Linux - Договор 0301400003023000002 от 14 марта 2023 бессрочно
5. LIBREOFFICE - Договор №0301400003023000002 от 14.03.2023 (бессрочный)

7. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Аудитория 311(ФМ)	Лекционная, Семинарская, Для курсового проектирования, Для консультаций, Для контроля и аттестации, Для лабораторных занятий, Для практических занятий	Доска, компьютер, мебель, проектор, экран. Программное обеспечение 1. Браузер Яндекс 2. Visual Studio Community 3. Операционная система Astra Linux 4. LIBREOFFICE
Аудитория 311 а(ФМ)	Для хранения оборудования	Компьютер, учебно-методическая литература. Программное обеспечение 1. Операционная система Astra Linux 2. LIBREOFFICE
Аудитория 313(ФМ)	Лекционная, Семинарская, Для консультаций, Для контроля и аттестации, Для лабораторных занятий, Для практических занятий	Доска, компьютер, мебель, проектор, экран. Программное обеспечение 1. Система дистанционного обучения Moodle 2. Операционная система Astra Linux 3. Visual Studio Community 4. LIBREOFFICE
Аудитория 420(ФМ)	Для самостоятельной работы	Компьютер, мебель. Программное обеспечение 1. Visual Studio Community 2. LIBREOFFICE 3. Операционная система Astra Linux
Читальный зал(ФМ)	Для самостоятельной работы	Компьютер, ксерокс, мебель, принтер, учебно-методические материалы. Программное обеспечение 1. LIBREOFFICE 2. Операционная система Astra Linux 3. Браузер Яндекс