

Кафедра _____ прикладной информатики высшей математики

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>Очная</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

Оглавление

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	3
2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	5
3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ.....	12
4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ.....	17
5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ.....	23
6. ЛИСТ СОГЛАСОВАНИЯ.....	45

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 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#; уметь: - описывать отдельные 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.	Семестр 3 Тема 3. Разработка класса Тема 4. Отношения между классами, семейства классов Тема 5. Технология разработки компонентов	- устный опрос по теме 3 - практические работы по темам 3,5; - курсовая работа 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

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

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;

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 полном 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):
 - а. Развернуть установленные шаблоны и выбрать Win32, затем Консольное приложение Win32.
 - б. Укажите имя для проекта, например, MathFuncsLib, и имя для решения, например, StaticLibrary.
 - в. 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 И 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, отражающего 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	Сформированные систематизированные прочные знания основных положений и базовых понятий объектно-ориентированной технологии программирования, этапов и правил проектирования отдельного класса и

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
				семейств классов	семейств классов
Уметь: - выявлять классы в прикладной области, их данные, методы и операции; - устанавливать связи между классами и проектировать их семейства для разработки информационн ых систем в заданной прикладной области;	Не умеет	Частично освоенные умения выявлять классы в прикладной области, их данные, методы и операции, устанавливать связи между классами и проектировать их семейства для разработки информационн ых систем в заданной прикладной области;	В целом освоенные, но не подкрепляемые знаниями и самостоятельно стью выполнения умения выявлять классы в прикладной области, их данные, методы и операции, устанавливать связи между классами и проектировать их семейства для разработки информационн ых систем в заданной прикладной области;	В целом сформирован ные и самостоятель но выполняемые, но не всегда интеллектуал ьно обоснованные умения выявлять классы в прикладной области, их данные, методы и операции, устанавливать связи между классами и проектироват ь их семейства для разработки информацион ных систем в заданной прикладной области;	Успешно сформированн ые, самостоятельн о и осознанно выполняемые умения выявлять классы в прикладной области, их данные, методы и операции, устанавливать связи между классами и проектировать их семейства для разработки информационн ых систем в заданной прикладной области;
Владеть: - навыками детализации проектируемы х классов;	Не владеет	Фрагментарно сформированн ые навыки детализации проектируемых классов	В целом сформированн ые, но выполняемые на элементарном уровне навыки детализации проектируемых классов	Сформирован ные, но не устойчивые в сложных ситуациях навыки детализации проектируем ых классов	Успешно сформированн ые, устойчивые, технологическ и грамотно выполняемые навыки детализации проектируемы х 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:	Не знает	Фрагментарн	Общие, но не	Сформирова	Сформирова

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
- формат описания классов и их семейств на языке C#		ые знания форматов описания классов и их семейств на языке C#	структурированные знания форматов описания классов и их семейств на языке C#	ные, но содержащие отдельные пробелы знания форматов описания классов и их семейств на языке C#	ные систематизированные прочные знания форматов описания классов и их семейств на языке C#
Уметь: - описывать отдельные классы и их семейства для решения прикладных задач; - создавать программы и программные прототипы в объектно-ориентированной технологии	Не умеет	Частично освоенные умения описывать отдельные классы и их семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии	В целом освоенные, но не подкрепляемые знаниями и самостоятельно стью выполнения умения описывать классы и их семейства для решения прикладных задач, создавать программы и программные прототипы в объектно-ориентированной технологии	В целом сформированные и самостоятельны о выполняемые, 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	Успешно сформированные, самостоятельно и осознанно выполняемые 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	Сформированные, но не устойчивые в сложных ситуациях навыки работы с библиотечным и классами, отладки и тестирования работы	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки работы с библиотечным и классами, отладки и

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
			работы классов и их семейств	классов и их семейств	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, правила проектирования семейств для прикладной области	Не знает	Фрагментарные знания сути объектно-ориентированной технологии программирования, ее преимуществ для конечного пользователя, правил проектирования семейств для прикладной области	Общие, но не структурированные знания сути объектно-ориентированной технологии программирования, ее преимуществ для конечного пользователя, правил проектирования семейств для прикладной области	Сформированные, но содержащие отдельные пробелы знания сути объектно-ориентированной технологии программирования, ее преимуществ для конечного пользователя, правил проектирования семейств для прикладной области	Сформированные систематизированные прочные знания сути объектно-ориентированной технологии программирования, 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	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал	Успешно сформированные, самостоятельные и осознанно выполняемые представлять программную разработку для потенциальных пользователей и демонстрировать ее функционал для обучения, в том числе с применением средств

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
		ия	применением средств удаленного доступа и взаимодействия	для обучения, в том числе с применением средств удаленного доступа и взаимодействия	удаленного доступа и взаимодействия
Владеть: - навыками публичного выступления и демонстрации программной разработки	Не владеет	Фрагментарно сформированные навыки публичного выступления и демонстрации программной разработки	В целом сформированные, но выполняемые на элементарном уровне навыки публичного выступления и демонстрации программной разработки	Сформированные, но не устойчивые в сложных ситуациях навыки публичного выступления и демонстрации программной разработки	Успешно сформированные, устойчивые, технологические и грамотно выполняемые 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.

5. ДИАГНОСТИЧЕСКИЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ УРОВЕНЬ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция ПК-2 - Способен проектировать информационные системы

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

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.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

Задание 1. В основе концепции объектно-ориентированного программирования лежит понятие:

- а) объект;*
- б) класс;*
- в) проект;*
- г) инкапсуляция.*

Ответ: а

Задание 2. Инкапсуляция – это:

- а) Возможность описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью;*
- б) Способность системы (программы) изменять алгоритм действий в зависимости от сложившейся программной ситуации ;*
- в) Объединение данных и методов, работающих с ними, в одном типе, что позволяет скрыть детали реализации от пользователя;*
- г) Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.*

Ответ: в

Задание 3. Наследование – это:

- а) Возможность описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью;*
- б) Способность системы (программы) изменять алгоритм действий в зависимости от сложившейся программной ситуации ;*
- в) Объединение данных и методов, работающих с ними, в одном типе, что позволяет скрыть детали реализации от пользователя;*
- г) Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.*

Ответ: а

Задание 4. Абстракция (абстрагирование) – это:

- а) Возможность описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью;*
- б) Способность системы (программы) изменять алгоритм действий в зависимости от сложившейся программной ситуации ;*
- в) Объединение данных и методов, работающих с ними, в одном типе, что позволяет скрыть детали реализации от пользователя;*
- г) Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.*

Ответ: г

Задание 5. Полиморфизм – это:

- а) Возможность описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью;*
- б) Способность системы (программы) изменять алгоритм действий в зависимости от сложившейся программной ситуации ;*
- в) Объединение данных и методов, работающих с ними, в одном типе, что позволяет скрыть детали реализации от пользователя;*
- г) Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.*

Ответ: б

Задание 6. Выберите корректное определение конструктора:

- а) Метод, адрес которого определяется непосредственно перед вызовом;*

- б) *Метод, отвечающий за создание объекта в памяти и выполнение необходимых стартовых действий;*
- в) *Метод, адрес которого определяется во время компиляции;*
- г) *Метод, вызываемый «сборщиком мусора» для выполнения необходимых завершающих действий;*
- д) *Метод без реализации.*

Ответ: б

Задание 7. Выберите корректное определение деструктора:

- а) *Метод, адрес которого определяется непосредственно перед вызовом;*
- б) *Метод, отвечающий за создание объекта в памяти и выполнение необходимых стартовых действий;*
- в) *Метод, адрес которого определяется во время компиляции;*
- г) *Метод, вызываемый «сборщиком мусора» для выполнения необходимых завершающих действий;*
- д) *Метод без реализации.*

Ответ: г

Задание 8. Выберите корректное определение виртуального метода:

- а) *Метод, адрес которого определяется непосредственно перед вызовом;*
- б) *Метод, отвечающий за создание объекта в памяти и выполнение необходимых стартовых действий;*
- в) *Метод, адрес которого определяется во время компиляции;*
- г) *Метод, вызываемый «сборщиком мусора» для выполнения необходимых завершающих действий;*
- д) *Метод без реализации.*

Ответ: а

Задание 9. Выберите корректное определение абстрактного метода:

- а) *Метод, адрес которого определяется непосредственно перед вызовом;*
- б) *Метод, отвечающий за создание объекта в памяти и выполнение необходимых стартовых действий;*
- в) *Метод, адрес которого определяется во время компиляции;*
- г) *Метод, вызываемый «сборщиком мусора» для выполнения необходимых завершающих действий;*
- д) *Метод без реализации.*

Ответ: д

Задание 10. Что такое класс?

- а) *Вид программы;*
- б) *Пользовательский тип;*
- в) *Вид подпрограммы;*

Ответ: б

ЗАДАНИЯ ОТКРЫТОГО ТИПА

Задание 11. Соотнесите указанные понятия с их описанием.

Понятия: 1) состояние объекта; 2) поведение объекта; 3) свойство объекта.

Описание: а) механизм работы с закрытым атрибутом объекта; б) набор методов (программный код), оперирующих над состоянием объекта; в) набор значений его атрибутов.

Ответ: 1 - в; 2 – б; 3 - а.

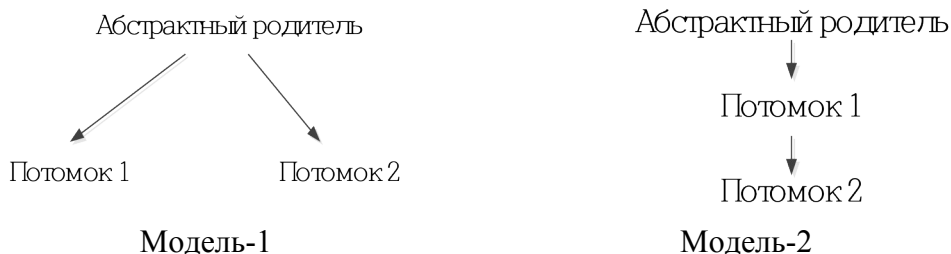
Задание 12. Что означает термин «наследование»?

Ответ: В производных классах присутствует часть состояния родительского класса.

Задание 13. В каких отношениях по отношению друг к другу могут находиться классы?

Ответ: Между классами существует два принципиальных вида отношений: наследование – когда на базе одного класс строится другой класс; вложенности (использование) – когда один класс (объект этого класса) используется в другом классе, т.е. является его элементом.

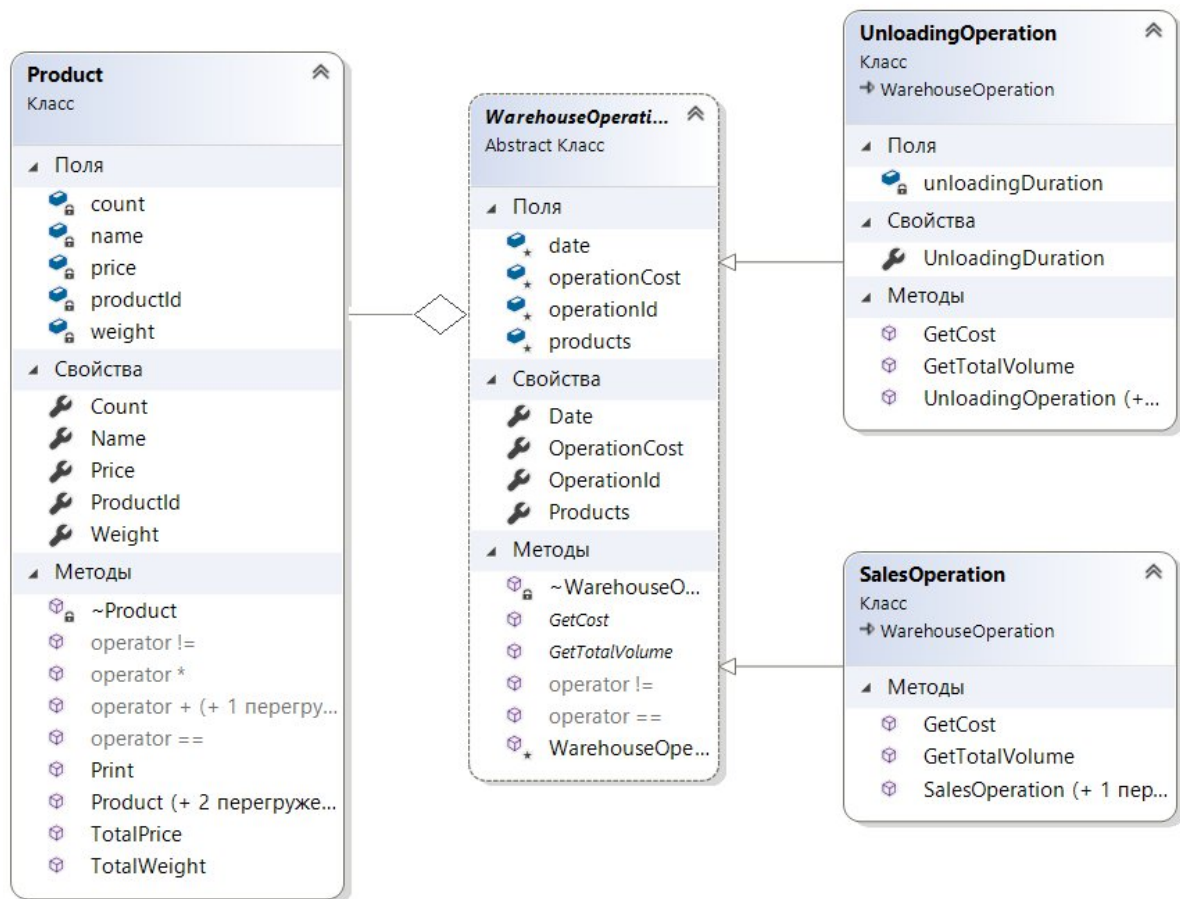
Задание 14. Опишите отличие двух моделей наследования?



Ответ: В модели-1 родственные связи существуют между парами классов «Родитель – Потомок1», «Родитель-Потомок2». При этом родитель содержит общую для обоих потомков часть реализации, включая общие абстрактные методы. Каждый потомок по-своему реализует абстрактные методы. В модели-2 родственные связи существуют между парами классов «Родитель – Потомок1», «Потомок1-Потомок2». В этой модели Потомок1 наследует часть реализации Родителя и определяет все абстрактные методы. Потомок 2 может просто унаследовать все, что есть в Потомке1.

Задание 15. Опишите связи между классами, представленными на рисунке?

Ответ: Отношение наследования задано между классами: WarehouseOperation и UnloadingOperation, WarehouseOperation и SalesOperation. Отношение использования задано между классами Product и WarehouseOperation (а также классами UnloadingOperation и SalesOperation как наследниками WarehouseOperation).



Задание 16. Задана предметная область «Математические функции». Предположите (выявите) классы с потенциальными родственными связями (не менее 3-х)

Ответ: Для данной предметной области можно спроектировать следующее семейство классов, связанных отношениями наследования: абстрактный родитель - функция (содержит аргументы, диапазон значений аргумента, массив результатов, экстремумы и т.п.) и от него серия потомков: линейная функция, парабола, гипербола.

Задание 17. Задана предметная область «Программные сообщения». Предположите (выявите) классы с потенциальными родственными связями (не менее 3-х)

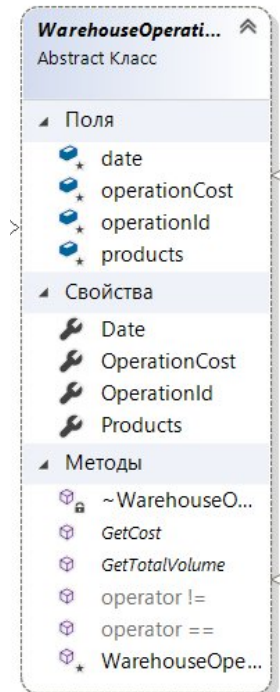
Ответ: Для данной предметной области можно спроектировать следующее семейство классов, связанных отношениями наследования: родитель – простейшее информационное сообщение, от него потомок – сообщение с элементами дизайна, от него потомок - сообщение и с predefined вариантами ответов.

Задание 18. Задана предметная область «Клиенты». Предположите (выявите) классы с потенциальными родственными связями (не менее 3-х)

Ответ: Для данной предметной области можно спроектировать следующее семейство классов, связанных отношениями наследования: абстрактный родитель - клиент (содержит общие данные и методы работы с клиентами) и от него два потомка: клиент - физическое лицо и клиент – юридическое лицо.

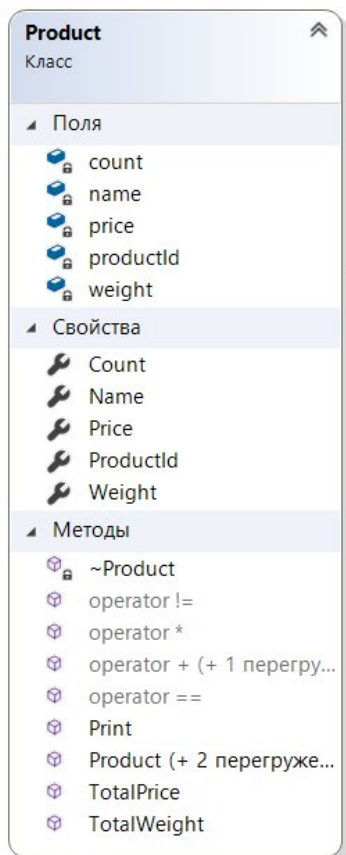
Задание 19. По схематичному представлению класса на рисунке опишите содержимое касса

Ответ: В классе WarehouseOperation представлено 4 переменных (поля) защищенного уровня доступа – date, operationCost, operationId, products. Для каждого данного запланировано свойство. В классе предусмотрены следующие методы: деструктор ~WarehouseOperation и конструктор WarehouseOperation, открытые методы GetCost и GetTotalVolume, перегруженные операторы парные сравнения != (неравенство) и ==(равенство).



Задание 20. По схематичному представлению класса на рисунке опишите содержимое касса

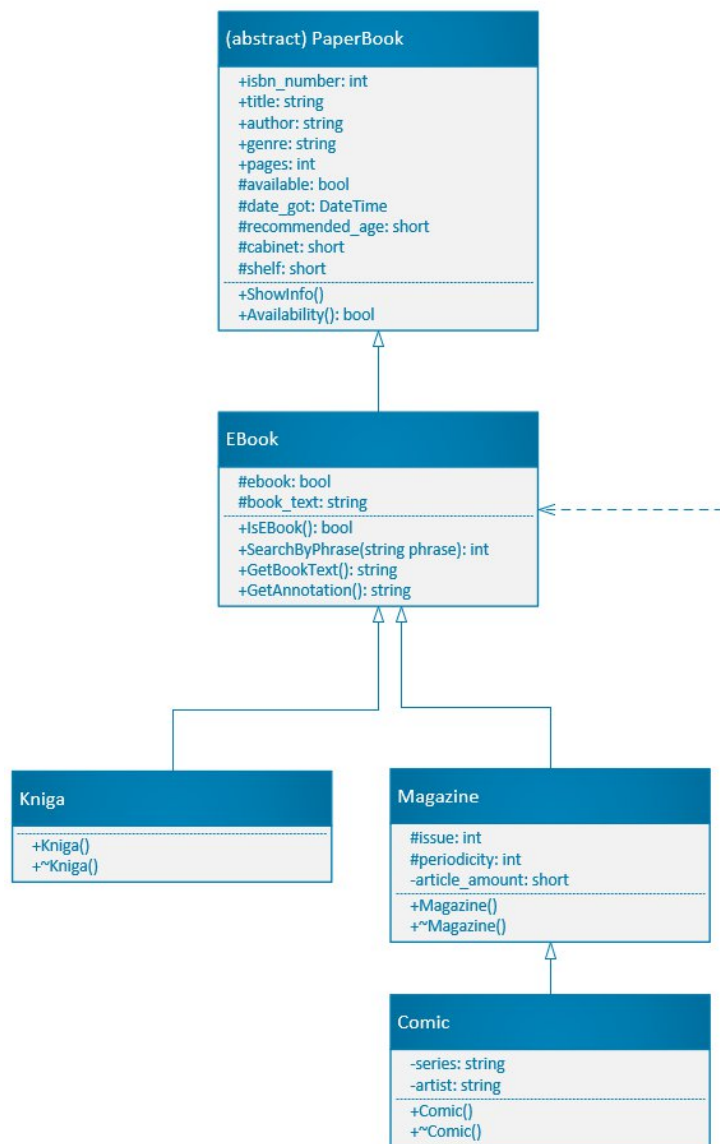
Ответ: В классе Product представлено 5 переменных (поля) закрытого уровня доступа – count, name, price, productId, weighth. Для каждого данного запланировано соответствующее свойство. В классе предусмотрены следующие методы: деструктор ~ Product и три перегруженных конструктора Product, открытые методы Print, TotalPrice, TotalWeight, перегруженные операторы парные сравнения != (неравенство) и ==(равенство), арифметические операторы умножения (*) и сложения (+). Оператор сложения имеет две реализации внутри класса.



Задание 21. Опишите связи между классами, представленными на рисунке?

Ответ: Отношение наследования задано между классами:

- 1) Абстрактным классом PaperBook и его потомков EBook,
- 2) Родительски классом EBook и его двумя потомками Книга и Magazine.
- 3) Родительски классом Magazine и его потомком Comic.



Задание 22. Спроектируйте содержание класса «Геометрическая фигура» так, чтобы в ней присутствовало как можно больше разных элементов (данные, свойства, методы)

Ответ:

Данные	- Длина фигуры (dlina) - Ширина фигуры (shirina)
Свойства	+Длина (Dlina) – для проверки корректности значения +Ширина (Shirina) – для проверки корректности значения
Конструкторы	+Square() – по умолчанию с присвоением единичных значений: +Square (int x, int y) – инициализатор
Деструктор	~Square():
Методы	Print(): печатает на консоли длину и ширину фигуры. Draw(): рисует фигуру на консоли.
Операторы	Оператор “+”: складывает длину и ширину двух фигур и передаёт значения в новый класс.

Задание 23. Спроектируйте содержание класса «Студент» так, чтобы в ней присутствовало как можно больше разных элементов (данные, свойства, методы)

Ответ:

Данные	private string FIO; Фамилия имя отчество
--------	--

	private int age; возраст private string group; группа private double srball; средний балл
Свойства	+ Группа (group_) - проверяем корректность значение . + Средний балл (srball_) - проверяем корректность значение .
Конструкторы	+ Student(): по умолчанию с присвоением значений по умолчанию + Student(string FIO, int age, string group, double srball) - инициализатор
Деструктор	~ Student ():
Методы	Print(): Печатает на консоли фио, возраст, группу и средний балл..
Операторы	Оператор "+": объединяет всю информацию о студенте в одну строку

Задание 24. Для семейства классов «Геометрические фигуры» предложите хотя бы один полиморфный метод и к нему виртуальный (ые) метод (ы).

Ответ: Для данного семейства классов можно реализовать полиморфный метод «Информация о фигуре», который будет строить образ фигуры и выводить текущие данные фигуры. К такому родительскому полиморфному методу необходимо в каждом потомке переопределить два виртуальных метода: 1) метод, который рисует фигуру (прямоугольник или круг или треугольник); 2) метод, который выводит текущие параметры (стороны прямоугольника, радиус круга, стороны треугольника и его углы).

Задание 25. Предложите интерфейс для семейства классов «Геометрические фигуры»

Ответ: Для данного семейства можно разработать интерфейс, который содержит прототипы функций: 1) вывод (рисовка) изображения фигуры; 2) удаление фигуры; 3) перемещение фигуры.

Компетенция ПК-3 - Способен разрабатывать информационные системы

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

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.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

Задание 1. Какое служебное слово используется для описания класса?

- a) *delegate;*
- б) *interface;*
- в) *class;*
- г) *struct.*

Ответ: в

Задание 2. Какое служебное слово используется для описания интерфейса?

- a) *delegate;*
- б) *interface;*
- в) *class;*
- г) *struct.*

Ответ: б

Задание 3. Какое служебное слово используется для описания делегатов?

- а) *delegate*;
- б) *interface*;
- в) *class*;
- г) *struct*.

Ответ: а

Задание 4. Какое служебное слово используется для обращения к родителю?

- а) *this*;
- б) *base*;
- в) *owner*;

Ответ: б

Задание 5. Какое служебное слово используется для обращения к своим ресурсам?

- а) *this*;
- б) *base*;
- в) *owner*;

Ответ: а

Задание 6. В каких элементах класса используется служебное слово *value*?

- а) конструкторы;
- б) абстрактные методы;
- в) свойства;
- г) операции.

Ответ: в

Задание 7. В каких элементах класса используется служебное слово *explicit*?

- а) конструкторы;
- б) абстрактные методы;
- в) свойства;
- г) операции.

Ответ: г

Задание 8. Для каких методов допустима и/или необходима перегрузка внутри класса?

- а) конструкторы;
- б) абстрактные методы;
- в) деструктор;
- г) операции.

Ответ: а, г

Задание 9. Для описания каких методов используется служебное слово *override*?

- а) конструкторы;
- б) абстрактные методы;
- в) деструктор;
- г) операции.

Ответ: б

Задание 10. Какие из методов нельзя перегрузить внутри класса?

- а) конструкторы;
- б) абстрактные методы;
- в) деструктор;
- г) операции.

Ответ: в

ЗАДАНИЯ ОТКРЫТОГО ТИПА

Задание 11. Сопоставьте обозначение уровня доступа и его описание. При ответе расположите их в порядке увеличения прав доступа и с их кратким описанием.

Уровни доступа: 1) protected; 2) private; 3) public.

Описание: а) закрытый доступ; б) общедоступный; в) защищенный доступ.

Ответ: private (2) - закрытый доступ (а), то есть к нему можно обращаться только в текущем классе; protected (1) – защищенный (в), разрешено обращаться как с текущего класса, так и с классов наследников; public (3) – общедоступный (б), разрешено обращаться из любого места программы.

Задание 12. По представленному описанию класса перечислите все открытые элементы класса и дайте их краткое обозначение.

```
class Avtor
{
    protected MyFIO avtor;
    protected string Titul;
    protected string vid;
    protected DateTime dt;
    public Avtor()
    {
        avtor = new MyFIO();
        Titul = "без названия";
        vid = "не определено";
        dt = new DateTime();
    }
    public Avtor(MyFIO fio, string titul, string vid, DateTime Date)
    {
        this.avtor = fio;
        this.Titul = titul;
        this.vid = vid;
        this.dt = Date;
    }
    protected virtual string Result()
    {
        return $"{avtor.fio_pub} {Titul} [Текст] : {vid} / {avtor.iof}. - {dt.Year}";
    }
    public void Show()
    {
        Console.WriteLine(Result());
    }
    ~Avtor()
    {
        Console.WriteLine("объект удалён");
    }
}
```

Ответ:

К открытым элементам класса относятся: конструктор по умолчанию (Avtor ()) и конструктор инициализатор (Avtor(MyFio fio, string titul, string Vid, DateTime Date)), метод Show.

Задание 13. По представленному описанию класса перечислите все защищенные элементы класса и дайте их краткое обозначение.

```

class Avtor
{
    protected MyFIO avtor;
    protected string Titul;
    protected string vid;
    protected DateTime dt;
    public Avtor()
    {
        avtor = new MyFIO();
        Titul = "без названия";
        vid = "не определено";
        dt = new DateTime();
    }
    public Avtor(MyFIO fio, string titul, string vid, DateTime Date)
    {
        this.avtor = fio;
        this.Titul = titul;
        this.vid = vid;
        this.dt = Date;
    }
    protected virtual string Result()
    {
        return $"{avtor.fio_pub} {Titul} [Текст] : {vid} / {avtor.iof}. - {dt.Year}";
    }
    public void Show()
    {
        Console.WriteLine(Result());
    }
    ~Avtor()
    {
        Console.WriteLine("объект удалён");
    }
}

```

Ответ:

К защищенным элементам класса относятся: переменные Titul и vid строкового типа, avtor типа MyFIO, dt типа DateTime; виртуальный метод Result, возвращающий сформированную строку

Задание 14. Определите по представленному описанию класса, какой из методов вызывается автоматически тез написания кода разработчика. Дайте пояснения этой ситуации.

```

class Avtor
{
    protected MyFIO avtor;
    protected string Titul;
    protected string vid;
    protected DateTime dt;
    public Avtor()
    {
        avtor = new MyFIO();
        Titul = "без названия";
        vid = "не определено";
        dt = new DateTime();
    }
    public Avtor(MyFIO fio, string titul, string vid, DateTime Date)
    {
        this.avtor = fio;
        this.Titul = titul;
        this.vid = vid;
        this.dt = Date;
    }
    protected virtual string Result()
    {
        return $"{avtor.fio_pub} {Titul} [Текст] : {vid} / {avtor.iof}. - {dt.Year}";
    }
    public void Show()
    {
        Console.WriteLine(Result());
    }
    ~Avtor()
    {
        Console.WriteLine("объект удалён");
    }
}

```

Ответ:

Из всех методов класса только деструктор (~Avtor()) имеет закрытый доступ и вызывается через другой класс программной платформы - сборщик мусора. Необходим для выполнения завершающих программных действий прежде, чем объект будет уничтожен.

Задание 15. Определите по представленному описанию класса: 1) сколько свойств реализовано в классе; 2) для каких переменных разработаны свойства; 3) какого вида эти свойства?

Ответ:

- 1) реализовано два свойства: `int Dlina_`, `string Stroka_`;
- 2) свойства реализованы для закрытых переменных `int Dlina`, `string Stroka`, соответственно;
- 3) свойство `int Dlina_` - только для чтения, свойство `string Stroka_` - для чтения и для записи.

```
class MyString
{
    protected string Stroka;
    private int Dlina;
    public int Dlina_ {get{return Dlina;}}
    public string Stroka_ {get { return Stroka;} set {if (value != "") Stroka = value;Dlina = Stroka.Length;}}
    public void Print()
    {
        Console.WriteLine(Stroka);
        return;
    }
    public void Print(int x, int y)
    {
        Console.SetCursorPosition(x, y);
        Console.WriteLine(Stroka);
    }
    public MyString()
    {
        Stroka = "нусто";
        Dlina = Stroka.Length;
        return;
    }
    public MyString(string Stroka)
    {
        this.Stroka = Stroka;
        Dlina = Stroka.Length;
        return;
    }
    public static MyString operator -(MyString Object)
    {
        int k = Object.Dlina;
        string Time = Object.Stroka;
        string str = "";
        for (int i = k - 1; i >= 0; i--)
            str += Time[i].ToString();
        MyString ObjectTime = new MyString(str);
        return ObjectTime;
    }
    public static MyString operator +(MyString x, MyString y)
    {
        string sum = x.Stroka + y.Stroka;
        MyString Result = new MyString(sum);
        return Result;
    }
    public static bool operator &(MyString Object, MyString Object2)
    {
        bool a = (Object.Stroka == "");
        bool b = (Object2.Stroka == "");
        return (a & b);
    }
    public static implicit operator string(MyString p)
    {
        string time = "Содержимое текущего объекта - " + p.Stroka.ToUpper();
        return time;
    }
    public static explicit operator byte(MyString p)
    {
        return (byte)p.Stroka.Length;
    }
    ~MyString(){Console.WriteLine("Объект удалён");}
}
```

Задание 16. Определите по представленному описанию класса в задании 5: 1) сколько операторов перегружено в классе; 2) сколько из них унарных операторов, а сколько бинарных; 3) какого вида перегруженные операторы?

Ответ:

- 1) всего перегружено 5 операторов;
- 2) из них 3 оператора унарные (operator -, implicit operator, explicit operator), 2 оператора – бинарные (operator +, operator &);
- 3) два оператора арифметические (operator -, operator +) и возвращают новое состояние класса (новый объект); один оператор логический (operator &), возвращает результат проверки состояния двух объектов; два оператора преобразования (implicit operator, explicit operator) состояния объекта в стандартные типы.

Задание 17. Определите по представленному описанию класса, какие пользовательские типы использует наследник?

Ответ: класс `Прямoug` выступает наследником родительского класса `Figura` и должен реализовать два интерфейса: `IFiguraDvig`, `IFiguraRaschet`.

```
class Прямoug : Figura, IFiguraDvig, IFiguraRaschet
{
    private int Size2;
    public Прямoug() : base()
    {
        Size2 = 0;
    }
    public Прямoug ( string name, ConsoleColor color, int size, int size2):base( name, color, size)
    {
        this.Size2 = size2;
    }
    public int Perimeter()
    {
        return 2*(Size2 +Size);
    }
    public int Square()
    {
        return Size * Size2;
    }
    public override void show()
    {
        Console.ForegroundColor = Color;
        Console.WriteLine($"Я - {Name}! Мой размер - {Size} и {Size2}.");
    }
}
```

Задание 18. Представьте описание класса `Primer`, включив в него два целочисленных закрытых данных `x` и `y` и конструктор-инициализатор для присвоения этим данным переданных значений.

Ответ:

```
class Primer
{
    private byte x, y;
    public Primer (byte x, byte y){this.x=x; this.y=y;}
}
```

Задание 19. Представьте описание класса `Primer`, включив в него две строковые защищённые данные `x` и `y` и конструктор по умолчанию для присвоения этим данным значения «noname».

Ответ:

```
class Primer
{
    protected string x, y;
    public Primer () {this.x=this.y=«noname»;}
}
```

Задание 20. Опишите действия, происходящие в строках 102-110 представленного фрагмента кода с учетом того, что `MyString` – это пользовательских класс, описанный в этом же пространстве имен

Ответ:

102 – объявление объекта класса `MyString` с одновременным его созданием с помощью конструктора по умолчанию.

103 – вызов метода `Print` для созданного объекта.

104 – объявление второго объекта класса `MyString` с одновременным его созданием с помощью конструктора-инициализатора.

105 – вызов метода `Print` для второго созданного объекта.

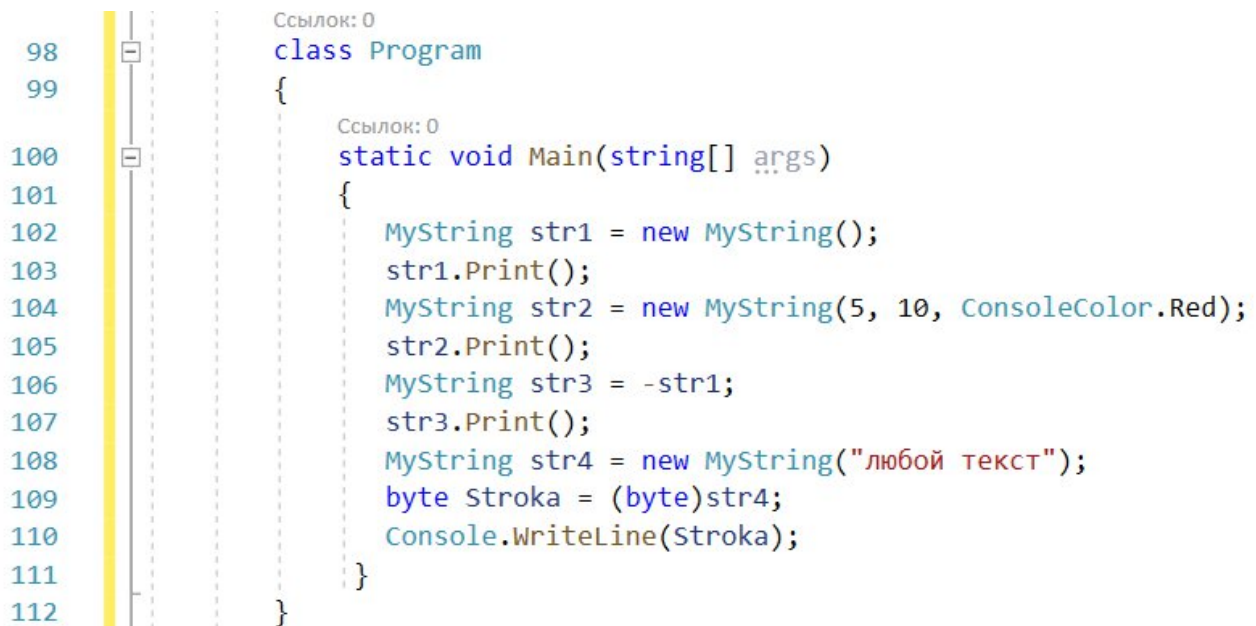
106 – объявление третьего объекта класса MyString и присвоение ему значений первого объекта с помощью операции «унарный минус».

107 – вызов метода Print для третьего созданного объекта.

108 – объявление четвертого объекта класса MyString с одновременным его созданием с помощью перегруженного конструктора-инициализатора.

109 – объявление целочисленной переменной с присвоением ей стартового значения, полученного явным преобразованием четвертого объекта класса MyString к типу byte.

110 – вывод значения целочисленной переменной на экран.



```
Ссылка: 0
class Program
{
    Ссылка: 0
    static void Main(string[] args)
    {
        MyString str1 = new MyString();
        str1.Print();
        MyString str2 = new MyString(5, 10, ConsoleColor.Red);
        str2.Print();
        MyString str3 = -str1;
        str3.Print();
        MyString str4 = new MyString("любой текст");
        byte Stroka = (byte)str4;
        Console.WriteLine(Stroka);
    }
}
```

Задание 21. Опишите подробно отличие действий, происходящие в строках 102, 104, 106, 108 представленного фрагмента кода с учетом того, что MyString – это пользовательских класс, описанный в этом же пространстве имен

Ответ:

102 – объявление объекта класса MyString с одновременным его созданием с помощью конструктора по умолчанию

104 – объявление второго объекта класса MyString с одновременным его созданием с помощью конструктора-инициализатора

106 – объявление третьего объекта класса MyString и присвоение ему значений первого объекта с помощью операции «унарный минус»

108 – объявление четвертого объекта класса MyString с одновременным его созданием с помощью перегруженного конструктора-инициализатора

Задание 22. В каких целях при описании класса используется модификатор доступа static?

Ответ: данный модификатор позволяет использовать объявленный ресурс (переменную или метод) без объявления объекта класса, что в свою очередь позволяет создавать ресурсы класса, которые работают сразу со всеми созданными объектами (например, создать счетчик объектов класса)

Задание 23. В каких целях при описании класса используется служебное слово sealed?

Ответ: для запрета наследования элементов этого класса

Задание 24. Опишите подробно, в каких случаях и для чего можно использовать служебное слово `base`?

Ответ: данный указатель содержит ссылку на родителя, поэтому его можно использовать для вызова конструкторов родителя, одноименных ресурсов (методов, переменных, свойств)

Задание 25. Перечислите требования к описанию конструктора класса.

Ответ: При описании конструктора класса необходимо соблюдать следующие требования:

1. имя конструктора совпадает с именем класса
2. уровень доступа – `public`, чтобы его можно было вызывать за пределами класса автоматически при объявлении объекта
3. возвращаемый результат не указывается, т.к. он однозначно определен – это экземпляр этого же класса
4. может быть перегружен
5. должен быть хотя бы пустой

Задание 26. Перечислите требования к описанию деструктора класса.

Ответ: При описании деструктора класса необходимо соблюдать следующие требования:

1. имя конструктора совпадает с именем класса, перед которым указывается символ “~”
2. закрытый уровень доступа (даже `private` не рекомендуется указывать), так как метод вызывается другим системным классом – сборщиком мусора
3. возвращаемый результат не указывается
4. не перегружается
5. для семейств классов должен быть хотя бы пустой, чтобы не прерывать «цепочку» вызовов

Компетенция ПК-4 - Способен управлять процессами создания информационных систем

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

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.

ЗАДАНИЯ ЗАКРЫТОГО ТИПА

Задание 1. Выделите главную причину, способствующую появлению ООП?

- а) *потребность внести в программы черты человеческого интеллекта;*
- б) *изменение операционной системы и языков программирования;*
- в) *переход на модульную технологию программирования.*

Ответ: а

Задание 2. Выделите преимущество, которое получают конечные пользователи, работающие с программным продуктом, написанным по методологии ООП?

- а) *графический интерфейс программного продукта;*
- б) *понятную логику работы программного продукта;*

в) *высокую эффективность (скорость) работы программного продукта.*

Ответ: б

Задание 3. Выделите те программные преимущества, которые обусловлены инкапсуляцией:

- а) *использование одинаковых идентификаторов для разных реализаций, имитация механизма принятия решения в программе;*
- б) *надежность работы программы, сокрытие части кода, непосредственная связь данных и методов для их обработки;*
- в) *сокращение кода в программных конструкциях, унификация поведения объектов, использование «стороннего» кода для собственных разработок;*

Ответ: б

Задание 4. Выделите те программные преимущества, которые обусловлены наследованием:

- а) *использование одинаковых идентификаторов для разных реализаций, имитация механизма принятия решения в программе;*
- б) *надежность работы программы, сокрытие части кода, непосредственная связь данных и методов для их обработки;*
- в) *сокращение кода в программных конструкциях, унификация поведения объектов, использование «стороннего» кода для собственных разработок;*

Ответ: в

Задание 5. Выделите те программные преимущества, которые обусловлены полиморфизмом:

- а) *использование одинаковых идентификаторов для разных реализаций, имитация механизма принятия решения в программе;*
- б) *надежность работы программы, сокрытие части кода, непосредственная связь данных и методов для их обработки;*
- в) *сокращение кода в программных конструкциях, унификация поведения объектов, использование «стороннего» кода для собственных разработок;*

Ответ: а

Задание 6. Выделите правильную последовательность действий, выполняемую для проектирования семейства классов (компонента):

- а) *выявить классы в предметной области; определить зависимость между классами; определить интерфейсы классов; определить набор операций для каждого класса;*
- б) *выявить классы в предметной области; определить интерфейсы классов; определить набор операций для каждого класса; определить зависимость между классами;*
- в) *выявить классы в предметной области; определить интерфейсы классов; определить зависимость между классами; определить набор операций для каждого класса;*
- г) *выявить классы в предметной области; определить набор операций для каждого класса; определить зависимость между классами; определить интерфейсы классов;*

Ответ: г

Задание 7. Выделите рекомендуемые правила для определения классов в предметной области:

- а) *выделить наиболее значимые понятия предметной области, сопутствующие им прилагательные и глаголы;*

- б) *определите набор операций, который необходим для каждого понятия из предметной области;*
- в) *проведите беседу со специалистами из предметной области, обращая внимание на используемые ими понятийный аппарат;*
- г) *рассмотрите, каким образом будет создаваться, копироваться и уничтожаться потенциальный объект;*

Ответ: а, в

Задание 8. Выделите рекомендуемые правила для определения классов в предметной области:

- а) *выделить наиболее значимые понятия предметной области, сопутствующие им прилагательные и глаголы;*
- б) *определите набор операций, который необходим для каждого понятия из предметной области;*
- в) *проведите беседу со специалистами из предметной области, обращая внимание на используемые ими понятийный аппарат;*
- г) *рассмотрите, каким образом будет создаваться, копироваться и уничтожаться потенциальный объект;*

Ответ: б, г

Задание 9. В каких случаях следует пересмотреть иерархию классов и создать общий родительский класс?

- а) *есть несколько наборов операций одного класса с различными способами использования, и эти наборы обеспечивают доступ к отдельным подмножествам объектов реализации*
- б) *один и тот же класс возникает в процессе обсуждения несвязанных тем;*
- в) *разные классы имеют схожие способы использования, сходство наборов операций, сходство реализаций и просто тот факт, что часто в процессе обсуждения проекта оба класса появляются одновременно;*
- г) *разные классы часто в процессе обсуждения проекта появляются одновременно;*

Ответ: в, г

Задание 10. В каких случаях следует пересмотреть иерархию классов и разделить класс(ы) на части?

- а) *несколько наборов операций одного класса с различными способами использования, и эти наборы обеспечивают доступ к отдельным подмножествам объектов реализации*
- б) *один и тот же класс возникает в процессе обсуждения несвязанных тем;*
- в) *разные классы имеют схожие способы использования, сходство наборов операций, сходство реализаций и просто тот факт, что часто в процессе обсуждения проекта оба класса появляются одновременно;*
- г) *разные классы часто в процессе обсуждения проекта появляются одновременно;*

Ответ: а, б

ЗАДАНИЯ ОТКРЫТОГО ТИПА

Задание 11. В чем смысл этапа «сопровождение» для такого программного проекта как семейство классов?

Ответ:

Традиционно этап «сопровождения» входит в обычный цикл развития программного обеспечения. Для семейства классов это тоже имеет место быть, предполагая действия перепроектирования семейства и/или улучшения реализации классов. Если при проектировании семейства учтены вопросы расширяемости, гибкости и переносимости, то задача сопровождения решается естественным образом.

Задание 12. Каким способом можно представить работоспособность семейства классов для Заказчика?

Ответ: Разработать демо-программу (прототип программы).

Задание 13. В чем суть этапа «тестирования» для такого программного проекта как семейство классов?

Ответ:

Традиционно этап «тестирования» входит в обычный цикл развития программного обеспечения. Тестирование может проходить в виде первичного тестирования на этапах отладки, а может проходить в виде комплексного тестирования, когда создается некоторая демо-программа (условный «сюжет»), в которой для реализации условно-поставленных задач из предметной области используются тестируемые классы. Важно, в ходе работы программы проверить качество работы всех элементов класса (конструкторов и деструкторов, доступы к переменным и методам, работу свойств и методов, перегрузку операторов и т.д.), а также связи между классами.

Задание 14. Опишите понятие «эффективность» для ООП.

Ответ: Понятие «эффективность» можно рассмотреть с нескольких сторон: 1) с точки зрения разработчика семейства классов – это очень долгий, итеративный процесс, требующий участия большого авторского коллектива; 2) зато разработка программных продуктов на основе семейства классов идет намного быстрее, т.к. программа «собирается» из уже работающих фрагментов; 3) в тоже время программные продукты, построенные на основе ООП, требуют больших ресурсов памяти и времени, т.к. поздний полиморфизм требует работы дополнительных системных механизмов, кроме того в процессе трансляции объектно-ориентированных программ в исполняемый код центрального процессора возникает ряд «неоптимальностей» по использованию памяти и вычислительного времени процессорных ядер.

Задание 15. Сопоставьте способы взаимодействия с Заказчиком и их потенциальные преимущества.

Способы взаимодействия: 1) телефонный разговор; 2) видеоконференцсвязь; 3) переписка по почте.

Потенциальные преимущества: а) скорость отклика и получения ответов на вопросы; б) архивация сообщений в письменном формате, обмен документами; в) максимально понятное общение и минимум причин для разногласий.

Ответ: 1 – а; 2 – в; 3 – б.

Задание 16. Сопоставьте классификацию методов класса с описанием.

Классификация: 1) базовые методы; 2) селекторы; 3) модификаторы; 4) повторители.

Описание: а) методы, изменяющие состояние объекта; б) конструкторы, деструкторы, операции копирования; в) методы, не изменяющие состояние объекта; г) методы, открывающие доступ к объектам.

Ответ: 1 - б; 2 - в; 3 - а; 4 - г.

Задание 17. Оцените «пользу» от позднего полиморфизма и виртуальных функций и дайте рекомендации по их количеству в классах.

Ответ: Поздний полиморфизм позволяет реализовать механизм принятия решения в уже работающей программе (имитация человеческого интеллекта), опираясь на виртуальные и переопределяемые методы. Однако, обеспечение позднего полиморфизма требует особого подхода к компиляции программного продукта, использованию особых функций операционной системы и постоянных переходов в памяти и особого распределения самой памяти. Чем больше виртуальных функций, тем «тяжелее» работает программный продукт. Большое количество виртуальных функций может свести на нет все преимущества объектной технологии. Кроме того, объявив функцию виртуальной, мы существенно повлияем на использование ее в классе и на взаимоотношения этого класса с другими классами, что требует тщательного проектирования иерархии классов. Проектирование осложняет то, что класс с хотя бы одной виртуальной функцией по сути выступает в роли шаблона по отношению к классам, которые «еще могут быть определены».

Задание 18. Опишите стратегию минимализма (по Бьерну Страуструпу) на этапе определения методов класса.

Ответ:

По Бьерну Страуструпу (автору ООП) возможно несколько подходов к процессу определения набора методов для класса (семейства), но можно выделить минимальную последовательность шагов: 1) определить, каким образом объект класса будет создаваться, копироваться (если нужно) и уничтожаться; 2) определить минимальный набор прикладных методов, отвечающих за «работу» того понятия предметной области, которому соответствует класс; 3) определить методы, которые могут быть добавлены для удобства записи; 4) определить, какие методы должны быть виртуальными, т.е. такими, для которых класс выступает в роли шаблона для реализации производного класса. Необходимо отметить, чем больше функций, тем больше вероятность, что они наложат определенные ограничения на реализацию и затруднят эволюцию системы. К тому же добавление функций добавляет работы не только программисту, но и требует пересмотра проекторочных решений.

Задание 19. Заказчик не может приехать лично на демонстрацию работы программного продукта. Какими способами можете продемонстрировать работу программы удаленно?

Ответ:

Существует несколько способов демонстрации работающего программного продукта Заказчику. Продемонстрировать свой экран с работающим программным обеспечением через любую платформу видеоконференцсвязи. Записать видеоролик и отправить его Заказчику, либо продемонстрировать через любую платформу видеоконференцсвязи. Отправить ссылку на облачный ресурс, в рамках которого можно запустить программный продукт на работу (если такая возможность принципиально существует). Первый способ наиболее оптимальный, т.к. работа программного продукта происходит на оборудовании разработчика с отлаженным под данную архитектуру кодом. При этом демонстрацию также осуществляет сам разработчик, что

гарантирует безошибочную демонстрацию. Если же демонстрация будет проходить в гибридном формате, то необходимо обеспечить хорошее качество звука для взаимодействия со всеми участниками.

Задание 20. Какие правила необходимо соблюдать при удаленном взаимодействии с участниками проектной группы и/или представителями Заказчика для создания рабочей и доверительной обстановки.

Ответ: Визуальный контакт, корректное имя (без сокращений и упрощений), хороший звук.

Задание 21. Какие межличностные барьеры могут снижать эффективность взаимодействия?

Ответ:

Межличностным барьерам относятся барьеры восприятия, семантические, невербальные, плохая обратная связь и неумение слушать. Восприятие информации человеком зависит от большого количества факторов. Человек, используя зрительные, звуковые, осязательные каналы получения информации, воспринимает не всю поступающую к нему информацию, а только ту, которая имеет для него значение. Именно поэтому необходимо донести значимость сообщения при его составлении. Семантические проблемы связаны с неоднозначным пониманием переданного сообщения. Использование сленга еще более усугубляет данную проблему. Попыткой ее решения в проекте является создание и использование глоссария терминов. Однако в процессе коммуникации между Заказчиком и Исполнителем взаимодействуют не только различные люди, но и сотрудники различных организаций. А каждая организация имеет свою корпоративную культуру. Корпоративная культура это некий регламент (гласный или негласный) поведения сотрудника в организации, который может включать в себя и особенности коммуникаций. Таким образом, у участника коммуникации могут возникнуть те черты, которые свойственны его организации. Эти черты могут наложить свой отпечаток на процесс коммуникации и их необходимо учитывать. Дополнительно, к межличностным барьерам коммуникации необходимо отнести и проблему психологической «настройки» специалистов на общение. Важный момент общения Исполнителя и Заказчика, как человека настроят на эту компанию: положительно или отрицательно. В идеале, конечно, необходимо обеспечивать нейтральный настрой.

Задание 22. Какие организационные барьеры могут снижать эффективность взаимодействия?

Ответ:

К организационным барьерам относят проблемы построения коммуникационных сетей и обработки информации. Задача организации коммуникационных сетей состоит в том, чтобы используемый вариант их построения отвечал конкретным условиям проекта и характеру решаемых им проблем. Согласно результатам исследования известно, что сеть типа «колесо» используются, если проблема, которую надо решить, проста. Сеть типа «каждый с каждым» стимулирует создание хорошего морального климата и приспособляемости, а так же обмен мнениями в рабочей группе. Сеть типа "круг" более эффективна при выполнении заданий, требующих адаптации и творческого подхода. Правильный выбор коммуникационной сети для проекта позволит обеспечить необходимую скорость передачи сообщений, оптимальную нагрузку на всех участников проекта, достигнуть минимального искажения сообщений в процессе передачи информации.

Задание 23. Каким способом можно представить концептуальное решение по разрабатываемой библиотеке классов удаленному Заказчику (технологические средства)?

Ответ: Для представления концептуального решения по разрабатываемой библиотеке классов Заказчику на расстоянии необходимо подготовить качественный визуальный материал (схемы, модели) и воспользоваться средствами удаленного взаимодействия, которые позволяют проводить видеоконференцсвязь и обеспечивают хороший уровень видеоизображения и звука: Yandex.Телемост, Zoom, Scurе. Первый инструмент предпочтительнее, т.к. он отечественного производства.

Задание 24. Каким способом лучше всего представить возможности разработанной библиотеки классов потенциальным пользователям?

Ответ: Для представления возможностей разработанной библиотеки классов потенциальным пользователям необходимо представить: 1) перечень предметных понятий, которые легли в основу библиотеки (элементы абстракции); 2) модель классов с описанием назначения каждого класса и отношений между классами; 3), идею наследования и идею полиморфизма; 4) разработанные интерфейсы (если они есть); 5) показать работу демо-программы, демонстрирующую реализацию функциональных (прикладных) возможностей классов.

Задание 25. Кто выступает потенциальным пользователем библиотеки классов?

Ответ: разработчики прикладных программных продуктов.

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)