

Б1.В.02

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	<i>Объектно-ориентированное программирование</i>
По направлению подготовки	<i>09.03.03</i>
	<i>«Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Прикладная информатика в цифровой экономике»</i>
Форма обучения	<i>Очно-заочная</i>

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы «Прикладная информатика в цифровой экономике» (уровень бакалавриата) в процессе обучения по дисциплине «Объектно-ориентированное программирование» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Результаты освоения образовательной программы (компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения по дисциплине	Этапы формирования компетенции (семестры и темы)	Оценочные средства
ПК-2 Способен проектировать информационные системы	ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения	знать: - основные положения и базовые понятия объектно-ориентированной технологии программирования; - этапы и правила проектирования отдельного класса и семейств классов. уметь: - выявлять классы в прикладной области, их данные, методы и операции; - устанавливать связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области; владеть: навыками детализации проектируемых классов;	Семестр 3 Тема 1. Введение в разработку объектно-ориентированных программ Тема 2. Основы объектно-ориентированного программирования Тема 3. Разработка класса Тема 4. Отношения между классами, семейства классов	- устный опрос по теме 1 - доклад-презентация по теме 2 - практические работы по теме 3 - курсовая работа по теме 4; - устный ответ на зачете с оценкой; - защита курсовой работы;
ПК-3 Способен разрабатывать информационные системы	ПК-3.1 - Осуществляет кодирование на современных языках	знать: - формат описания классов и их семейств на языке C#; уметь: - описывать отдельные	Семестр 3 Тема 3. Разработка класса Тема 4. Отношения	- устный опрос по теме 3 - практические работы по темам 3,5; - курсовая работа

	программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем	классы и их семейства для решения прикладных задач; - создавать программы и программные прототипы в объектно-ориентированной технологии; владеть: - навыками работы с библиотечными классами; - навыками отладки и тестирования работы классов и их семейств.	между классами, семейства классов Тема 5. Технология разработки компонентов	по теме 4; - устный ответ на зачете с оценкой; - защита курсовой работы;
ПК-4 Способен управлять процессами создания информационных систем	ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи	знать: - суть объектно-ориентированной технологии программирования, ее преимущества для конечного пользователя, правила проектирования семейств для прикладной области; уметь: - представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия; владеть: - навыками публичного выступления и демонстрации программной разработки;	Семестр 3 Тема 1. Введение в разработку объектно-ориентированных программ Тема 2. Основы объектно-ориентированного программирования Тема 4. Отношения между классами, семейства классов	- устный опрос по теме 1 - доклад-презентация по теме 2 - устный ответ на зачете с оценкой; - защита курсовой работы по теме 4;

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

*Примерные вопросы для проведения устного опроса
по теме «Введение в разработку объектно-ориентированных программ»*

1. Суть структурно-функционального подхода к созданию прикладных программ.
2. Суть объектно-ориентированного подхода к созданию прикладных программ.
3. Причины появления объектно-ориентированной технологии программирования.
4. Базовые принципы работы человеческого интеллекта.
5. Объект как основной компонент программы.
6. Отличия объектно-ориентированной программы от других программ.
7. Преимущества объектно-ориентированной технологии программирования.
8. Недостатки объектно-ориентированной технологии программирования.
9. Суть инкапсуляции, ее связь с принципами работы человеческого интеллекта.
10. Суть наследования, ее связь с принципами работы человеческого интеллекта.
11. Суть полиморфизма, ее связь с принципами работы человеческого интеллекта.

*Примерные вопросы для проведения устного опроса
по теме «Разработка класса»*

1. Характеристики типа «класс».
2. Уровни доступа к данным и методам класса
3. Виды данных класса
4. Виды методов класса
5. Правила описания конструктора
6. Виды конструкторов.
7. Правила вызова конструкторов
8. Правила описания деструктора.
9. Порядок вызова деструктора.
10. Понятие и виды свойств.
11. Правила описания свойства.
12. Формат объявления объекта.
13. Формат обращения к данным и методам объекта.
14. Формат перегрузки унарного оператора.
15. Формат перегрузки бинарного оператора.
16. Работа операторов с объектами класса

Критерии оценивания устного опроса:

- оценка *«отлично»* выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других обучающихся или задавал им дополнительные вопросы;
- оценка *«хорошо»* выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;
- оценка *«удовлетворительно»* выставляется обучающемуся, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других обучающихся, задавал им дополнительные вопросы;
- оценка *«неудовлетворительно»* выставляется обучающемуся, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Подготовка и выступление с докладом-презентацией
Примерные темы для подготовки докладов
по теме «Основы объектно-ориентированного программирования»

1. Автор объектно-ориентированной технологии программирования (Б.Страуструп) о ее достоинствах, перспективах использования и сложностях применения.
2. Суть инкапсуляции, ее преимущества, сложности реализации.
3. Состав класса как пользовательского типа данных, основные принципы проектирования отдельного класса.
4. Сравнительный анализ синтаксиса описания отдельного класса в разных языках программирования (любые два на выбор).
5. Наследование классов как базовый принцип объектно-ориентированной технологии программирования.
6. Суть понятия «полиморфизм» в разных научных областях и программировании. Причины и перспективы применения полиморфизма в программировании. Виды полиморфизма.

Требования к презентации:

1. Доклад не должен занимать более 15 минут.
2. На слайдах должны быть представлены ключевые моменты выступления, необходимые форматы и программный код. Допустимы графические образы.
3. Текст на слайды выносить в минимальном объеме, а его размер должен обеспечивать хорошую читаемость слайдов.
4. Презентация должна дополнять доклад.

Критерии оценивания доклада-презентации:

- оценка «отлично» выставляется обучающемуся, если доклад-презентация полностью раскрывает тему, структура презентации логично выстроена, обучающийся хорошо ориентируется в материале и текст доклада, в основном, рассказывает (а не читает), отвечает на дополнительные вопросы; сопровождающая доклад презентация выполнена в соответствии с требованиями и дополняет доклад, докладчик уложился в регламент;
- оценка «хорошо» выставляется обучающемуся, если доклад-презентация в целом раскрывает тему, но остаются не освещенные стороны вопросы, не на все дополнительные вопросы обучающийся может дать ответ; структура презентации логично выстроена, текст доклада обучающийся, в основном, рассказывает (а не читает); презентация выполнена в соответствии с требованиями, но есть ряд замечаний по ее оформлению, доклад и презентация дополняют друг друга, докладчик уложился в регламент;
- оценка «удовлетворительно» выставляется обучающемуся, если его доклад-презентация не раскрывает тему в достаточной мере, не отвечает на дополнительные вопросы; к оформлению презентации есть много замечаний; презентация выполнена не по требованиям, доклад дублирует презентацию, а не дополняет ее, обучающийся не укладывается в регламент или наоборот;
- оценка «неудовлетворительно» выставляется обучающемуся, если доклад-презентация отсутствует или совсем не раскрывает тему, не соответствует требованиям, обучающийся не может ответить на дополнительные вопросы.

Практические работы

Практическое задание по теме «Разработка класса»

Практическое задание:

1. Описать класс MyString:

```
class MyString
{
    public string Stroka;
    public void Print() {Console.WriteLine(Stroka); return;}
    public void Copy (int Count) {for (int i=1; i<=Count; i++) Print(); return;}}
```

2. Объявить объект, проинициализировать его данные и вызвать доступные функции
3. Для каждой строки функции Main подписать комментарии о выполняемых действиях.
4. Добавить в класс MyString перегрузку метода Print для вывода строки в переданные координаты.
5. Добавить в класс конструктор по умолчанию. Проверить работу конструктора.
6. Добавить в класс конструктор-инициализатор, используя модификатор this. Проверить работу конструктора.
7. Изменить модификатор доступа к данному Stroka с public на protected (защищенный). Убедиться в том, что прямой доступ в данном случае невозможен.
8. Сформировать в классе свойство Stroka_ для чтения и записи закрытого данного Stroka, используя в записи проверку записываемого значения (оно не должно быть пустым).
9. *Добавить в класс закрытое данное Dlina (аналогия свойства Length стандартного класса string), его инициализацию в конструкторах, соответствующее свойство только для чтения, изменение во всех участках кода, где происходит изменение строки.
10. Описать в классе унарный оператор «-» для конвертирования строки (запись в обратном порядке). Проверить работу оператора.
11. *Добавить в класс бинарный оператор «+» для сложения двух строк, используя стандартный метод Concat. Проверить работу оператора.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание было выполнено полным объеме, текст программы хорошо структурирован, в тексте программы присутствуют комментарии, предложен корректный (логично выстроенный) программный код проверки всех возможностей класса;
- оценка «хорошо» выставляется обучающемуся, если задание было выполнено почти в полном объеме (могут отсутствовать пункты со звездочкой), в тексте программы могут отсутствовать комментарии, присутствует программный код проверки всех возможностей класса;
- оценка «удовлетворительно» выставляется обучающемуся, если задание было выполнено частично, но не менее 50% (5-6 пунктов), присутствует программный код проверки описанных возможностей класса;
- оценка «неудовлетворительно» выставляется обучающемуся, если задание было выполнено менее, чем на 50% (менее 5 пунктов); программа фактически не работает.

Практическое задание по теме «Отношения между классами, семейства классов»

Задание:

1. Открыть консольное приложение с выполненным предыдущим заданием (класс MyString).

2. Добавить новый класс TimeString как потомка класса MyString (перед классом Program) для обработки форматированной строки, хранящей время в коротком формате:

```
class TimeString: MyString {}
```

3. Для проверки работоспособности класса необходимо объявить объект, проинициализировать унаследованные данные и вызвать доступные функции:

```
static void Main (string [] args)
{
    TimeString Str2=new TimeString ();
    Str2.Print(); Console.ReadLine();
    Str2.Stroka_="свойство"; Str2.Print(); Console.ReadKey();
    Console.Clear(); Str2.Copy(15); Console.ReadKey();
}
```

4. В классе в комментариях описать унаследованные данные и действия.

5. Добавить в класс закрытые данные: hour (час), minutes (минуты), div (разделитель в формате времени). Разделителю присвоить значение «:».

6. Сформировать в наследнике конструктор по умолчанию, изменив модификатор доступа к данному Stroka при необходимости:

```
public TimeString () {this.hour=0; this.minutes=0; Stroka="00"+div+"00";}
```

7. Проверить работу конструктора:

```
static void Main (string [] args)
{
    TimeString Str2=new TimeString ();
    Str2.Print(); Console.WriteLine(Str2.Dlina_);
    Console.ReadKey();
}
```

8. Изменить конструктор таким образом, чтобы длина тоже инициализировалась:

```
public TimeString () : base("00:00") {this.hour=0; this.minutes=0; }
```

9. Выполнить пункт 7.

10. В классе TimeString разработать свойства доступа к закрытым данным hour и minutes таким образом, чтобы производился контроль по максимальным значениям минут и часов. Проверить работу свойств.

11. Разработать конструктор-инициализатор с параметрами для создания форматированной строки, хранящей время в коротком формате с проверкой корректности передаваемых данных.

12. Перегрузить оператор сложения двух форматированных строк, контролируя результат сложения минут (40 мин + 40 мин = 1 час 20 мин) и часов.

13. Отредактировать наследника таким образом, чтобы можно было задавать короткий или полный формат отображения времени (добавить данное, отредактировать конструктор, перегрузить конструктор).

14. Создать класс-потомок MyFIO на основе родителя MyString со следующей структурой:

а) закрытые методы класса предназначены для вырезки данного из исходной строки (фамилии, имени, отчества) с приведением внешнего вида к стандартному: первая буква заглавная;

MyFIO
- pr_fam : string - pr_name : string - pr_otch : string
+ fam {get}: string + name {get}: string + otch {get}: string + fio {get}: string + iofo {get}: string + fio_pub {get}: string
+ MyFIO () + MyFIO (stroka: string) - sub_fam(): string - sub_name(): string - sub_otch(): string

- б) приватные данные, предназначенные для хранения вырезанных слов, инициализируются в конструкторах вызовом методов класса;
- в) все свойства потомка – только для чтения – возвращают значения закрытых данных или формируют из них следующие конструкции: «Фамилия И.О.», «И.О. Фамилия», «Фамилия, И.О.».
15. Проверить работу всех свойств потомка выводом их значений на экран в функции Main.
16. Проверить работу классов в консольном приложении

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание было выполнено полным объеме, текст программы хорошо структурирован, в тексте программы присутствуют комментарии, предложен корректный (логично выстроенный) программный код проверки всех возможностей класса;
- оценка «хорошо» выставляется обучающемуся, если задание было выполнено почти в полном объеме и присутствует программный код проверки всех возможностей классов, программа работает с незначительными ошибками (в логике или в эффективности кода), в тексте программы могут отсутствовать комментарии;
- оценка «удовлетворительно» выставляется обучающемуся, если задание было выполнено частично, но не менее 60% (может отсутствовать потомок MyFIO, но предыдущий потомок разработан в полном объеме), присутствует программный код проверки описанных возможностей разработанных классов;
- оценка «неудовлетворительно» выставляется обучающемуся, если в задание было выполнено менее 9 пунктов (менее 60%).

Практическое задание по теме «Технология разработки компонентов»

Задание:

1. Создать проект статической библиотеки (Файл/Создать/Проект):
 - a. Развернуть установленные шаблоны и выбрать Win32, затем Консольное приложение Win32.
 - b. Укажите имя для проекта, например, MathFuncsLib, и имя для решения, например, StaticLibrary.
 - c. На странице Обзор диалогового окна Мастер приложений Win32 нажмите кнопку Далее. На странице Параметры приложения в разделе Тип приложения выберите Статическая библиотека, в разделе Дополнительные параметры снимите флажок Предкомпилированный заголовок.
2. Добавьте класс в статическую библиотеку: откройте контекстное меню в обозревателе решений и добавьте новый элемент, в разделе Visual C# выберите Код и добавьте код вашего класс.
3. Создайте консольное приложение (Файл / Создать / Проект, Win32, Консольное приложение Win32), в раскрывающемся списке рядом с полем Решение выберите Добавить в решение.
4. После создания консольного приложения будет создана пустая программа. Для использования класса из статической библиотеки необходимо сослаться на эту библиотеку. Для этого откройте контекстное меню в обозревателе решений, а затем выберите пункт Ссылки. В диалоговом окне Страницы свойств разверните узел Общие свойства, выберите .NET Framework и ссылки и нажмите кнопку Добавить новую ссылку. На вкладке Проекты установите флажок для статичной библиотеки.

5. Создайте код, использующий возможности вашего класса и проверьте его работу.
6. Повторите действия еще для одного класса.

Критерии оценивания:

- оценка «*отлично*» выставляется обучающемуся, если задание было выполнено, доступ к обоим классам осуществлен, предложен корректный (логично выстроенный) программный код проверки всех возможностей внедренных классов;
- оценка «*хорошо*» выставляется обучающемуся, если задание было выполнено только для одного класса, предложен корректный (логично выстроенный) программный код проверки всех возможностей внедренного класса;
- оценка «*удовлетворительно*» выставляется обучающемуся, если студент добавил класс в библиотеку, но не смог полноценно обратиться к нему;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если задание не было выполнено совсем, класс не был добавлен в библиотеку.

Курсовая работа

к теме «Отношения между классами, семейства классов»

Цель: проектирование, программирование и тестирование семейства классов для заданной предметной области.

Общее задание:

1. Проанализировать предметную область Вашего варианта (допустимо предложить собственный вариант) и выделить в ней не менее 4 ключевых классов.
2. Установить связи между классами и спроектировать их иерархию (семейство). Допустимо в качестве родителя использовать абстрактный класс.
3. Спроектировать структуру каждого класса (см. требования ниже)
4. Описать семейство классов в виде диаграммы классов (UML, освоить самостоятельно)
5. Запрограммировать классы и семейство.
6. Разработать программу для демонстрации возможностей семейства классов.
7. Описать проектную разработку (см. требования ниже)

Требования к структуре класса:

1. Достаточное количество данных для полноценного функционирования объекта (исключение – абстрактный класс).
2. Достаточное количество методов для полноценного функционирования объекта (исключение – абстрактный класс).
3. Все виды доступа к данным и методам класса.
4. Обязательное наличие свойств, в том числе «только-для-чтения».
5. Конструктор по умолчанию, инициализатор, копирующий.
6. Деструктор.
7. Три операции на выбор, но обязательно хотя бы одна - логическая.
8. Полиморфные, виртуальные или абстрактные методы.

Требования к пояснительной записке:

1. Титульный лист.
2. Введение (значимость ООП для программирования)
3. Описание предметной области
4. Проектирование классов – перечень выделенных классов, диаграмма классов, описание связей и структуры классов в виде схемы в любой методологии (например, UML) и необходимого

описания назначения данных и методов, обоснования выбора свойств и операций, описания в целом идеи организации полиморфизма.

5. Кодирование классов – описание особенностей и/или сложностей разработки (код – в Приложении).
6. Тестирование классов – описание сценария проверки работоспособности семейства классов: проверка конструкторов и деструктора, реализация проверок внутри свойств, реализация функциональных возможностей классов, качество работы операций, демонстрация механизма принятия решения (код программы – в Приложении). Экранные образы (скриншоты) работы программы.
7. Заключение (подведение итогов).
8. Список литературы.
9. Приложение.

Варианты предметных областей:

1. Электронный документооборот компании.
2. Складской учет поступающих товаров.
3. Информационные сообщения программного продукта.
4. Математические выражения и/или функции.
5. Персонажи развивающих игр для детей.
6. Библиотечный учет поступающих публикаций.
7. Результаты образовательной и/или учебной деятельности.
8. Тестирование знаний обучающихся
9. Персонажи обучающих программ
10. Обработка динамичных (связанных) данных (вектор, графы, деревья)
11. Схематизация деятельности компании

Критерии оценивания хода выполнения курсовой работы:

- оценка *«отлично»* выставляется обучающемуся, если работа над курсовым проектом идет в соответствии с установленным графиком, модель семейства обсуждена с преподавателем, демонстрируемые отдельные фрагменты кода написана стилистически и синтаксически грамотно, обучающийся старается соблюдать все требования к разработке;
- оценка *«хорошо»* выставляется обучающемуся, если работа над курсовым проектом идет в соответствии с установленным графиком, но есть отдельные паузы в работе, модель семейства обсуждена с преподавателем, но есть неотработанные замечания, демонстрируемые отдельные фрагменты кода написана с незначительными ошибками, которые требуют доработки, обучающийся старается соблюдать требования к разработке;
- оценка *«удовлетворительно»* выставляется обучающемуся, если работа над курсовым проектом началась, но график выполнения не соблюдается, модель семейства не обсуждена с преподавателем, демонстрируемые отдельные фрагменты кода написаны с ошибками, обучающийся не старается соблюдать требования к разработке;
- оценка *«неудовлетворительно»* выставляется обучающемуся, если работа над курсовым проектом не началась, модель и код ни разу не демонстрировался преподавателю.

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ОЦЕНКА ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ

ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ (ПРОЕКТА) – 3 СЕМЕСТР

Выполнение курсовой работы позволяет практически закрепить знания, умения, навыков по следующим компетенциям:

Раздел пояснительной записки (выполняемое задание)	Описание содержания	Формируемые и проверяемые компетенции (индикаторы)
Введение	Значимость ООП для современных разработок, ее преимущества для пользователя	ПК-2.2, ПК-4.3
Описание предметной области	С точки зрения выделения классов	ПК-2.2, ПК-4.3
Проектирование классов	Перечень выделенных классов, диаграмма классов, описание данных и методов, обоснование выбора свойств и операций, идеи полиморфизма	ПК-2.2
Кодирование классов	Описание особенностей и/или сложностей разработки	ПК-3.1
Тестирование классов	Описание сценария проверки работоспособности, полученные результаты	ПК-3.1
Приложение	Код семейства	ПК-3.1

Формой контроля выступает защита курсовой работы:

- к защите допускаются обучающиеся, представившие работающий программный код и пояснительную записку;
- демонстрация работы программы на проекторе перед аудиторией;
- ответы на дополнительные вопросы аудитории.

ОЦЕНКА ЗНАНИЙ ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ – 3 СЕМЕСТР

Компетенция ПК-2 - Способен проектировать информационные системы

Индикатор ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся знает: основные положения и базовые понятия объектно-ориентированной технологии программирования; этапы и правила проектирования отдельного класса и семейств классов.

Оценка достижения обучающимся запланированного результата обучения осуществляется по полноте представленной пояснительной записки во время защиты курсовой работы (разделы «Введение», «Описание предметной области», «Проектирование классов») и ответов на вопросы во время защиты:

1. Поясните набор классов в семействе и их связь с предметной областью?

2. Какие структурные элементы класса реализованы в работе? Почему именно такая «инкапсуляция» предложена вами? Какие преимущества дает инкапсуляция разработчику?
3. Поясните перечень определенных внутри классов свойств и их связь с деятельностью пользователя?
4. Какова идея наследования в семействе?
5. Какова идея полиморфизма в семействе?
6. Какие операции были определены Вами для работы классов и почему? Как это связано с предметной областью?

Компетенция ПК-3 - Способен разрабатывать информационные системы

Индикатор ПК-3.1 - Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся знает: формат описания классов и их семейств на языке C#.

Оценка достижения обучающимся запланированного результата обучения осуществляется по полноте представленной пояснительной записки во время защиты курсовой работы (разделы «Кодирование классов», «Тестирование классов», «Приложение») и ответов на вопросы во время защиты:

1. Какие структурные элементы (ресурсы) класса скрыты? Чем это обосновано?
2. Какие структурные элементы (ресурсы) класса полностью открыты? Чем это обосновано?
3. Какие методы родителя являются полиморфными и почему?
4. Какие методы объявлены виртуальными и почему?
5. Какова идея полиморфизма в семействе?
6. Какие виды конструкторов были реализованы вами и почему?
7. Какие исходные данные необходимы для создания классов?
8. Использовались ли при разработке библиотечные ресурсы? Какие и зачем?
9. Каким образом производилось тестирование работоспособности отдельных классов и семейства в целом? Какие были получены результаты?

Компетенция ПК-4 - Способен управлять процессами создания информационных систем

Индикатор ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи

Обучающийся знает: суть объектно-ориентированной технологии программирования, ее преимущества для конечного пользователя, правила проектирования семейств для прикладной области.

Оценка достижения обучающимся запланированного результата обучения осуществляется по полноте представленной пояснительной записки во время защиты курсовой работы (разделы «Введение», «Описание предметной области») и ответов на вопросы во время защиты:

1. Каковы преимущества объектной технологии для разработчика ПО?
2. Каковы преимущества объектной технологии для пользователя ПО?
3. Каким образом производилось выделение классов (их данных и методов) из предметной области?

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ПО РЕЗУЛЬТАТАМ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ – 3 СЕМЕСТР

Компетенция ПК-2 - Способен проектировать информационные системы

Индикатор ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся умеет: выявлять классы в прикладной области, их данные, методы и операции; устанавливать связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области.

Обучающийся владеет: навыками детализации проектируемых классов.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения раздела курсовой работы «Проектирование классов», представления модели семейства, пояснения проектных решений.

Компетенция ПК-3 - Способен разрабатывать информационные системы

Индикатор ПК-3.1 - Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся умеет: описывать отдельные классы и их семейства для решения прикладных задач; создавать программы и программные прототипы в объектно-ориентированной технологии.

Обучающийся владеет: навыками работы с библиотечными классами; навыками отладки работы классов и их семейств.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения раздела курсовой работы «Кодирование классов», «Тестирование классов», представления работоспособного кода и его пояснения.

Компетенция ПК-4 - Способен управлять процессами создания информационных систем

Индикатор ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи

Обучающийся умеет: представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия.

Обучающийся владеет: навыками публичного выступления и демонстрации программной разработки.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам защиты курсовой работы: представление презентации с проектными решениями, в том числе разработанной для удаленного взаимодействия с аудиторией (с соблюдением правил построения и демонстрации презентаций в удаленном формате), демонстрации работы кода, ответов на вопросы аудитории.

ОЦЕНКА ЗНАНИЙ ОБУЧАЮЩИХСЯ НА ЗАЧЕТЕ С ОЦЕНКОЙ – 8 СЕМЕСТР

Список вопросов для подготовки к зачету с оценкой

Компетенция ПК-2 - Способен проектировать информационные системы

Индикатор ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся знает: основные положения и базовые понятия объектно-ориентированной технологии программирования; этапы и правила проектирования отдельного класса и семейств классов.

1. Существующие технологии программирования.
2. Базовые понятия и принципы объектно-ориентированного программирования.
3. Объектно-ориентированные языки программирования. Представление программы в объектной технологии.
4. Этапы проектирования отдельного класса
5. Этапы проектирования семейства классов

Компетенция ПК-3 - Способен разрабатывать информационные системы

Индикатор ПК-3.1 - Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся знает: формат описания классов и их семейств на языке C#.

1. Описание класса как пользовательского типа данных. Данные класса. Объекты как экземпляры классов.
2. Методы класса. Конструкторы класса. Деструктор и «сборщик мусора».
3. Перегрузка операторов класса.
4. Свойства класса.
5. Наследование. Статические, абстрактные и виртуальные методы.
6. Ранний и поздний полиморфизм. Таблица виртуальных методов.
7. Абстрактные методы и класса.
8. Интерфейсы.
9. Проектирование семейства классов.
10. Методы и средства программирования компонентов.
11. Технология создание собственных компонентов.
12. Разработка визуального компонента. Внедрение нового компонента в существующую библиотеку.

Компетенция ПК-4 - Способен управлять процессами создания информационных систем

Индикатор ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи

Обучающийся знает: суть объектно-ориентированной технологии программирования, ее преимущества для конечного пользователя, правила проектирования семейств для прикладной области.

1. Причины появления и актуальность объектно-ориентированного программирования.
2. Области применения объектно-ориентированного программирования. Преимущества применения объектно-ориентированного программирования для конечного пользователя.
3. Правила выделения классов в предметной области и их семейств

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ НА ЗАЧЕТЕ С ОЦЕНКОЙ –
8 СЕМЕСТР

Компетенция ПК-2 - Способен проектировать информационные системы

Индикатор ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения

Обучающийся умеет: выявлять классы в прикладной области, их данные, методы и операции; устанавливать связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области.

Обучающийся владеет: навыками детализации проектируемых классов.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения курсовой работы, в которой необходимо спроектировать каждый отдельный класс семейства и родственные связи между ним, представив проектное решение в схематичном образе и необходимого поясняющего текста.

Компетенция ПК-3 - Способен разрабатывать информационные системы

Индикатор ПК-3.1 - Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем

Обучающийся умеет: описывать отдельные классы и их семейства для решения прикладных задач; создавать программы и программные прототипы в объектно-ориентированной технологии.

Обучающийся владеет: навыками работы с библиотечными классами; навыками отладки работы классов и их семейств.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выполнения практических заданий с учетом курсовой работы, в которой необходимо написать код спроектированного семейства классов и код программы для его тестирования.

Компетенция ПК-4 - Способен управлять процессами создания информационных систем

Индикатор ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи

Обучающийся умеет: представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия.

Обучающийся владеет: навыками публичного выступления и демонстрации программной разработки.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам выступления с докладом-презентацией, а также по результатам публичной защиты курсовой работы, в том числе с презентацией, разработанной для удаленного взаимодействия с аудиторией (с соблюдением правил построения и демонстрации презентаций в удаленном формате).

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Для контроля усвоения обучающимися данной дисциплины и достижения запланированных результатов обучения учебным планом предусмотрены две формы аттестации: зачет с оценкой, проводимый в форме устного ответа на теоретический вопрос, и

защита курсовой работы. При выставлении оценки на зачете с оценкой учитываются результаты текущего контроля, отражающие сформированность практических умений и навыков обучающегося. Во время обучения обучающиеся участвуют в разных мероприятиях текущего контроля: в устных теоретических опросах, выполняют практические задания, выступают с докладом. Для допуска к зачету с оценкой обучающемуся необходимо выступить с докладом и выполнить практические задания, а также выполнить курсовую работу.

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Компетенция ПК-2 - Способен проектировать информационные системы					
Индикатор ПК-2.2 - Разрабатывает модель информационной системы, в том числе с опорой на современные облачные решения					
Знать: - основные положения и базовые понятия объектно-ориентированной технологии программирования; - этапы и правила проектирования отдельного класса и семейств классов.	Не знает	Фрагментарные знания основных положений и базовых понятий объектно-ориентированной технологии программирования, этапов и правил проектирования отдельного класса и семейств классов	Общие, но не структурированные знания основных положений и базовых понятий объектно-ориентированной технологии программирования, этапов и правил проектирования отдельного класса и семейств классов	Сформированные, но содержащие отдельные пробелы знания основных положений и базовых понятий объектно-ориентированной технологии программирования, этапов и правил проектирования отдельного класса и семейств классов	Сформированные систематизированные прочные знания основных положений и базовых понятий объектно-ориентированной технологии программирования, этапов и правил проектирования отдельного класса и семейств классов
Уметь: - выявлять классы в прикладной области, их данные, методы и операции; - устанавливать связи между классами и проектировать их семейства для разработки	Не умеет	Частично освоенные умения выявлять классы в прикладной области, их данные, методы и операции, устанавливать связи между классами и проектировать их семейства	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполняемые умения выявлять классы в прикладной области, их данные, методы и	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения выявлять классы в прикладной области, их	Успешно сформированные, самостоятельно и осознанно выполняемые умения выявлять классы в прикладной области, их данные, методы и устанавливать

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
информационных систем в заданной прикладной области;		для разработки информационных систем в заданной прикладной области;	операции, устанавливать связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области;	данные, методы и операции, устанавливать связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области;	связи между классами и проектировать их семейства для разработки информационных систем в заданной прикладной области;
Владеть: - навыками детализации проектируемых классов;	Не владеет	Фрагментарно сформированные навыки детализации проектируемых классов	В целом сформированные, но выполняемые на элементарном уровне навыки детализации проектируемых классов	Сформированные, но не устойчивые в сложных ситуациях навыки детализации проектируемых классов	Успешно сформированные, устойчивые, технологически и грамотно выполняемые навыки детализации проектируемых классов
Компетенция ПК-3 - Способен разрабатывать информационные системы Индикатор ПК-3.1 - Осуществляет кодирование на современных языках программирования, в том числе используя возможности специализированных цифровых платформ для индивидуальной и совместной разработки информационных систем					
Знать: - формат описания классов и их семейств на языке C#	Не знает	Фрагментарные знания форматов описания классов и их семейств на языке C#	Общие, но не структурированные знания форматов описания классов и их семейств на языке C#	Сформированные, но содержащие отдельные пробелы знания форматов описания классов и их семейств на языке C#	Сформированные систематизированные прочные знания форматов описания классов и их семейств на языке C#
Уметь: - описывать отдельные классы и их семейства для решения	Не умеет	Частично освоенные умения описывать отдельные классы и их	В целом освоенные, но не подкрепляемые знаниями и самостоятельно	В целом сформированные и самостоятельно выполняемые,	Успешно сформированные, самостоятельно и осознанно выполняемые

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
прикладных задач; - создавать программы и программные прототипы в объектно-ориентированной технологии		семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии	стью выполнения умения описывать отдельные классы и их семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии	но не всегда интеллектуально обоснованные умения описывать отдельные классы и их семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии	умения описывать отдельные классы и их семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии
Владеть: - навыками работы с библиотечным и классами; - навыками отладки и тестирования работы классов и их семейств.	Не владеет	Фрагментарно сформированные навыки работы с библиотечным и классами, отладки и тестирования работы классов и их семейств	В целом сформированные, но выполняемые на элементарном уровне навыки работы с библиотечным и классами, отладки и тестирования работы классов и их семейств	Сформированные, но не устойчивые в сложных ситуациях навыки работы с библиотечным и классами, отладки и тестирования работы классов и их семейств	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки работы с библиотечным и классами, отладки и тестирования работы классов и их семейств
Компетенция ПК-4 - Способен управлять процессами создания информационных систем Индикатор ПК-4.3 - Осуществляет сопровождение информационных систем, в том числе с применением средств удаленного доступа и видеоконференцсвязи					
Знать: - суть объектно-ориентированной технологии программирования, ее преимущества для конечного пользователя,	Не знает	Фрагментарные знания сути объектно-ориентированной технологии программирования, ее преимуществ для конечного пользователя,	Общие, но не структурированные знания сути объектно-ориентированной технологии программирования, ее преимуществ для конечного	Сформированные, но содержащие отдельные пробелы знания сути объектно-ориентированной технологии программирования	Сформированные систематизированные прочные знания сути объектно-ориентированной технологии программирования

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
правила проектирования семейств для прикладной области		правил проектирования семейств для прикладной области	пользователя, правил проектирования семейств для прикладной области	ания, ее преимуществ для конечного пользователя, правил проектирования семейств для прикладной области	ания, ее преимуществ для конечного пользователя, правил проектирования семейств для прикладной области
Уметь: - представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия	Не умеет	Частично освоенные умения представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия	Успешно сформированные, самостоятельные и осознанно выполняемые представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств удаленного доступа и взаимодействия
Владеть: - навыками публичного выступления и демонстрации программной разработки	Не владеет	Фрагментарно сформированные навыки публичного выступления и демонстрации программной	В целом сформированные, но выполняемые на элементарном уровне навыки	Сформированные, но не устойчивые в сложных ситуациях навыки публичного	Успешно сформированные, устойчивые, технологически и грамотно выполняемые

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
		разработки	публичного выступления и демонстрации программной разработки	выступления и демонстрации программной разработки	навыки публичного выступления и демонстрации программной разработки

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценка за курсовую работу выставляется по результатам ее защиты с учетом результатов текущего контроля за выполнение курсовой работы в течение семестра.

Критерии оценивания для курсовой работы:

- оценка «*отлично*» выставляется обучающемуся, если задание курсовой работы было выполнено полном объеме, создано не менее трех классов, имеющих родственные отношения, в классах присутствуют все заявленные позиции, предложен корректный (логично выстроенный) программный код проверки всех возможностей классов, в ходе построения семейства не нарушены концептуальные положения объектной технологии, во время защиты индивидуального проектного задания обучающийся логически стройно, доходчиво и грамотно продемонстрировал весь функционал семейства классов, особенности их программной разработки и работы, полно и доступно отвечал на дополнительные вопросы; во время демонстрации не возникает программных ошибок; пояснительная записка соответствует установленным требованиям в полном объеме; график выполнения курсовой работы был соблюден и работа была представлена своевременно;
- оценка «*хорошо*» выставляется обучающемуся, если задание курсовой работы было выполнено не полном объеме, но хотя бы на 75%; создано не менее трех классов, имеющих родственные отношения, в классах присутствуют почти все заявленные позиции (например, вместо трех конструкторов только один, вместо двух свойств только одно и т.д.), предложен программный код проверки всех возможностей классов, во время защиты индивидуального проектного задания обучающийся достаточно подробно продемонстрировал функционал семейства классов и особенности его разработки, доступно отвечал на дополнительные вопросы; во время демонстрации не возникают программных ошибок (или иногда возникают, но быстро разрешаются); пояснительная записка соответствует установленным требованиям в полном объеме или с незначительными нарушениями; график выполнения курсовой работы был соблюден и работа была представлена своевременно;
- оценка «*удовлетворительно*» выставляется обучающемуся, если задание курсовой работы было выполнено частично, создано минимум два класса, имеющих родственные отношения; в классах присутствует большинство заявленных позиций (например, вместо трех конструкторов только один, вместо двух свойств только одно, мало данных и методов, а их набор отражает далеко не все возможности запланированного класса),

реализован хотя бы один виртуальный метод, есть программный код проверки возможностей классов, во время защиты индивидуального проектного задания обучающийся невянятно представлял функционал семейства классов, не смог выделить особенности их разработки, не смог ответить на все дополнительные вопросы; во время демонстрации возникают программные ошибки, но обучающийся тем или иным образом их объясняет; пояснительная записка не в полном объеме соответствует установленным требованиям; график выполнения курсовой работы не всегда соблюдался и/или работа была представлена не своевременно;

- оценка *«неудовлетворительно»* выставляется обучающемуся, если разработка отсутствует или разработан только один класс, или в представленном семействе не реализовано большинство заявленных позиций (более половины), в том числе механизм полиморфизма, или обучающийся не в состоянии объяснить представленный код; отсутствует пояснительная записка.

Критерии оценивания для зачета с оценкой:

При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля и критерии оценивания сформированности компетенций.

Оценка *«отлично»* выставляется обучающемуся, если:

- при ответе на теоретический вопрос были продемонстрированы глубокие, систематизированные и прочные знания (в соответствии с критериями сформированности компетенций), умения логично и обоснованно излагать материал, быстро ориентироваться в дополнительных вопросах и давать полные ответы;
- обучающийся участвовал во всех видах текущего контроля и получал по ним максимальные баллы (допустимо отдельные отметки «хорошо»).

Все вышеперечисленное говорит о сформированности запланированных компетенций на высоком уровне.

Оценка *«хорошо»* выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано хорошее, но недостаточно полное изложение материала, с отдельными пробелами в знаниях (возможно, незначительными неточностями в употреблении терминов), логичное изложение материала, но недостаточно быстрая ориентация в нем при ответе на дополнительные вопросы;
- обучающийся участвовал во всех видах текущего контроля и получал по ним в основном отметки «хорошо».

Все вышеперечисленное говорит о сформированности запланированных компетенций на хорошем уровне.

Оценка *«удовлетворительно»* выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано поверхностное владение материалом и фрагментарные знания, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, плохая ориентация в материале при ответе на дополнительные вопросы;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о сформированности запланированных компетенций на низком уровне.

Оценка *«неудовлетворительно»* выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано незнание значительного объема материала, «фрагментарность имеющихся знаний, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, неумение выделить главное, сделать выводы и обобщения, неспособность отвечать на дополнительные вопросы;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о частичной сформированности запланированных компетенций.

6. ЛИСТ СОГЛАСОВАНИЯ

Составил:

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)