УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»

Кафедра информационных технологий

Утверждаю
Зав. кафедрой
О.В.Федорова

Протокол заседания
кафедры № 10
от 06.04.2026

Рабочая программа дисциплины

Наименование дисциплины	Объектно-ориентированное программирование
Направление подготовки	09.03.04 Программная инженерия
Профиль подготовки	Информационные технологии
Год набора	2026

Составитель:

ст. преподаватель Потапова Е. А.

Казань

Содержание

1. Цели и задачи учебной дисциплины	3
2. Место дисциплины в структуре ОПОП	4
3. Требования к результатам освоения дисциплины	4
4. Структура и содержание дисциплины	5
4.1. Модульно-тематический план и пояснительная записка с указанием этапов формирования компетенций	5
4.2. Содержание дисциплины по темам (разделам)	9
4.3. Планы практических и семинарских занятий	11
4.4. Планы практической подготовки/лабораторных занятий	11
5.Учебно-методическое обеспечение самостоятельной работы студентов	14
6. Учебно-методическое и информационное обеспечение дисциплины	16
7. Материально-техническое обеспечение дисциплины	18
8. Оценка компетенций по изучаемой дисциплине	18
Приложение 1. Методические указания для обучающихся по освоению дисциплины	21
Приложение 2. Фонд оценочных средств для проведения текущей и промежуточной аттестации по дисциплине	22

1. **Цели и задачи учебной дисциплины**

Дисциплина является одной из базовых при подготовке бакалавров по направлению «Программная инженерия».

Цель дисциплины - сформировать у будущего бакалавра по направлению 09.03.04 «Программная инженерия» комплекс знаний, умений и навыков (компетенций) в сфере информационных и «сквозных технологий», которые позволяют ему применять основные концепции и принципы объектно-ориентированного подхода к разработке программ. Существенными особенностями дисциплины являются:

- использование межъязыкового подхода с упором на рассмотрение общих фундаментальных принципов с примерами их реализации в основных объектных языках (Java, C#, C++, Object Pascal)
- практическая направленность, связанная с написанием большого количества программ на практических занятиях с обязательным тестированием и отладкой программ на компьютере.

Задачи дисциплины:

Изучив курс, студенты должны иметь представление:

- о базовых возможностях современных объектных языков, их использовании и направлениях развития
- об основных задачах, решаемых с использованием объектного подхода и современных инструментальных средств

Знать: основные объектно-ориентированные языки программирования и современные инструменты быстрой разработки приложений.

(- Знать инструменты разработки приложений: MS Visual Studio, Netbeans IDE, Delphi 10.1 Berlin

- Знать операторы и управляющие конструкции Greenfoot, средства объектно-ориентированного программирования, используемые в данной среде.)

Уметь: применять объектные языки программирования и инструменты быстрой разработки для решения прикладных задач

(- Уметь применять MS Visual Studio, Netbeans, Delphi 10.1 Berlin для решения прикладных задач

- Уметь составлять алгоритмы решения задач с использованием технологии в визуальной среде Greenfoot)

Владеть: навыками программирования, отладки и тестирования объектных программ

(- Владеть навыками программирования, отладки и тестирование объектных программ с помощью инструментов разработки: MS Visual Studio, Netbeans, Delphi 10.1 Berlin

- Владеть опытом использования визуальной учебной среды Greenfoot на этапах разработки программ)

2. Место дисциплины в структуре ОПОП

Данная дисциплина относится к обязательной части блока 1 учебного плана и находится во взаимосвязи с дисциплинами согласно схеме:

Обеспечивающие учебные дисциплины

Методы программирования

Объектно-ориентированное программирование

Обеспечиваемые учебные дисциплины

Введение в системы баз данных

Инструменты и методы разработки систем

До начала изучения дисциплины у студента должны быть сформированы компоненты компетенций, полученные в результате изучения дисциплины Методы программирования.

3. Требования к результатам освоения содержания дисциплины

Дисциплина «Объектно-ориентированное программирование» участвует в формировании следующих компетенций в соответствии с ФГОС ВО по направлению «Программная инженерия»:

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

После освоения дисциплины студент должен получить следующие образовательные результаты**,** соотнесенные с индикаторами достижения компетенций

Декомпозиция компетенций

Индикаторы	Результаты обучения по дисциплине
ОПК-6.1 Применяет языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных	**ОПК-6.1. З.3.** Знает основные объектно-ориентированные языки программирования и современные инструменты быстрой разработки приложений **ОПК-6.1. У.3.** Умеет применять объектные языки программирования и инструменты быстрой разработки для решения прикладных задач

ОПК-6.2 Программирует, отлаживает и тестирует программные продукты	**ОПК-6.2. В.3.** Имеет навыки программирования, отладки и тестирование объектных программ

4. Структура и содержание дисциплины

4.1. Модульно-тематический план и пояснительная записка с указанием этапов формирования компетенций

Общая трудоемкость дисциплины составляет 7 зачетных единиц (252 академических часа)

	Модульная разбивка учебной дисциплины					
	Направление «Программная инженерия» Дисциплина « Объектно-ориентированное программирование»					
Наименование модулей	Количество ауд. часов			Самостоятельная работа Очное	Всего часов	Индикаторы компетенции
	Лекции Очное	Практики Очное	Практическая подготовка/ Лабораторные занятия Очное			
<u>Модуль 1</u>: «Объекты и классы»						
<u>Тема 1:</u> История развития объектной технологии. Понятие об объектах. Абстрагирование и инкапсуляция	1	-	-	4	5	ОПК-6.1
<u>Тема 2</u>: Классы: назначение и правила описания.	1	-	2	4	7	
<u>Тема 3:</u> Объектные переменные	1	-	2	6	9	
<u>Тема 4:</u> Программирование экземпляров объектов в визуальной среде Greenfoot	1	-	2	6	9	ОПК-6.1 ОПК-6.2
<u>Тема 5:</u> Реализация простейшего контейнерного класса	1	-	4	10	15	
<u>Модуль 2</u>: «Взаимодействие объектов и классов»						
<u>Тема 1.</u> Взаимодействие на основе механизма композиции/агрегации	1	-		6	7	ОПК-6.1 ОПК-6.2

Тема 2. Примеры разработки классов на основе композиции	2	-	4	8	14	
Тема 3. Реализация контейнерного класса на основе адресного списка	2	-	4	8	14	
Тема 4. Взаимодействие классов на основе механизма наследования	2	-	-	8	10	
Тема 5. Примеры разработки иерархий классов	2	-	4	8	14	
	Модуль 3: «Полиморфизм, его проявления и использование»					
Тема 1: Полиморфизм на уровне методов	2	-	2	6	10	ОПК-6.1 ОПК-6.2
Тема 2: Примеры иерархий классов с виртуальными методами	2	-	4	8	14	
Тема 3. Полиморфизм на уровне объектных переменных	2	-	4	8	14	
Тема 4. Разработка полиморфных контейнеров	2	-	4	6	12	
	Модуль 4: «Дополнительные возможности объектных языков»					
Тема 1: Интерфейсные классы	2	-	2	6	10	ОПК-6.1 ОПК-6.2
Тема 2: Обобщенные классы	2	-	2	6	10	
Тема 3: Исключения и их объектная обработка	2	-		10	12	
Тема 4: Компонентные классы и модели	2	-		10	12	
Подготовка к зачету				18	18	
Подготовка к экзамену				36	36	
Всего	**30**		**40**	**182**	**252**	

Пояснительная записка с этапами формирования компетенций

Данный модульный курс состоит из 4 модулей, порядок освоения которых выстраивает траекторию и этапы формирования заявленных компетенций или их составляющих.

Модуль 1 «Объекты и классы» включает в себя пять учебных тем.

В результате прохождения первого модуля студент должен:

- знать назначение и особенности объектного подхода, знать понятия «объект», «класс», «метод», понимать сущность переменных объектного типа
- уметь разрабатывать простейшие классы объектов и использовать их на практическом уровне
- уметь программировать экземпляры объектов в визуальной среде Greenfoot
- иметь навык разработки простейших объектных программ

Уровень освоения полученных знаний проверяется выполнением практических заданий, прохождением тестирования, реализацией первого этапа курсовой работы и ответами на теоретические вопросы.

Модуль 2 «Взаимодействие объектов» включает в себя пять учебных тем.

В результате прохождения второго модуля студент должен:

- знать такие понятия как композиционное взаимодействие объектов, механизм наследования классов, иерархия классов
- уметь разрабатывать классы с использованием двух базовых способов взаимодействия
- иметь навыки разработки объектных программ на основе изучаемых способов взаимодействия

Уровень освоения полученных знаний и умений проверяется выполнением практических заданий, прохождением тестирования, результатом выполнения этапов 1-3 курсовой работы и ответами на теоретические вопросы.

Модуль 3 «Полиморфизм» включает в себя 4 учебные темы.

В результате прохождения третьего модуля студент должен:

- знать такие понятия как переопределение методов, виртуальные методы и таблицы ВМ, динамическое связывание, полиморфизм объектных переменных, метаданные, полиморфные контейнеры/коллекции
- уметь разрабатывать классы с виртуальными методами и классы полиморфных контейнеров
- иметь навыки разработки объектных программ с использованием полиморфных механизмов

Уровень освоения полученных знаний проверяется выполнением практических заданий, прохождением тестирования и ответами на теоретические вопросы.

Модуль 4 «Дополнительные возможности» включает в себя 4 учебных темы.

В результате прохождения модуля студент должен:

- знать такие понятия как интерфейсные и обобщенные классы, компонентные классы, объекты-исключения
- уметь разрабатывать классы указанных типов и выполнять обработку исключений
- иметь навыки разработки программ с использованием указанных понятий

Уровень освоения полученных знаний проверяется выполнением практических заданий, прохождением тестирования и ответами на теоретические вопросы.

Данное деление дисциплины на модули активизирует самостоятельную работу студентов, повышает интенсивность и системность учебной работы, регулирует контроль учебной деятельности студентов в течении семестров, усиливает мотивацию студентов к изучению учебного материала.

4.2. **Содержание дисциплины по темам (разделам)**

Тема 1. История развития объектной технологии. Понятие об объектах. Абстрагирование и инкапсуляция

Основные этапы развития объектной технологии. Язык C++ и его реализации. Пакеты Borland Delphi/C++ Builder и библиотека классов VCL. Язык Java, платформа Java 2, базовая библиотека классов JFC. Платформа MS .NET, библиотека Framework Class Library и язык C#. Современное состояние рынка объектных языков и инструментов разработки.

Понятие об объектах как способе моделирования. Принцип абстрагирования. Принцип инкапсуляции. Свойства объектов, их назначение и классификация. Методы и их классификация: конструкторы, методы доступа к свойствам, специализированные методы.

Тема 2. Классы: назначение и правила описания.

Классы как способ формализованного описания однотипных объектов. Общая структура класса. Открытые и закрытые элементы классов. Абстрактные классы. Перегрузка методов. Классовые свойства и методы. Примеры и особенности описания классов для различных языков. Тема 3. Классы: назначение и правила описания

Тема 3. Объектные переменные

Способы объявления объектных переменных, объектные переменные как неявные указатели на объекты. Создание объектов с помощью конструкторов.

Использование объектов через вызовы их методов. Правила использования объектных переменных, возможные ошибки.

Тема 4. Программирование экземпляров объектов в визуальной среде Greenfoot

Структура программы в среде Greenfoot. Создание нового сценария. Панель классов и объектов. Редактор сцен. Создание экземпляров класса. Программирование экземпляров объектов. Greenfoot API: основные классы и методы. Типы данных. Объявление и инициализация переменных. Типы методов. Методы с параметрами. Управление объектом. Работа с изображениями объектов.

Тема 5. Реализация простейшего контейнерного класса

Постановка задачи на разработку контейнера и основные этапы решения. Проектирование контейнера. Формализованное описание в виде класса. Рекомендации по реализации основных методов. Пример использования контейнерного класса.

Тема 6. Взаимодействие на основе механизма композиции/агрегации

Общая характеристика основных способов взаимодействия объектов. Понятие об агрегации и композиции как способах моделирования отношения «часть – целое». Примеры использования при построении объектных моделей. Программная реализация композиции с помощью объектных свойств. Правила реализации конструкторов составных классов. Использование методов агрегированных классов.

Тема 7. Примеры разработки классов на основе композиции

Основные этапы реализации композиции, преимущества использования. Использование композиции при разработке классов информационных объектов. Примеры использования разработанных классов.

Тема 8. Реализация контейнерного класса на основе динамического списка

Постановка задачи и особенности реализации контейнера на основе динамического списка. Два способа реализации спискового контейнера. Встраивание связующих полей в информационные объекты. Проектирование и реализация класса объектов-посредников. Проектирование и реализация класса для спискового контейнера. Пример использования разработанных классов.

Тема 9. Механизм наследования

Обобщение и наследование как способы моделирования отношения «Общее-Частное». Родительские и дочерние классы, общая структура дочернего класса. Иерархия классов. Простое и множественное наследование. Дерево классов как основа современных библиотек классов. Корневой класс Object. Преимущества использования иерархий классов при создании сложных программных систем. Защищенные элементы класса. Обращение к конструктору родительского класса. Примеры и особенности использования

механизма наследования в разных языках.

Тема 10. Примеры разработки небольших иерархий классов

Постановка задачи и преимущества использования механизма наследования. Проектирование и реализация иерархии классов для графических объектов. Примеры разработки иерархии классов для простых информационных задач.

Тема 11. Полиморфизм на уровне методов, механизм переопределения

Понятие о полиморфизме, различные проявления полиморфизма. Переопределение методов, отличие от перегрузки. Важность механизма переопределения как основы разработки динамически настраиваемых программ. Статическая и динамическая компоновка. Виртуальные методы. Таблицы виртуальных методов. Внутренние связи между объектами и таблицами. Особенности объявления и использования виртуальных методов в разных языках.

Тема 12. Примеры иерархий классов с виртуальными методами

Постановка задачи, возможность улучшения ранее созданной иерархии. Рекомендации по изменению структуры корневого и дочерних классов для иерархии графических и информационных объектов. Примеры описания классов с виртуальными методами для разных языков. Реализация внутренних связей между объектами и их таблицами как основа динамической настройки методов.

Тема 13. Полиморфизм на уровне объектных переменных

Полиморфность объектных переменных как способность манипулирования объектами разных классов. Ограничения. Объединение полиморфных переменных в единую структуру с возможностью циклической обработки. Использование полиморфных переменных в качестве параметров методов. Проблемы использования полиморфных переменных. Понятие о метаданных. Доступ к метаданным во время выполнения программы. Проверка динамического типа полиморфных переменных. Приведение типов. Понятие о механизме отражения (рефлексии).

Тема 14. Примеры разработки полиморфных контейнеров

Рекомендации по разработке универсальных контейнеров. Использование механизмов переопределения методов и полиморфных объектных переменных. Варианты реализации универсальных контейнеров на основе массива и динамического списка. Новые возможности универсальных контейнеров и проблемы их использования. Стандартные контейнерные классы в основных библиотеках классов..

Тема 15. Интерфейсные классы

Понятие об интерфейсных классах и причины их возникновения. Отличия интерфейсных классов от абстрактных и реальных классов. Множественность наследования на уровне интерфейсных классов. Применение интерфейсов при конструировании реальных классов. Примеры использования интерфейсных классов. Правила объявления и использования интерфейсных классов в разных

языках. Использование интерфейсов в стандартных библиотеках классов.

Тема 16. Обобщенные классы

Понятие об обобщенных (generic) классах. Параметризация классов на уровне объявления свойств и параметров методов. Правила описания обобщенных классов в различных языках. Использование обобщенных классов с конкретизацией типов. Шаблоны языка C++. Правила описания и использования шаблонных функций и классов. Понятие о библиотеке стандартных шаблонов STL.

Тема 17. Исключения и их объектная обработка

Понятие об исключительных или особых ситуациях при выполнении программы. Важность обработки исключений. Основные типы исключений. Объекты-исключения, их создание и использование. Выделение потенциально опасных фрагментов кода. Обработчики исключений. Программная генерация исключений. Оформление блока завершения.

Тема 18. Компонентные классы и модели

Понятие о компонентных объектах. Компонентные классы как основа современных технологий быстрой разработки приложений. Некоторые особенности компонентной модели пакета Delphi.

Назначение и особенности компонентной модели Java Beans. Правила описания компонентных классов. Особенности обработки событий на уровне объектов.

Назначение и особенности компонентной модели платформы .NET. Правила описания свойств. Обработка событий с помощью объектов-делегатов.

4.3. **Планы практических занятий**

Не предусмотрено учебным планом

4.4. **Планы практической подготовки/лабораторных занятий**

Тема 1: Разработка и использование простейших классов (на примере класса объектов «Студент»)

Порядок выполнения работы:

1. Изучить постановку задачи и выполнить неформальное проектирование класса с определением необходимых свойств и методов
2. Дать формальное описание класса с реализацией всех методов
3. Разработать тестовую программу консольного типа для проверки работоспособности методов
4. Создать оконное приложение с удобным и надежным интерфейсом для ввода исходных данных и отображения текущего состояния объекта
5. Проверить работу созданного приложения для различных ситуаций

Тема 2: Разработка и использование простейших классов в соответствии с индивидуальным заданием (ИЗ)

Порядок выполнения работы:

1. Изучить постановку задачи и выполнить неформальное проектирование класса с определением необходимых свойств и методов

2. Дать формальное описание класса с реализацией всех методов

3. Разработать тестовую программу консольного типа для проверки работоспособности методов

4. Создать оконное приложение с удобным и надежным интерфейсом для ввода исходных данных и отображения текущего состояния объекта

5. Проверить работу созданного приложения для различных ситуаций

Тема 3. Программирование экземпляров объектов. Greenfoot API: основные классы и методы

Создание собственных объектов и классов в визуальной программной среде Greenfoot и управление данными объектами с помощью мыши.

Тема 4: Реализация простейшего контейнера для информационных объектов (в соответствии с ИЗ)

Порядок выполнения работы:

1. Создать описание контейнерного класса на основе обычного массива для хранения и обработки информационных объектов

2. Реализовать методы контейнера: добавление, удаление, поиск, отображение

3. Разработать тестовую программу консольного типа для создания контейнера и проверки всех его методов

4. Создать оконное приложение для демонстрации возможностей объекта-контейнера и проверить его работу для разных ситуаций

5. Изменить реализацию контейнерного класса с заменой обычного массива динамическим

6. Проверить работу модифицированного контейнера на консольном и оконном приложениях

Тема 5: Реализация контейнера для информационных объектов на основе динамического списка

Порядок выполнения работы:

Первая версия:

1. Внести в класс информационных объектов изменения, необходимые для связывания этих объектов в адресную структуру

2. Разработать класс-контейнер для управления набором информационных объектов на основе адресного списка

3. Выполнить реализацию всех необходимых методов

4. Разработать тестовую программу консольного типа для создания контейнера и проверки всех его методов

5. Создать оконное приложение для демонстрации возможностей объекта-контейнера и проверить его работу для разных ситуаций

Вторая версия:

1. Ввести вспомогательный класс объектов-посредников для доступа к информационным объектам и связывания их в единый набор
2. Внести необходимые изменения в структуру контейнерного класса и реализацию его методов
3. Разработать тестовую программу консольного типа для создания спискового контейнера и проверки всех его методов
4. Создать оконное приложение для демонстрации возможностей объекта-контейнера и проверить его работу для разных ситуаций

Тема 6: Создание и использование библиотеки классов для графических фигур на основе наследования

Порядок выполнения работы:

1. Создать базовый абстрактный класс иерархии графических фигур с минимально необходимыми свойствами, методами доступа и абстрактными методами отображения и перемещения
2. Создать класс окружностей как дочерний по отношению к классу фигур с добавлением необходимых свойств и методов
3. На основе класса окружностей создать дочерний класс объектов-эллипсов с добавлением необходимых свойств и методов
4. Создать класс прямоугольников как дочерний по отношению к классу фигур с добавлением необходимых свойств и методов
5. Внести необходимые изменения в демо-программу и проверить правильность ее работы

Тема 7: Разработка небольшой иерархии классов для информационных объектов в соответствии с индивидуальным заданием (ИЗ)

Порядок выполнения работы:

1. Изучить постановку задачи в соответствии с ИЗ и выполнить анализ объектов этой задачи с точки зрения возможности использования принципа наследования
1. Разработать родительский абстрактный класс с общими данными, методами доступа и одним абстрактным методом
2. Разработать два дочерних класса со своими дополнительными данными и методами доступа и своей реализацией родительского абстрактного метода
3. Разработать тестовую консольную (оконную) программу для проверки работоспособности переопределенных методов

Тема 8: Реализация полиморфного контейнера-массива в соответствии с ИЗ.

Порядок выполнения работы:

1. Спроектировать класс-контейнер, позволяющий организовать с помощью массива хранение и обработку информационных объектов из созданной ранее иерархии

2. Выполнить программную реализацию основных методов контейнера (добавление, удаление, поиск, вывод данных)

3. Разработать демонстрационную программу, в которой реализовать проверку всех методов контейнера

Тема 9: Реализация полиморфного контейнера на основе адресного списка в соответствии с ИЗ

Порядок выполнения работы:

1. Спроектировать класс-контейнер, позволяющий организовать хранение и обработку любых информационных объектов из созданной ранее иерархии (без использования объектов-посредников)

2. Выполнить программную реализацию основных методов контейнера (добавление, удаление, поиск, вывод данных)

3. Разработать демонстрационную программу, в которой реализовать проверку всех методов контейнера

4. Модифицировать контейнер с использованием объектов-посредников, разработать класс посредников и реализовать все необходимые методы

5. Модифицировать демонстрационную программу, в которой реализовать проверку всех методов нового контейнера

Тема 10: Использование интерфейсных классов

Порядок выполнения работы:

1. Изучить задачу «Студенты и сотрудники учебного заведения» с точки зрения появления ситуации множественного наследования и ее отработки с использованием интерфейсов

2. Разработать структуру необходимых интерфейсных классов

3. Разработать описание реальных классов, построенных на основе созданных интерфейсов

4. Разработать демонстрационную программу и проверить ее работу

Тема 11: Обобщенные (параметризованные) классы

Порядок выполнения работы:

1. Изучить постановку задачи на разработку обобщенного класса для реализации стека-массива объектов любых типов

2. Выполнить реализацию необходимых методов с использованием обобщенного типа как параметра класса

3. Разработать тестовую демонстрационную программу, в которой созданный обобщенный класс используется для хранения и обработки объектов разных типов

5. Учебно-методическое обеспечение самостоятельной работы

студентов

Самостоятельная работа является важным элементом освоения основной программы курса. В соответствии со спецификой предмета, самостоятельная работа включает в себя:

- самостоятельное изучение материала по заданным преподавателем темам, например:
  - Основные классы в визуальной среде Greenfoot
  - Основы технологии программирования кроссплатформенных приложений на Java
  - Экспорт Java в среду Greenfoot
  - И др.
- Практическая подготовка по темам;
- выполнение курсового проекта
- работу с литературой и учебно-методическими пособиями

Самостоятельная практическая подготовка выполняется студентом в свободное от занятий время и включает в себя:

- проработку лекционного материала по указанной теме;
- подготовку алгоритмов решения задач по теме;
- написание программ в соответствии с составленным алгоритмом;
- проверку работоспособности программ, как в домашних условиях, так и в компьютерных классах института;
- предоставление результатов самостоятельной работы преподавателю.

В качестве основного направления для самостоятельного изучения предлагается освоение одного из современных объектно-ориентированных языков (Java, C#, C++) и соответствующих инструментов разработки (MS Visual Studio, Java NetBeans, Eclipse, Intelli Java IDEA и.т.д.). Рекомендуемая литература приводится в следующем разделе.

Выполнение курсового проекта

Цель работы: получение навыков разработки объектных программ, включая создание набора собственных взаимосвязанных классов для объектной реализации специализированного контейнера. Контейнер предназначен для хранения и обработки данных некоторой информационной задачи. Контейнер представляет собой одно- или двухуровневую структуру данных, в которой уровни реализуются разными способами – один статически на базе массива (непрерывная реализация), другой – динамически с использованием адресных связей.

Требования к проекту:

1. Полная объектная реализация с определением классов для всех элементов

реализуемой структуры: информационные объекты (обязательно!), объекты-элементы списка (динамическая реализация), объекты-списки, объект-контейнер
2. Соблюдение принципа инкапсуляции – использование в классах только закрытых свойств и реализация необходимого набора методов доступа
3. Реализация в классах всех необходимых методов: конструкторы, методы доступа к свойствам, методы добавления и удаления на каждом из двух уровней, метод поиска (при необходимости)
4. Возможность сохранения всей структуры во внешнем файле с обратной загрузкой
5. Наличие модуля, демонстрирующего все возможности созданной библиотеки классов и обладающего удобным оконным пользовательским интерфейсом
6. Язык и среда разработки – по выбору: Delphi, Java, C++, C#
7. Оформленная пояснительная записка в MS Word или с помощью интернет-ресурса Google Docs

6. Учебно-методическое и информационное обеспечение дисциплины

Основная литература

1. Зыков, С. В. Объектно-ориентированное программирование: учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561434

2. Мейер, Б. Объектно-ориентированное программирование и программная инженерия: учебное пособие / Б. Мейер. — 4-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 284 с. — ISBN 978-5-4497-2464-9. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/133956.html

3. Тузовский, А. Ф. Объектно-ориентированное программирование: учебник для вузов / А. Ф. Тузовский. — Москва: Издательство Юрайт, 2025. — 213 с. — (Высшее образование). — ISBN 978-5-534-16316-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561394

Дополнительная литература

1.Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563618

2. Подбельский, В. В. Программирование. Базовый курс C#: учебник для вузов / В. В. Подбельский. — Москва: Издательство Юрайт, 2025. — 369 с. —

(Высшее образование). — ISBN 978-5-534-10616-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/560848

3. Ермаков, А. В. Объектно-ориентированное программирование в задачах на языке Java: учебное пособие / А. В. Ермаков. — Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2022. — 156 с. — ISBN 978-5-7433-3478-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/128034.html

4.Гуськова, О. И. Объектно ориентированное программирование в Java: учебное пособие / О. И. Гуськова. — 2-е изд. — Москва: Московский педагогический государственный университет, 2024. — 240 с. — ISBN 978-5-4263-0648-6. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: https://www.iprbookshop.ru/146115.html

Интернет-ресурсы и перечень ежегодно обновляемых современных профессиональных баз данных и информационных справочных систем:

1. «IPRSmart» www.iprbookshop.ru
2. Образовательная платформа «ЮРАЙТ» https://urait.ru
3. http://citforum.ru/
4. https://stackoverflow.com/
5. https://social.microsoft.com//Forums/ru-RU/home
6. www.greenfoot.org

7. Материально-техническое обеспечение дисциплины

В процессе изучение данной дисциплины используется учебная аудитория, кабинет для самостоятельной работы студентов, читальный зал, видеопроекционное оборудование, компьютер, оснащенный типовым пакетом системного и офисного ПО, в соответствии с Реестром материально-технического обеспечения аудиторного фонда Университета управления «ТИСБИ».

Комплект презентационного оборудования (компьютер с выходом в интернет, проектор, экран).

Персональный компьютер с выходом в интернет в компьютерных классах Университета для каждого студента на практических занятиях.

Пакет лицензионного системного и офисного ПО:

- Операционная система Microsoft Windows 10
- Microsoft Office 2013

Программное обеспечение, входящее в пакет, получает обновление в автоматическом, установленном разработчиком (компанией Microsoft) порядке, посредством сети Интернет.

Открытая среда разработки программного обеспечения - Lazarus.

Delphi 10.1 Berlin Professional Academic Concurrent License. License Certificate Number: 457049

NetBeans IDE. Бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения

IDE Visual Studio 2019 (Community) - Бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения

Greenfoot – бесплатная интерактивная среда разработки для языка Java

8. Оценка компетенций по изучаемой дисциплине

Для оценки компетентности рекомендуется использовать рейтинговую оценку знаний, умений и навыков студента по окончанию изучения каждого Модуля в соответствии с Положением о модульно-рейтинговой системе организации образовательного процесс. Итоговая оценка (в баллах) складывается из баллов, набранных по каждому Модулю (семестровая оценка) и баллов, набранных, непосредственно на экзамене (зачете).

Расчет набранных баллов по дисциплине осуществляется в следующей последовательности:

, где М – количество баллов по модулю; n – количество модулей

, где К - количество баллов на экзамене (зачете);

, где П – поощрительные баллы (от 1 до 5).

Уровень сформированности компетенций и их основные признаки оцениваются по таблицам.

Оценка сформированности компетенции ОПК-6. «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов» в части дисциплины «Объектно-ориентированное программирование»

№ п/п	Уровни сформированности компетенции	Основные признаки уровня	Инструменты оценки сформированности уровня
1	2	3	4
1	**Пороговый уровень (как минимально допустимый)** (от 60 до 70 баллов)	- знает назначение и основные понятия объектного подхода; - умеет разрабатывать классы для простых задач - владеет простейшими навыками разработки объектных программ	Практические задания Тестирование зачет Экзамен
2	**Базовый уровень (относительно порогового уровня)** **(**От 71 до 85 баллов**)**	- знает назначение, основные понятия объектного подхода и способы взаимодействия объектов и классов; - умеет разрабатывать наборы взаимосвязанных классов на основе композиции и наследования - владеет навыками разработки объектных программ средней сложности	Практические задания Тестирование зачет Экзамен
3	**Повышенный уровень (относительно порогового уровня) (От 86 до 100 баллов)**	- знает все понятия и механизмы объектного подхода; - умеет разрабатывать классы на основе двух механизмов взаимодействия, включая использование принципа полиморфизма - владеет навыками разработки объектных программ с использованием всех изучаемых в курсе механизмов	Практические задания Тестирование зачет Экзамен

Приложение 1

Методические рекомендации для обучающихся по освоению дисциплины

Студентам на первом занятии необходимо ознакомиться с Рабочей программой дисциплины, где прописаны цели, задачи и трудоемкость дисциплины. Перед началом изучения дисциплины необходимо повторить учебный материал обеспечивающих учебных дисциплин предшествующих курсов.

Затем необходимо ознакомиться с порядком изучения дисциплины, т.е. модульно-тематическим планом и пояснительной запиской с указанием этапов формирования заявленных компетенций.

И, наконец, ознакомиться с порядком оценивания результатов обучения, для чего необходимо изучить следующие документы: Положение о модульно-рейтинговой системе оценивания и Принципы оценки уровня знаний, умений и навыков (характеристика ответа).

Студент должен внимательно изучить перечень основной (дополнительной) литературы и взять необходимые учебники в библиотеке.