Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ганеев Винер Валиахметович
Должность: Директор
Дата подписания: 02.11.2026 14:28:19
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

Объектно-ориентированное программирование
*(наименование дисциплины)*

ОПОП ВО программа бакалавриата
01.03.02 Прикладная математика и информатика
*шифр и наименование направления подготовки (специальности)*

направленность (профиль, специализация)

Математическое моделирование и управление процессами и системами
*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 01.03.02 Прикладная математика и информатика профиль Математическое моделирование и управление процессами и системами, 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 обучающихся базовых знаний, умений и навыков в области программирования, современных средств объектно-ориенированного событийного программирования; овладение приемами разработки нового программного и информационного обеспечения для решения актуальных и перспективных профессиональных 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 | |
| ПК-2 | Способен проектировать программные средства в профессиональной деятельности | ПК-2.1. Демонстрирует знания методов проектирования программных средств |
| | | ПК-2.2. Применяет методы проектирования программных средств |
| | | ПК-2.3. Проектирует программные средства в профессиональной деятельности |
| ПК-3 | Способен проводить научно-исследовательские и расчетно-модельные разработки при исследовании самостоятельных тем | ПК-3.1. Анализирует литературные источники по сформулированным темам |
| | | ПК-3.2. Применяет методы проведения научно-исследовательских и расчетно-модельных разработок |
| | | ПК-3.3. Проводит научно-исследовательские и расчетно-модельные разработки при исследовании самостоятельных тем |

## 2. Структура и трудоемкость дисциплины

Общая трудоемкость (объем) дисциплины составляет 8 зачетные единицы (з.е.), 288 академических часов.

Таблица 2 – Объем дисциплины

| Виды учебной работы | Всего, часов | Количество часов в семестре |
|---|---|---|
| Общая трудоемкость дисциплины | 288 | 3 семестр - 180 |

| Виды учебной работы | Всего, часов | Количество часов в семестре |
|---|---|---|
| | | 4 семестр - 108 |
| Контактная работа обучающихся с преподавателем по видам учебных занятий (всего) | 124 | 3 семестр - 88<br>4 семестр - 36 |
| в том числе: | | |
| лекции | 50 | 3 семестр - 34<br>4 семестр - 16 |
| лабораторные занятия | 62 | 3 семестр - 42<br>4 семестр - 20 |
| практические занятия | 12 | 3 семестр - 12 |
| Другие виды работ в соответствии с УП: | | |
| контрольная работа | 0 | |
| консультации | 1 | 4 семестр - 1 |
| Самостоятельная работа обучающихся (всего) | 164 | 3 семестр - 92<br>4 семестр - 72 |
| Из них: | | |
| контроль | 34.8 | |
| ФКР: | | |
| *зачет* | 0 | |
| *зачет с оценкой* | 0.2 | 3 семестр - 1 |
| *курсовая работа (проект)* | 0 | |
| *экзамен* | 1.2 | 4 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. | ДЗ, час. | Эк, час. | СРС, час. | |
| 2 курс / 3 семестр | | | | | | | | |
| 1 | Среда программирования visual studio.net. | | | | | | | |
| 1.1 | Среда программирования visual studio.net. Классы, объекты, конструкторы, модификаторы доступа и инкапсуляция.<br><br>Среда программирования visual studio.net в контексте объектно-ориентированного программирования. Элементы управления. Классы, объекты, методы, модификаторы доступа, конструкторы. Перегрузка методов, классов, конструкторов, инкапсуляция. | 8 | 8 | 10 | | | 20 | Лабораторная работа, Семинар |
| 1.2 | Среда .NET. Работа с WindowsForms проектами. | 4 | 4 | 2 | | | 24 | Индивидуальный опрос, Лабораторная |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Работа с WindowsForms проектами. | | | | | | | работа, Семинар |
| 2 | Основы объектно-ориентированного программирования. | | | | | | | |
| 2.1 | Индексаторы, свойства, наследование, интерфейсы, абстрактные классы.<br><br>Константы. Индексаторы и свойства, многомерные индексаторы. Инкапсуляция. Перегрузка индексаторов. Ключевое слово static. Статический конструктор. Классы. Наследование. Защищенный доступ и исключение наследования.Конструкторы и наследование. Ключевое слово base. Многоуровневая иерархия. Ссылки на базовый класс и объекты производных классов. Ключевое слово virtual. Виртуальные методы, свойства и индексаторы. Абстрактные классы. Ключевое слово abstract. Интерфейс. Интерфейсные ссылки. Ключевое слово is и as. | 20 | 26 | | | | 26 | Лабораторная работа |
| 2.2 | Обработка исключительных ситуаций. Работа с потоками и файловой системой.<br><br>Обработка исключительных ситуаций. | 2 | 4 | | | | 21.8 | Лабораторная работа, Групповой опрос, Индивидуальный опрос |
| 3 | Дифференцированный зачет | | | | 1 | | 0.2 | |
| Итого по 2 курсу 3 семестру | | 34 | 42 | 12 | 1 | | 92 | |
| 2 курс / 4 семестр | | | | | | | | |
| 1 | Основы объектно-ориентированного программирования. | | | | | | | |
| 1.1 | Обработка исключительных ситуаций. Работа с потоками и файловой системой.<br><br>Генерирование исключений вручную. Повторное генерирование исключение. Работа с потоками и файловой системой. | 4 | 6 | | | | 8 | Лабораторная работа |
| 1.2 | Делегаты. | 8 | 6 | | | | 10 | Лабораторная работа |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Делегаты. Лямбда-выражения. Применение методов экземпляра в качестве делегатов. Групповая адресация. Анонимные методы. | | | | | | | |
| 2 | События и обобщения. | | | | | | | |
| 2.1 | События.<br><br>События, объявление, механизм подписки. Обработка событий в среде .NET Framework. | 2 | 2 | | | | 10 | Лабораторная работа |
| 2.2 | Обобщения.<br><br>Обобщения. Коллекции. Стандартные коллекции. Необобщенные и обобщенные коллекции. | 2 | 6 | | | | 8 | Групповой опрос, Лабораторная работа |
| 3 | Экзамен | | | | | 1 | 36 | |
| Итого по 2 курсу 4 семестру | | 16 | 20 | | | 1 | 72 | |
| Итого по дисциплине | | 50 | 62 | 12 | 1 | 1 | 164 | |

Таблица 4 – Лабораторные работы

| № п/п | Наименование лабораторных работ | Объем, час. |
|---|---|---|
| 2 курс / 3 семестр | | |
| 1. | Классы, объекты, конструкторы, модификаторы доступа и инкапсуляция. | 8 |
| 2. | Работа с WindowsForms проектами. | 4 |
| 3. | Индексаторы, свойства, наследование, интерфейсы, абстрактные классы. | 26 |
| 4. | Обработка исключительных ситуаций. Работа с потоками и файловой системой. | 4 |
| 2 курс / 4 семестр | | |
| 1. | Обработка исключительных ситуаций. Работа с потоками и файловой системой. | 6 |
| 2. | Делегаты. | 6 |
| 3. | События. | 2 |
| 4. | Обобщения. | 6 |

Таблица 5 – Практические (семинарские) занятия

| № | Наименование семинарских и практических работ | Объем, час. |
|---|---|---|

| | | |
|---|---|---|
| п/п | | |
| 2 курс / 3 семестр | | |
| 1. | Классы, объекты, конструкторы, модификаторы доступа и инкапсуляция. | 10 |
| 2. | Работа с WindowsForms проектами. | 2 |

## 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**

**Решение задач**

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

Перечень примерных задач определяющие процедуру оценивания решения задач на рубежном контроле в первом семестре.

**Перечень примерных задач определяющие процедуру оценивания решения задач на рубежном контроле в первом семестре.**

Модуль 1. Среда программирования visual studio.net.

**Задачи на классы в среде программирования visual studio.net.**

1. Написать класс со встроенным методом, с двумя перегрузками, выводящий значение функции f($x$), если x типа double, то вычислить значение = sin($x$)+cos($x$) от введенного аргумента $x$, если x типа int, то вычислить значение = $x^2$-3$x$ от введенного аргумента $x$. Создать 5 экземпляров класса, вызвать созданный метод несколько раз, чтобы результат был представлен в виде таблицы в диапазоне -5 до 5 с шагом 1.
2. Написать класс со встроенным методом, с двумя перегрузками, выводящий значение функции f($x$), если x типа double, то вычислить значение = 1/sin($x$)+7*$x$ от введенного аргумента $x$, если x типа int, то вычислить значение = 0,5$x^2$+2$x$-1 от введенного аргумента $x$. Создать 5 экземпляров класса, вызвать созданный метод 5 раз, чтобы результат был представлен в виде таблицы в диапазоне -5 до 5 с шагом 1.
3. Написать класс Textovka() с несколькими перегрузками конструктора. Первая перегрузка Textovka(a,b,c,d,e) переменные строкового типа. Вводит пять слов, которые хранятся в памяти компьютера. Вторая перегрузка конструктора пустая, выдает на экран консоли текст случайным образом на основе первой перегрузки, где слова a,b,c,d,e расставляются в любом порядке, если значениям этих слов ничего не присвоено, то этим параметрам задать значения “один”, “два”, “три”, “четыре”, “пять”. (Общее количество слов выводимых на экран консоли должно превышать 1000.)
4. Написать класс Textovka() с несколькими перегрузками конструктора. Первая перегрузка Textovka(a,b,c,d,e) переменные строкового типа. Вводит пять слов, которые хранятся в памяти компьютера. Вторая перегрузка конструктора пустая, выдает на экран консоли текст в алфавитном порядке на основе первой перегрузки, где слова a,b,c,d,e, если значениям этих слов ничего не присвоено, то этим параметрам задать значения “один”, “два”, “три”, “четыре”, “пять”.
5. Реализовать задачу три построением цепочки вызовов конструкторов с помощью ключевого слова this. В задаче допускается применение авторских конструкторов.
6. Реализовать задачу четыре построением цепочки вызовов конструкторов с помощью ключевого слова this. В задаче допускается применение авторских конструкторов.
7. Реализовать класс выдающий в консоли спадающий вниз строку из любой позиции консоли. Текст спадает строго вертикально. В качестве текста формируется случайные

символы. Градация цвета букв синий, зеленый, желтый, красный. При реализации использовать цепочку конструкторов, методы. (На подобии из фильма матрица).

8. Реализовать класс выдающий на экран консоли плавающую строку, заранее введенное пользователем. Каждый раз достигнув края экрана строка поворачивается слева на право, меняет цвет текста и скорость перемещения по экрану консоли.

Модуль 2. Основы объектно-ориентированного программирования.

**Задачи на наследование**

1. Создать базовый класс Auto со строковыми полями model и marka. На его основе создать класс Price с полями отвечающими за компоновку модели и их стоимость. Производному классу предоставить полный доступ к полям базового класса. В производном классе реализовать метод выводящий текущие характеристики автомобиля, включая параметры описанные в полях базового класса.
    1. В методе Main, реализовать создание четырех объектов производного класса.
    2. Подсчитать стоимость четырех автомобилей которые описаны в объектах пункта а.
2. Создать базовый класс Computer со строковыми полями отвечающие за наличие основных компонентов компьютера system_block, monitor, mouse, klava их название и характеристики. На его основе создать класс Price с полями отвечающими за стоимость компьютера. Производному классу предоставить полный доступ к полям базового класса. В производном классе реализовать метод выводящий текущие характеристики автомобиля, включая параметры описанные в полях базового класса.
    1. В методе Main, реализовать создание четырех объектов производного класса.
    2. Подсчитать стоимость четырех компьютеров которые описаны в объектах пункта а.
3. Создать базовый класс Auto со строковыми полями model и marka. На его основе создать три производных класса Dvigatel, Korobka, Obch_harakteristiki, соответственно отвечающие за характеристики двигателя (топливо, мощность, макс. скорость, время разгона до 100 км/ч, расход топлива литров на 100 км.), коробки передач (вид коробки, количество ступеней) и общие характеристики (количество мест, снаряженная масса, объем багажника, диаметр разворота). Все поля отвечающие за характеристики машины в производных классах описать в базовом классе в режиме зачищенного доступа, а доступ к этим полям реализовать через публичные свойства. Так же в базовом классе реализовать метод выводящий все характеристики автомобиля. А ввод этих характеристик по отдельным категориям, реализовать в методах производного класса. В методе Main, реализовать вывод полной информации со всеми характеристиками о трех автомобилях.
4. Создать базовый класс Computer со строковыми полями которые отвечают за наличие основных компонентов компьютера. На основе базового класса создать три производных класса System_block (мощность блока питания, материнская плата, процессор, оперативная память, вид и объем винчестера), Monitor (марка, диагональ, вид матрицы), Peref (мышь, клавиатура, колонки, наушники, веб микрофон с камерой, принтер, сканер), соответственно отвечающие за характеристики системного блока, монитора и периферийных устройств. Все поля отвечающие за характеристики компьютера в производных классах описать в базовом классе в режиме зачищенного доступа, а доступ к этим полям реализовать через публичные свойства. Так же в базовом классе реализовать публичный метод выводящий все характеристики компьютера. А ввод этих характеристик по отдельным категориям, реализовать в методах производного класса. В методе Main, реализовать вывод полной информации со всеми характеристиками о трех компьютерах.
5. Дан класс:

```
class Cat
{
string tip_cat;
string poroda_cat;
string klichka;
public Cat() {}
```

```
public string Tip_cat
{
get { return tip_cat; }
set { tip_cat = value; }
}
public string Poroda_cat
{
get { return poroda_cat; }
set { poroda_cat = value; }
}
public string Klichka
{
get { return klichka; }
set { klichka = value; }
}
public void vivod_1()
{
Console.Write("Тип кошки: " + tip_cat + " \n" + " Порода кошки: " + poroda_cat + " \n" + " кличка кошки: " + klichka + " \n");
}
}
```

На основе базового класса Cat, создать производный класс Fis_harac_cat (физические характеристики кота). Внутри производного класса задать приватные поля отвечающие за рост, длину, вес и возраст кота. Ввод этих данных в производном классе реализовать через конструктор, а вывод этой информации через публичный метод vivod_2(). В методе Main, реализовать вывод полной информации со всеми характеристиками о трех кошках.

6. Дан класс Cat из условия 5 задания.

На основе базового класса Cat, создать производный класс Psih_harac_cat (психические характеристики кота). Внутри производного класса задать приватное поле отвечающее за количество команд которые понимает кошка. Ввод этого поля в производном классе реализовать через публичный метод Vvod(), а вывод этой информации через публичный метод vivod_2(). В методе Main, реализовать вывод полной информации со всеми характеристиками о трех кошках.

7. Дан класс:

```
class Chelovek
{
private string fam;
private string im;
private string ot;
private string data_rojd;
public string Fam
{
get { return fam; }
set { fam = value; }
}
public string Im
{
get { return im;}
set { im = value; }
}
public string Ot
{
get { return ot; }
```

```
set { ot = value; }
}
public string Data_rojd
{
get { return data_rojd; }
set { data_rojd = value; }
}
}
```
На основе базового класса Chelovek, создать производный класс Client. Внутри производного класса задать приватные поля отвечающее за номер паспорта и скрытое поле отвечающее за дату рождения. Ввод этих полей в производном классе реализовать через публичные свойства, а вывод всей информации о клиенте (фамилия, имя, отчество, год рождения, номер паспорта) реализовать через публичный метод vivod(). В методе Main, реализовать вывод полной информации со всеми характеристиками о трех клиентах.

8. Дан класс Chelovek из условия 7 задания.

На основе базового класса Chelovek, создать производный класс Student. Внутри производного класса задать приватные поля отвечающее за номер курса и скрытые поля от базового класса (fam, im, ot) отвечающее за фамилию, имя и отчество. Ввод этих полей в производном классе реализовать через публичные свойства, а вывод всей информации о студенте (фамилия, имя, отчество, номер курса) реализовать через публичный метод vivod(). В методе Main, реализовать вывод полной информации со всеми характеристиками об n студентах с помощью цикла.

Список примерных задач на экзамене во втором семестре

**События**

**Задание:** Приведен фрагмент кода с классами и событием.

```
class Plateg
{
public void Message(int a)
{
Console.WriteLine($"Произведен платеж {a} рублей");
}
}
class Spicivanie
{
public void Message(int a)
{
Console.WriteLine($"Со счета списано {a} рублей");
}
}
class Koshelek
{
int sum;
public delegate void Oplata(int a);
public event Oplata Delo;
public Koshelek(int sum)
{
this.sum = sum;
}
public void Vivod(int a)
{
if (a &lt; sum)
{
```

```
Delo(a);
}
else
{
Console.WriteLine("недостаточно средств!");
}
}
}
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Введите количество средств в вашем электронном кошелке");
int v = Convert.ToInt32(Console.ReadLine());
Koshelek K1 = new Koshelek(v);
Plateg P1 = new Plateg();
Spicivanie S1 = new Spicivanie();
Console.Write("Введите сумму платежа: ");
int b = Convert.ToInt32(Console.ReadLine());
K1.Delo += P1.Message;
K1.Delo += S1.Message;
K1.Vivod(b);
Console.ReadLine();
}
}
```

1. Добавьте в класс Koshelek метод Ostatok - остаток на счете после снятия. Подпишите его к событию и выведите на экран консоли как дополнительный параметр, после снятия средств.
2. Преобразуйте исходный код класса Koshelek, с возможностью добавления средств в первоначальную сумму. Создайте метод Dobavit – добавление средств в кошелек. Подпишите его к событию и выведите на экран консоли как дополнительный параметр, предлагающий пополнить счет после снятия средств.
3. Создайте класс Password, реализовать в нем делегат, событие, метод вызывающий событие при совпадении пароля с именем владельца. В методе void Main реализовать работу событие на предмет проверки пароля минимум пять раз.
4. Создайте класс Auto, реализовать в нем делегат, событие, метод вызывающий событие при совпадении номера машины с именем владельца. В методе void Main реализовать работу событие на предмет проверки номера машины с именем владельца минимум пять раз.
5. Создайте windows проект, с обработчиком одиночного нажатия мышью по форме, в результате которой форма убегает от указателя мыши в случайном направлении. Т.е. новое местоположение формы устанавливается, случайным образом. На десятой попытке форма закрывается.
6. Создайте windows проект, с обработчиком события, когда указатель мыши входит в видимую часть формы в результате которой форма убегает от указателя мыши в случайном направлении. Т.е. новое местоположение формы, устанавливается случайным образом. На десятой попытке форма закрывается.

Методические материалы, определяющие процедуру оценивания выполнения решения задач на экзамене

Решение задач на экзамене осуществляется с целью проверки уровня знаний, умений, навыков (владений) студента по решению практической задачи.
Ход выполнения задачи требует осмысления, анализа, а затем решения. Решение задачи должно быть аргументированным, содержать пояснения.

Методические материалы, определяющие процедуру оценивания решения задач.

Решение задач является эффективной формой рубежного контроля. Осуществляется с целью проверки уровня знаний, умений, навыков (владений) студента по решению практической задачи. Ход выполнения задачи требует осмысления, анализа, а затем решения. Решение задачи должно быть аргументированным, содержать пояснения.

**Вопросы для семинаров**

Семинарское занятие 1.
Введение в объектно-ориентированное программирование.
Семинарское занятие 2.
Классы и объекты. Методы и параметры в методах.
Семинарское занятие 3.
Классы и объекты. Модификаторы доступа. Инкапсуляция.
Семинарское занятие 4.
Классы и объекты. Перегруженные методы. Рекурсия.
Семинарское занятие 5
Классы и объекты. Конструкторы. Перегрузка конструкторов (this).
Семинарское занятие 6.
Среда .NET. Работа с WindowsForms проектами.

Методические материалы, определяющие процедуру оценивания ответа на семинаре

При оценивании ответа на семинаре следует уделять внимание тому, насколько полно раскрыто содержание материала, четко и правильно даны определения, раскрыто ли содержание понятий, верно ли использованы научные термины; использованы ли при ответе ранее приобретенные знания; раскрыты ли в процессе причинно-следственные связи; демонстрируются высокий уровень умения оперировать знениями, анализировать информацию.
**Критерии оценки (в баллах)**:
- **5** баллов выставляется студенту, если полно раскрыто содержание материала; четко и правильно даны определения, раскрыто содержание понятий; верно использованы научные термины; ответ самостоятельный, использованы ранее приобретенные знания; раскрыты причинно-следственные связи; демонстрируются высокий уровень умения оперировать научными категориями, анализировать информацию;
- **4** балла выставляется студенту, если 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; демонстрируются хороший уровень умения оперировать научными категориями, анализировать информацию;
- **3** балла выставляется студенту, если недостаточно раскрыто основное содержание учебного материала, не последовательно; определения понятий недостаточно четкие; допущены ошибки и неточности в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию низкий;
- **0-2** балла выставляется студенту, если не раскрыто содержание учебного материала, изложено фрагментарно, определения понятий не четкие; допущены значительные ошибки в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию очень низкий.

**Индивидуальный опрос**

**Перечень примерных вопросов для индивидуального опроса.**
Модуль 1. Среда программирования visual studio.net.

1. Среда программирования visual studio.net в контексте объектно-ориентированного программирования.
2. Работа с WindowsForms проектами.
3. Структура обозревателя решений.
4. Средства разработки приложений .net (Обозреватель решений).
5. Элементы управления.
6. Базовые понятия объектно- ориентированного программирования (ООП). Инкапсуляция. Наследование. Полиморфизм.
7. Классы, объекты. (Создание, инициализация).
8. Отношения между классами и объектами.
9. Жизненный цикл объекта. Конструкторы и деструкторы.
10. Основные принципы объектного подхода. Инкапсуляция. (Модификаторы доступа)
11. Типы значения и ссылочные типы. (Примерами с присваиванием).
12. Классы и методы. (Виды методов, работа с ними через экземпляр класса).

Модуль 2. Основы объектно-ориентированного программирования.

1. Константы и поля чтения.
2. Свойства (get, set) методы.
3. Класс. Конструкторы. Перегрузка конструкторов.
4. Класс. Конструкторы. Ключевое слово this. Сцепление конструкторов.
5. Метод Main. Возвращение значений из метода и передача аргументов методу Main().
6. Применение модификатора static и его ограничения. Статические конструкторы.
7. Индексаторы. Многомерные индексаторы.
8. Основные принципы ООП. Наследование. На примере классов, базовый и производный класс.
9. Основные принципы ООП. Наследование. Защищенный доступ и исключение наследование. (Protected, Sealed).
10. Основные принципы ООП. Наследование. Конструкторы и наследование.
11. Основные принципы ООП. Наследование. Наследование и сокрытие имен.(new, base).
12. Основные принципы ООП. Наследование. Ссылки на базовый класс и объекты производных классов.
13. Основные принципы. Полиморфизм.
14. Виртуальные методы свойства и индексаторы (virtual и override).
15. Абстрактные классы и методы.
16. Структуры.
17. Перечисления.
18. Исключения. Перехват исключений. Класс Exception.
19. Конфигурирование исключений. Обработка многочисленных исключений. Перехват всех исключений.
20. Перехват исключений вручную (throw и finally).
21. Ключевые слова cheked и unchecked, при перехватах исключений.
22. Интерфейс.
23. Интерфейсные ссылки.
24. Интерфейс. Ключевые слова as и is.
25. Интерфейсные свойства и индексаторы.
26. Наследование интерфейсов.
27. Явная реализация интерфейсов.

Методические материалы, определяющие процедуру оценивания ответа на индивидуальном опросе

При выставлении оценки преподаватель учитывает правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, умение связывать теоретические положения с практикой, в том числе и с будущей профессиональной деятельностью.

При оценивании ответа на индивидуальном опросе следует уделять внимание тому, насколько полно раскрыто содержание материала, четко и правильно даны определения, раскрыто ли содержание понятий, верно ли использованы научные термины; использованы ли при ответе ранее приобретенные знания; раскрыты ли в процессе причинно-следственные связи; демонстрируются высокий уровень умения оперировать знаниями, анализировать информацию.

**Критерии оценки (в баллах)**:

- 9-10 баллов выставляется студенту, если полно раскрыто содержание материала; четко и правильно даны определения, раскрыто содержание понятий; верно использованы научные термины; ответ самостоятельный, использованы ранее приобретенные знания; раскрыты причинно-следственные связи; демонстрируются высокий уровень умения оперировать научными категориями, анализировать информацию;
- 7-8 баллов выставляется студенту, если 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; демонстрируются хороший уровень умения оперировать научными категориями, анализировать информацию;
- 4-6 баллов выставляется студенту, если недостаточно раскрыто основное содержание учебного материала, не последовательно; определения понятий недостаточно четкие; допущены ошибки и неточности в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию низкий;
- ниже 4 баллов выставляется студенту, если не раскрыто содержание учебного материала, изложено фрагментарно, определения понятий не четкие; допущены значительные ошибки в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию очень низкий.

**Групповой опрос**

1 семестр

**Перечень примерных вопросов для группового устного опроса (первый семестр).**

1. Среда программирования visual studio.net в контексте объектно-ориентированного программирования.
2. Работа с WindowsForms проектами.
3. Структура обозревателя решений.
4. Средства разработки приложений .net (Обозреватель решений).
5. Элементы управления.
6. Базовые понятия объектно- ориентированного программирования (ООП). Инкапсуляция. Наследование. Полиморфизм.
7. Классы, объекты. (Создание, инициализация).
8. Отношения между классами и объектами.
9. Жизненный цикл объекта. Конструкторы и деструкторы.
10. Основные принципы объектного подхода. Инкапсуляция. (Модификаторы доступа)
11. Типы значения и ссылочные типы. (Примерами с присваиванием).
12. Классы и методы. (Виды методов, работа с ними через экземпляр класса).
13. Константы и поля чтения.
14. Свойства (get, set) методы.
15. Класс. Конструкторы. Перегрузка конструкторов.
16. Класс. Конструкторы. Ключевое слово this. Сцепление конструкторов.
17. Метод Main. Возвращение значений из метода и передача аргументов методу Main().
18. Применение модификатора static и его ограничения. Статические конструкторы.
19. Индексаторы. Многомерные индексаторы.
20. Основные принципы ООП. Наследование. На примере классов, базовый и производный класс.

21. Основные принципы ООП. Наследование. Защищенный доступ и исключение наследование. (Protected, Sealed).
22. Основные принципы ООП. Наследование. Конструкторы и наследование.
23. Основные принципы ООП. Наследование. Наследование и сокрытие имен.(new, base).
24. Основные принципы ООП. Наследование. Ссылки на базовый класс и объекты производных классов.
25. Основные принципы. Полиморфизм.
26. Виртуальные методы свойства и индексаторы (virtual и override).
27. Абстрактные классы и методы.
28. Структуры.
29. Перечисления.
30. Исключения. Перехват исключений. Класс Exception.
31. Конфигурирование исключений. Обработка многочисленных исключений. Перехват всех исключений.
32. Перехват исключений вручную (throw и finally).
33. Ключевые слова cheked и unchecked, при перехватах исключений.
34. Интерфейс.
35. Интерфейсные ссылки.
36. Интерфейс. Ключевые слова as и is.
37. Интерфейсные свойства и индексаторы.
38. Наследование интерфейсов.
39. Явная реализация интерфейсов.

2 семестр

**Перечень примерных вопросов для группового устного опроса (второй семестр).**

1. Потоки в C#. (Символьные, байтовые, встроенные потоки).
2. Потоки в C#. Класс Stream.
3. Потоки в C#. Класс FileStream и байтовый ввод-вывод в файл.
4. Потоки в C#. Символьный ввод-вывод в файл. StreamWriter(Stream поток)
5. Работа с файлами в C#.
6. Потоки в C#. Чтение и запись текстовых файлов. StreamReader и StreamWriter.
7. Делегаты. Объявление. Обращение.
8. Групповое преобразование делегируемых методов.
9. Делегаты. Применение методов экземпляров в качестве делегатов.
10. Делегаты. Групповая адресация.
11. Делегаты. Понятия ковариантности и контравариантности.
12. Анонимные методы.
13. Анонимные методы. Передача аргумента анонимному методу и возврат значения из него.
14. Анонимные методы. Применение внешних переменных в анонимных методах.
15. Одиночные лямбда – выражения.
16. Блочные лямбда – выражения.
17. События. Объявление, механизм подписки.
18. События. Применение статических методов и методов экземпляра в качестве обработчика.
19. Обработка событий в среде .NET Framework.
20. Обобщения. Обобщенные классы.
21. Обобщения. Ограниченные типы.
22. Ограниченные классы.
23. Ограниченные интерфейсы и конструкторы.
24. Ограничения ссылочного типа и типа значений.
25. Иерархия обобщенного класса.
26. Обобщенные методы и типы.
27. Обобщенные структуры.

28. Обобщенные делегаты.
29. Коллекции.
30. Стандартные коллекции.
31. Необобщенные коллекции.
32. Обобщенные коллекции.

Методические материалы, определяющие процедуру оценивания группового устного опроса

При выставлении оценки преподаватель учитывает правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, умение связывать теоретические положения с практикой, в том числе и с будущей профессиональной деятельностью.
При оценивании ответа на групповом опросе следует уделять внимание тому, насколько полно раскрыто содержание материала, четко и правильно даны определения, раскрыто ли содержание понятий, верно ли использованы научные термины; использованы ли при ответе ранее приобретенные знания; раскрыты ли в процессе причинно-следственные связи; демонстрируются высокий уровень умения оперировать знаниями, анализировать информацию.
**Критерии оценки (в баллах)**:
- 9-10 баллов выставляется студенту, если полно раскрыто содержание материала; четко и правильно даны определения, раскрыто содержание понятий; верно использованы научные термины; ответ самостоятельный, использованы ранее приобретенные знания; раскрыты причинно-следственные связи; демонстрируются высокий уровень умения оперировать научными категориями, анализировать информацию;
- 7-8 баллов выставляется студенту, если 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; демонстрируются хороший уровень умения оперировать научными категориями, анализировать информацию;
- 4-6 баллов выставляется студенту, если недостаточно раскрыто основное содержание учебного материала, не последовательно; определения понятий недостаточно четкие; допущены ошибки и неточности в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию низкий;
- ниже 4 баллов выставляется студенту, если не раскрыто содержание учебного материала, изложено фрагментарно, определения понятий не четкие; допущены значительные ошибки в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию очень низкий.

## Лабораторная работа

**Лабораторная работа №1**
**Модификаторы доступа**
**Задания для самостоятельного выполнения**

1. Написать метод, выводящий значение функции f($x$) = sin($x$)+cos($x$) от введенного аргумента $x$. Вызвать метод несколько раз, чтобы результат был представлен в виде таблицы в диапазоне -5 до 5 с шагом 0,5.
2. Написать метод, выводящий значение функции f($x$) = 1/sin($x$)+7*$x$ от введенного аргумента $x$. Вызвать метод несколько раз, чтобы результат был представлен в виде таблицы в диапазоне -5 до 5 с шагом 0,5.
3. Написать метод fun($x$,$i$), выводящий значение функции от введенного аргумента $x$ и параметра $i$. Вызвать метод несколько раз, чтобы результат был представлен в виде таблицы в диапазоне -5 до 5 с шагом 0,5. Так при значении $i$ =1 вычисляется f($x$) = sin(2*$x$), при значении $i$ =2 вычисляется f($x$) = sin($x$), при значении $i$ =3 вычисляется f($x$) = 2*sin($x$)
4. Написать метод fun($x$,$i$), выводящий значение функции от введенного аргумента $x$ и параметра $i$. Вызвать метод несколько раз, чтобы результат был представлен в виде таблицы

в диапазоне -5 до 5 с шагом 0,5. Так при значении *i* =1 вычисляется f(*x*) = cos(2**x*), при значении *i* =2 вычисляется f(*x*) = cos(*x*), при значении *i* =3 вычисляется f(*x*) = 2*cos(*x*).

5. Написать метод void Kvadr(***a***), который передает в метод ссылку на аргумент ***a***, а затем возводит это число в квадрат. Вызвать метод более 5 раз.
6. Написать метод void Koren(***a***), который передает в метод ссылку на аргумент ***a***, а затем , берет квадратный корень из этого числа. Вызвать метод более 5 раз.
7. Написать метод void Mest(***a***,***b***), который передает в метод ссылку на аргументы ***a***, ***b***, а затем меняет местами значения этих аргументов не ссылочного типа. Вызвать метод более 5 раз.
8. Написать метод void Min_Maxt(***a***,***b***,***c***), который передает в метод ссылку на аргументы ***a***, ***b***, ***c*** а затем меняет местами значения максимального и минимального из этих аргументов не ссылочного типа. Вызвать метод более 5 раз.
9. Написать метод void max(***a***,***b***,***c***) ***a***, ***b*** входные аргументы, с выходной. Параметру ***c*** присваивается максимальное значение из двух введенных параметров. Вызвать созданный метод в методе Main 5 раз.
10. Написать метод void min(***a***,***b***,***c***) ***a***, ***b*** входные аргументы, с выходной. Параметру ***c*** присваивается минимальное значение из двух введенных параметров. Вызвать созданный метод в методе Main 5 раз.
11. Написать метод void Sfera(***r***,***s***,***v***) ***r*** входной аргумент – радиус сферы, ***s***, ***v*** выходные – площадь и объем сферы соответственно. Вызвать созданный метод в методе Main 5 раз.
12. Написать метод void Sfera(***r***,***s***,***v***) ***r*** входной аргумент – ребро октаэдра, ***s***, ***v*** выходные – площадь и объем октаэдра соответственно. Вызвать созданный метод в методе Main 5 раз.
13. Создать метод void array_s1(d), который находит сумму нечетных элементов массива, d это массив переменной длины. Вызвать созданный метод в методе Main 5 раз.
14. Создать метод void array_s2(d), который находит сумму элементов массива d с четными номерами, d это массив переменной длины. Вызвать созданный метод в методе Main 5 раз.
15. Создать метод void array_sort1(d), который сортирует элементы массива d по возрастанию, d это массив переменной длины. Применить сортировку методом выбора. Вызвать созданный метод в методе Main 5 раз.
16. Создать метод void array_sort2(d), который сортирует элементы массива d по убыванию, d это массив переменной длины. Применить пузырьковую сортировку. Вызвать созданный метод в методе Main 5 раз.
17. Создать метод void array_sort_1(d), который сортирует элементы массива d по возрастанию, d это массив переменной длины (Сортировка производится только над элементами с нечетными номерами). Применить сортировку методом выбора. Вызвать созданный метод в методе Main 5 раз.
18. Создать метод void array_sort_2(d), который сортирует элементы массива d по убыванию, d это массив переменной длины. (Сортировка производится только над элементами с нечетными номерами). Применить пузырьковую сортировку. Вызвать созданный метод в методе Main 5 раз.

Методические материалы, определяющие процедуру оценивания лабораторных работ

При оценивании лабораторной работы следует уделять внимание тому, насколько качественно выполнены задания и студентом демонстрируются владение освоенной тематикой; демонстрируются высокий уровень умения оперировать знаниями, анализировать информацию.
**Критерии оценки (в баллах)**:
- 1 балла выставляется студенту, если полно и качественно выполнены лабораторные задания; 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; демонстрируются хороший уровень умения оперировать научными категориями, анализировать информацию;

- **0** балла выставляется студенту, если качественно выполнены фрагментарно лабораторные задания; с некоторыми недочетами если не раскрыто содержание учебного материала, изложено фрагментарно, определения понятий не четкие; допущены значительные ошибки в использовании научной терминологии определения понятий; уровень умения оперировать научными категориями, анализировать информацию очень низкий.

**Дифференцированный зачет**

Примерные вопросы к дифзачету, 2 курс / 3 семестр

1. Базовые понятия объектно- ориентированного программирования (ООП).
2. Классы, объекты. (Создание, инициализация).
3. Отношения между классами и объектами.
4. Жизненный цикл объекта. Конструкторы и деструкторы.
5. Основные принципы объектного подхода. Инкапсуляция. (Модификаторы доступа)
6. Типы значения и ссылочные типы. (Примерами с присваиванием).
7. Классы и методы. (Виды методов, работа с ними через экземпляр класса).
8. Константы и поля чтения.
9. Свойства (get, set) методы.
10. Класс. Конструкторы. Перегрузка конструкторов.
11. Класс. Конструкторы. Ключевое слово this. Сцепление конструкторов.
12. Метод Main. Возвращение значений из метода и передача аргументов методу Main().
13. Применение модификатора static и его ограничения. Статические конструкторы.
14. Индексаторы. Многомерные индексаторы.
15. Основные принципы ООП. Наследование. На примере классов, базовый и производный класс.
16. Основные принципы ООП. Наследование. Защищенный доступ и исключение наследование. (Protected, Sealed).
17. Основные принципы ООП. Наследование. Конструкторы и наследование.
18. Основные принципы ООП. Наследование. Наследование и сокрытие имен.(new, base).
19. Основные принципы ООП. Наследование. Ссылки на базовый класс и объекты производных классов.
20. Основные принципы. Полиморфизм.
21. Виртуальные методы свойства и индексаторы (virtual и override).
22. Абстрактные классы и методы.
23. Структуры.
24. Перечисления.
25. Исключения. Перехват исключений. Класс Exception.
26. Конфигурирование исключений. Обработка многочисленных исключений. Перехват всех исключений.
27. Перехват исключений вручную (throw и finally).
28. Ключевые слова cheked и unchecked, при перехватах исключений.
29. Интерфейс.
30. Интерфейсные ссылки.
31. Интерфейс. Ключевые слова as и is.
32. Интерфейсные свойства и индексаторы.
33. Наследование интерфейсов.
34. Явная реализация интерфейсов.

Методические материалы, определяющие процедуру оценивания дифференцированного зачета

Оценка на дифференцированном зачете выставляется по баллу набранному по рейтинг-план дисциплины.

Перевод оценки из 100-балльной шкалы 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.

**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, 2 курс / 4 семестр

1. Потоки в C#. (Символьные, байтовые, встроенные потоки).
2. Потоки в C#. Класс Stream.
3. Потоки в C#. Класс FileStream и байтовый ввод-вывод в файл.
4. Потоки в C#. Символьный ввод-вывод в файл. StreamWriter(Stream поток)
5. Работа с файлами в C#.
6. Потоки в C#. Чтение и запись текстовых файлов. StreamReader и StreamWriter.
7. Делегаты. Объявление. Обращение.
8. Групповое преобразование делегируемых методов.
9. Делегаты. Применение методов экземпляров в качестве делегатов.
10. Делегаты. Групповая адресация.
11. Делегаты. Понятия ковариантности и контравариантности.
12. Анонимные методы.
13. Анонимные методы. Передача аргумента анонимному методу и возврат значения из него.
14. Анонимные методы. Применение внешних переменных в анонимных методах.
15. Одиночные лямбда – выражения.
16. Блочные лямбда – выражения.
17. События. Объявление, механизм подписки.
18. События. Применение статических методов и методов экземпляра в качестве обработчика.
19. Обработка событий в среде .NET Framework.
20. Обобщения. Обобщенные классы.
21. Обобщения. Ограниченные типы.
22. Ограниченные классы.
23. Ограниченные интерфейсы и конструкторы.
24. Ограничения ссылочного типа и типа значений.
25. Иерархия обобщенного класса.
26. Обобщенные методы и типы.
27. Обобщенные структуры.
28. Обобщенные делегаты.
29. Коллекции.
30. Стандартные коллекции.
31. Необобщенные коллекции.
32. Обобщенные коллекции.

Образец экзаменационного билета

| |
|---|
| МИНОБРНАУКИ РФ<br>ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ<br>ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |

| «УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»<br>БИРСКИЙ ФИЛИАЛ УУНиТ<br>Кафедра информатики и экономики | |
|---|---|
| Дисциплина: Объектно-ориентированное программирование<br>очная форма обучения<br>2 курс 4 семестр | Курсовые экзамены 20__-20__ г.<br>Направление 01.03.02 Прикладная математика и информатика<br>Профиль: Математическое моделирование и управление процессами и системами |
| **Экзаменационный билет № 1**<br>1. Потоки в C#. Класс Stream.<br>2. События. Применение статических методов и методов экземпляра в качестве обработчика.<br>3. Задача. | |
| Дата утверждения: __.__._____ | Заведующий кафедрой<br>_____ |

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 - дифзачет, 4 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. 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
2. C#. Программирование на языке высокого уровня : Учеб. для вузов / Т. А. Павловская .— СПб. : Питер, 2009 .— 432 с

### 5.2. Дополнительная учебная литература

1. 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> .
2. 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

**Перечень рекомендуемых ресурсов информационно-телекоммуникационной сети «Интернет», находящихся в свободном доступе**

1. Электронный ресурс поддержки microsoft: [сайт]. https://msdn.microsoft.com

## 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. Браузер Google Chrome - Бесплатная лицензия https://www.google.com/intl/ru_ALL/chrome/privacy/eula_text.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. LIBREOFFICE - Договор №0301400003023000002 от 14.03.2023 (бессрочный)
5. Операционная система Astra Linux - Договор 0301400003023000002 от 14 марта 2023 бессрочно
6. Система дистанционного обучения Moodle - Бесплатная лицензия http://www.gnu.org/licenses/gpl.html

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

| Наименование специализированных аудиторий, кабинетов, лабораторий | Вид занятий | Наименование оборудования, программного обеспечения |
|---|---|---|
| Аудитория 231(ФМ) | Лекционная, Семинарская, Для консультаций, Для контроля и аттестации | Доска, коммутатор, компьютер, мебель.<br>Программное обеспечение<br>1. Система дистанционного обучения Moodle<br>2. LIBREOFFICE<br>3. Браузер Google Chrome<br>4. Браузер Яндекс<br>5. Операционная система Astra Linux |
| Аудитория 301 Читальный зал (электронный каталог)(ФМ) | Для самостоятельной работы | Компьютер, мебель, принтер, сканер hp scanyet g2410.<br>Программное обеспечение<br>1. Браузер Google Chrome<br>2. LIBREOFFICE |

| | | |
|---|---|---|
| Аудитория 311(ФМ) | Лекционная, Семинарская, Для консультаций, Для контроля и аттестации | Доска, компьютер, мебель, проектор, экран.<br>Программное обеспечение<br>1. Браузер Google Chrome<br>2. Браузер Яндекс<br>3. Visual Studio Community<br>4. Система дистанционного обучения Moodle<br>5. Операционная система Astra Linux<br>6. LIBREOFFICE |
| Аудитория 311 а(ФМ) | Для хранения оборудования | Компьютер, мебель.<br>Программное обеспечение<br>1. LIBREOFFICE<br>2. Браузер Google Chrome |
| Аудитория 313(ФМ) | Лекционная, Семинарская, Для консультаций, Для контроля и аттестации | Доска, компьютер, мебель, экран.<br>Программное обеспечение<br>1. LIBREOFFICE<br>2. Система дистанционного обучения Moodle<br>3. Операционная система Astra Linux<br>4. Браузер Яндекс |
| Аудитория 313 а(ФМ) | Для хранения оборудования | Видеоплеер lgl315, проектор. |
| Аудитория 411(ФМ) | Лекционная, Семинарская, Для консультаций, Для контроля и аттестации | Компьютер, мебель, проектор, экран.<br>Программное обеспечение<br>1. Операционная система Astra Linux<br>2. LIBREOFFICE<br>3. Система дистанционного обучения Moodle<br>4. Браузер Яндекс<br>5. Браузер Google Chrome |
| Аудитория 411 а(ФМ) | Для хранения оборудования | Мебель.<br>Программное обеспечение<br>1. LIBREOFFICE<br>2. Операционная система Astra Linux<br>3. Браузер Google Chrome |
| Аудитория 420(ФМ) | Для самостоятельной работы | Компьютер, мебель, учебно-методические пособия, учебно-наглядные материалы.<br>Программное обеспечение<br>1. LIBREOFFICE |

| | | |
|---|---|---|
| | | 2. Операционная система Astra Linux<br>3. Браузер Google Chrome |
| Аудитория 422 а(ФМ) | Для хранения оборудования | Ноутбук.<br>Программное обеспечение<br>1. LIBREOFFICE<br>2. Операционная система Astra Linux<br>3. Браузер Google Chrome |