АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан ФИСТ
_____Ж.В. Игнатенко
«20» мая 2024 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование компьютерных игр

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) программы: Цифровизация экономической деятельности

Квалификация выпускника: Бакалавр

Форма обучения: очная, заочная

Год начала подготовки – 2024

Разработана
Канд. техн. наук, доцент
_____О.Х. Шаяхметов

Согласована
зав. кафедрой ПИМ
_____ Д.Г. Ловянников

Рекомендована
на заседании кафедры ПИМ
от «20» мая 2024 г.
протокол № 10
Зав. кафедрой _____ Д.Г. Ловянников

Одобрена
на заседании учебно-методической
комиссии ФИСТ
от «20» мая 2024 г.
протокол № 9
Председатель УМК _____ Ж.В. Игнатенко

Ставрополь, 2024 г.

Федоровский Александр Петрович
2024.05.29 22:40:19+03'00'

Содержание

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины является ознакомление студентов с комплексом современныхтехнологий и концепций, достаточных для профессиональной разработки компьютерных игр.

Дисциплина ориентирована на формирование системы понятий, знаний, умений и навыков в областиобъектно-ориентированного программирования, включающего в себя методы проектирования, анализа исоздания игровых продуктов и их сопровождения; развитие логического мышления, формированиенаучного мировоззрения, привитие склонности к творчеству.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование компьютерных игр» входит в часть, формируемуюучастниками образовательных отношений «Факультативные дисциплины (модули)».

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Знает: основные отличия игровых приложений от прочих;различные подходы к классификации компьютерных игр;основные жанры компьютерных игр и их принципиальные особенности; методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание игрового приложения

Умеет:описывать игровую ситуацию; видеть возможности применения технологии компьютерной игры при решении задач; применять различные методы поиска идей и создания инноваций: мозговой штурм, мозговая атака, метод фокальных объектов, метод маленьких человечков и др.; выполнять подбор среды разработки в соответствии с требованиями к игровому приложению (реализуемым возможностям, жанру, технически характеристикам и др.) реализовывать основные алгоритмы игрового приложения; реализовывать отдельные этапы разработки компьютерной игры.

Владеет:терминологией гейм-девелопинга; навыками автоматизации проектирования, производства, испытаний, оценки качества продукта, о направлениях развития методов и программных средств коллективной разработки компьютерных игр; навыками работы в отдельных средах визуального программирования; методами проектирования и разработки программного продукта; принципами построения, структуры и приемами работы с инструментальными средствами, поддерживающими создание игрового приложения.

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общий объем дисциплины составляет 2 зачетных единиц, 72 академических часа.

Очная форма обучения

Вид учебной работы	Всего часов	Триместры		
		4		
Контактная работа (всего)	**20**	20		
в том числе:				
1) занятия лекционного типа (ЛК)				

из них				
– лекции				
2) занятия семинарского типа (ПЗ)	20	20		
из них				
– семинары (С)				
– практические занятия (ПР)				
– лабораторные работы (ЛР)	20	20		
3) групповые консультации				
4) индивидуальная работа				
5) промежуточная аттестация				
Самостоятельная работа (всего) (СР)	**52**	52		
в том числе:				
Курсовой проект (работа)				
Расчетно-графические работы				
Контрольная работа				
Реферат	20	20		
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумами т.д.)	32	32		
Подготовка к аттестации				
Общий объем, час	72	72		
Форма промежуточной аттестации(Экзамен)	зачет	зачет		

Заочная форма обучения

Вид учебной работы	Всего часов	Триместры		
		4		
Контактная работа (всего)	**8**	8		
в том числе:				
1) занятия лекционного типа (ЛК)	4	4		
из них				
– лекции				
2) занятия семинарского типа (ПЗ)	4	4		
из них				
– семинары (С)				
– практические занятия (ПР)				
– лабораторные работы (ЛР)	4	4		
3) групповые консультации				
4) индивидуальная работа				
5) промежуточная аттестация				
Самостоятельная работа (всего) (СР)	**60**	60		
в том числе:				
Курсовой проект (работа)				
Расчетно-графические работы				
Контрольная работа				
Реферат	20	20		
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала	36	36		

учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумами т.д.)				
Подготовка к аттестации	4	4		
Общий объем, час	72	72		
Форма промежуточной аттестации (экзамен)	зачет	зачет		

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержание дисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1	Основы разработки компьютерных игр	Создание пустого игрового проекта "Chess" и знакомство с ним. Введение в XNA Game Studio 2.0. Разработка двумерных изображений для игры – доска, фигуры, вывод их на экран. 2D-графика в XNA Game Studio 2.0.
2	Инструментарий разработчика компьютерных игр	Устройства ввода, перемещение объектов Взаимодействие объектов Игровая физика Спрайтовая анимация Озвучивание игр
3	Начало разработки игры: игровая документация	Методы искусственного интеллекта (ИИ) в компьютерных играх Оформление игры Работа с файлами, сериализация Организация многоуровневых игр, конструктор уровней Сетевые игры

5.2. Структура дисциплины

Очная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов					
		Всего	ЛК	ЛР	ПР	С	СР
1	Основы разработки компьютерных игр	16			4		12
2	Инструментарий разработчика компьютерных игр	28			8		20
3	Начало разработки игры: игровая документация	28			8		20
	Групповая консультация						
	Промежуточная аттестация						
	Общий объем	72			20		52

Заочная форма обучения

№ раздела (темы)	Наименование раздела (темы)	Количество часов					
		Всего	ЛК	ЛР	ПР	С	СР
1	Основы разработки компьютерных	22	1		1		20

	игр						
2	Инструментарий разработчика компьютерных игр	22	1		1		20
3	Начало разработки игры: игровая документация	24	2		2		20
	Групповая консультация						
	Промежуточная аттестация	4					
	Общий объем	72	4		4		60

5.3. Занятия семинарского типа

очная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1	1	ПР	Основы разработки компьютерных игр	2
2	2	ПР	Инструментарий разработчика компьютерных игр	2
3	3	ПР	Начало разработки игры: игровая документация	6

заочная форма обучения

№ п/п	№ раздела (темы)	Вид занятия	Наименование	Количество часов
1	1	ПР	Основы разработки компьютерных игр	1
2	2	ПР	Инструментарий разработчика компьютерных игр	1
3	3	ПР	Начало разработки игры: игровая документация	2

5.4. Курсовой проект (курсовая работа, расчетно-графическая работа, реферат, контрольная работа)

Не предусмотрено

5.5. Самостоятельная работа

очная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1-3	Проработка и повторение лекционного материала	12
1-3	Подготовка к практическим занятиям	20
	Подготовка к аттестации	20
	Итого:	52

заочная форма обучения

№ раздела (темы)	Виды самостоятельной работы	Количество часов
1-3	Проработка и повторение лекционного материала	20
1-3	Подготовка к практическим занятиям	20
	Подготовка к аттестации	20
	Итого:	60

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине:

– сбор, хранение, систематизация и выдача учебной и научной информации;

– обработка текстовой и эмпирической информации;

– подготовка, конструирование и презентация итогов исследовательской и аналитической деятельности;

–самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;

– использование образовательных технологий в рамках ЭИОС для рассылки, переписки и обсуждения возникших учебных проблем.

Интерактивные и активные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине:

№ раздела (темы)	Вид занятия (ЛК, ПР, С, ЛР)	Используемые интерактивные и активные образовательные технологии	Количество часов ОФО/ЗФО
1	Л	Основы разработки компьютерных игр	1/1
2	Л	Инструментарий разработчика компьютерных игр	1/1
3	Л	Начало разработки игры: игровая документация	1/1

Практическая подготовка обучающихся

№ раздела (темы)	Вид занятия (ЛК, ПР, ЛР)	Виды работ	Количество часов	
			ОФО	ЗФО
-	-	-	-	-

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Описание показателей оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

Таблица 1 – Показатели оценивания и оценочные средства для оценивания результатов обучения по дисциплине

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Показатели оценивания (результаты обучения)	Процедуры оценивания (оценочные средства)	
			текущий контроль успеваемости	промежуточная аттестация

		Знает: основные отличия игровых приложений от прочих;различные подходы к классификации компьютерных игр;основные жанры компьютерных игр и их принципиальные особенности; методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание игрового приложения	Контрольные вопросы	Зачет (контрольные вопросы)
		Умеет:описывать игровую ситуацию; видеть возможности применения технологии компьютерной игры при решении задач; применять различные методы поиска идей и создания инноваций: мозговой штурм, мозговая атака, метод фокальных объектов, метод маленьких человечков и др.; выполнять подбор среды разработки в соответствии с требованиями к игровому приложению (реализуемым возможностям, жанру, технически характеристикам и др.) реализовывать основные алгоритмы игрового приложения; реализовывать отдельные этапы разработки компьютерной игры.	Практическое задание	Зачет (контрольные вопросы)

		Владееттерминологией гейм-девелопинга; навыками автоматизации проектирования, производства, испытаний, оценки качества продукта, о направлениях развития методов и программных средств коллективной разработки компьютерных игр; навыками работы в отдельных средах визуального программирования; методами проектирования и разработки программного продукта; принципами построения, структуры и приемами работы с инструментальными средствами, поддерживающими создание игрового приложения.	Практическое задание	Зачет (контрольные вопросы)
				зачет

7.1. ОЦЕНОЧНЫЕ СРЕДСТВА, КРИТЕРИИ И ШКАЛА ОЦЕНКИ

Типовые задания для текущего контроля
Типовые контрольные вопросы для устного опроса при текущем контроле

1. Основы разработки компьютерных игр
2. Очерки истории компьютерных игр
3. Этапы разработки компьютерной игры
4. Игровые профессии
5. Перспективы программиста-разработчика компьютерных игр
6. Инструментарий разработчика компьютерных игр
7. Состав игровых ресурсов
8. Игровая терминология
9. Обзор XNA GameStudio 2.0. – история, развитие, особенности применения
10. Психология компьютерных игр
11. Жанры компьютерных игр, анализ ведущих представителей жанров
12. Основные виды игр
13. Игры и обучение
14. Начало разработки игры: игровая документация
15. Концепт-документ
16. Дизайн-документ
17. План разработки игры
18. Трехмерная графика
19. Система координат

20. Преобразования в трехмерном пространстве
21. Объекты XNA для работы с 3D-графикой

отлично	1) студент полно излагает материал, дает правильное определение основных понятий; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.
хорошо	Студент дает ответ, удовлетворяющий тем же требованиям, что и для отметки, нодопускает1–2ошибки, которыесамже исправляет,и1–2недочетавпоследовательностиизыковом оформлении излагаемого.
удовлетворительно	студент обнаруживает знание и понимание основных положений данной темы, но: 1) излагает материал не полно и допускает не точности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
неудовлетворительно	студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «неудовлетворительно» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Типовые практические задания

Практическая работа 1. Основы разработки компьютерных игр.

Аннотация: В этой лабораторной работе мы рассмотрим среду разработки, в которой нам предстоит работать, а так же изучим стандартный игровой проект, на основе которого создаются компьютерные игры.

Ключевые слова: среда разработки, VisualStudio, пункт, поле, опыт, SolutionExplorer, error, list, пространство, деятельность, ПО, файл, Windows, C, исполнение, команда, папка, content, контент, конструктор класса, объект, Graphics, конструктор, класс, цикла, параметр, знание, реальное время, пространство имен, audio, утилита, input, мышь, манипулятор, storage, операции, net, live, pipeline, переменная, 3D, разделы

Задачи работы:

- Создать стандартный игровой проект
- Изучить его устройство
- Изучить содержимое и особенности файла Program.cs
- Ознакомиться с упрощенной схемой работы игры

- Изучить назначение методов стандартного игрового проекта и особенности подключаемых к нему пространств имен (файл Game1.cs)
  Создаем игровой проект
  Перед выполнением этой лабораторной работы убедитесь, что на ПК установлена *среда разработкиVisualStudio* и XNA 2.0.
  Для запуска среды разработки, выполните команду **Пуск** ⟹ **Все программы** ⟹ **MicrosoftVisual C# 2005 ExpressEdition**. Дальнейшие примеры будут показаны именно с использованием Express-версии. Выполните команду **File**⟹ **NewProject** – откроется окно **NewProject**, в котором нам предлагают выбрать один из стандартных типов проектов (рис.1.1.)

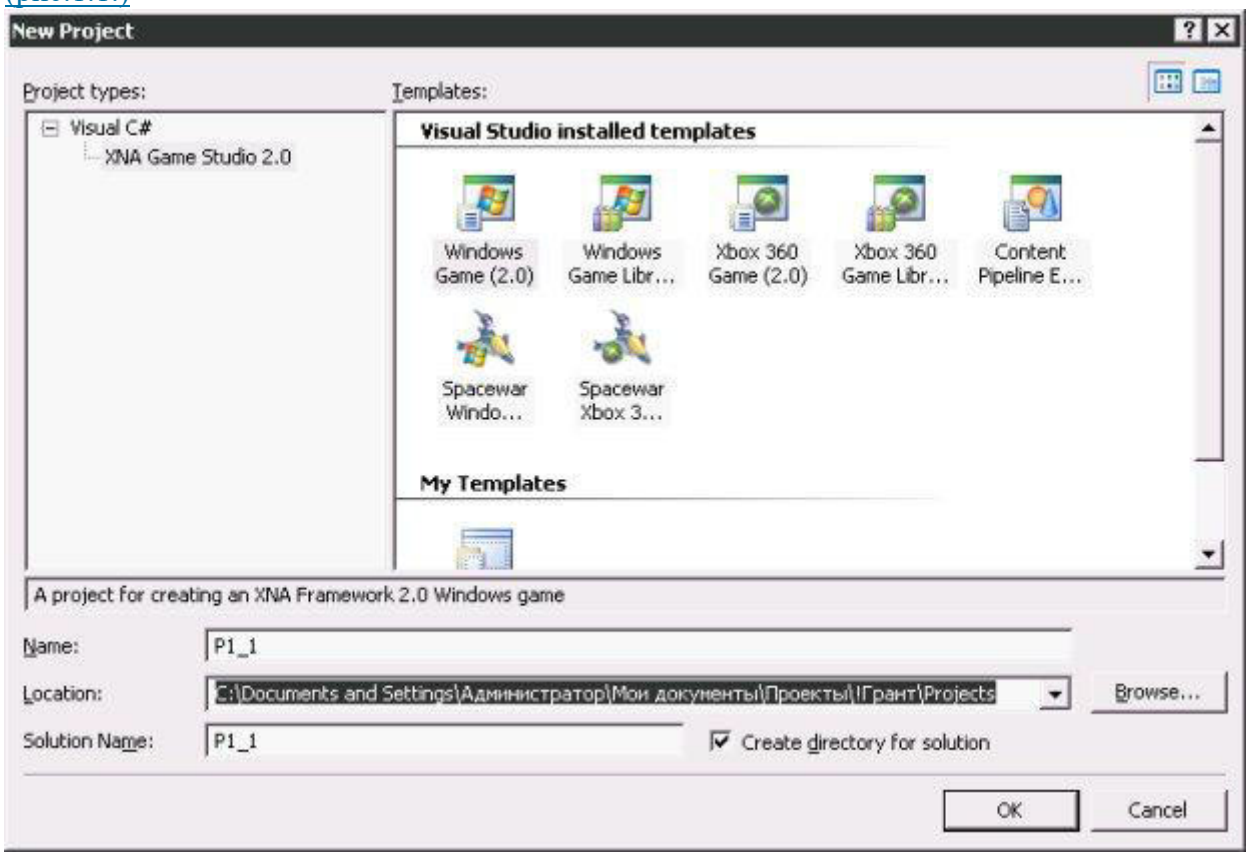

Рис. 1.1. Выбор типа создаваемого проекта
В левой части окна выберем *пункт***XNA GameStudio2.0**., в правой – **WindowsGame (2.0.)**.
Имя проекта, сгенерированное автоматически (*поле*Name) заменим на имя P1_1.
После нажатия на кнопку OK шаблонный проект будет создан. Вот как выглядит окно среды разработки после его создания (рис. 1.2.).

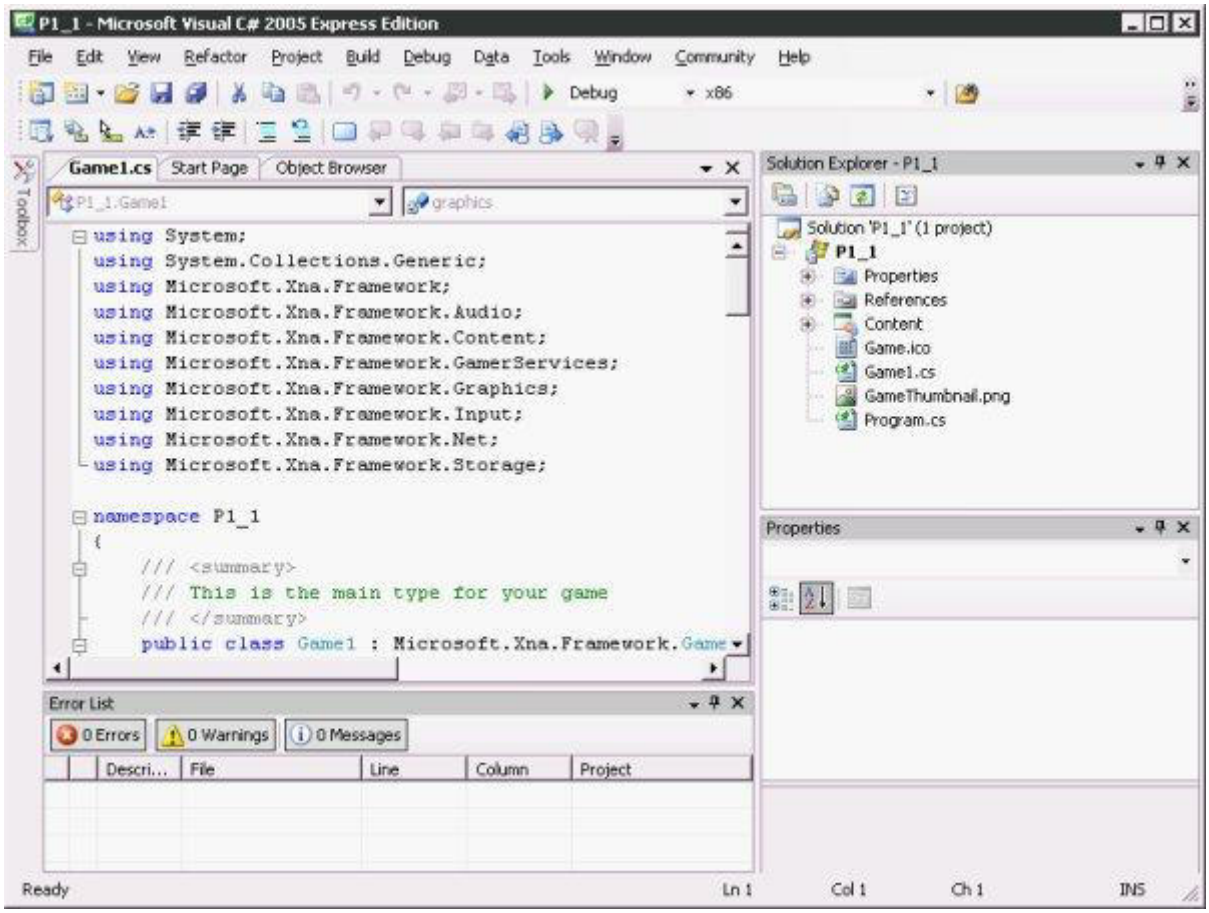

Рис. 1.2. Среда разработки после создания шаблонного проекта

Если вы выполняете сейчас эту лабораторную работу, то вы, скорее всего, имеете *опыт* работы с *VisualStudio*. Можно заметить, что внешний вид среды разработки выглядит вполне традиционно. Правую часть окна занимают панели *SolutionExplorer* и Properties, в нижней части окна можно видеть окно *ErrorList*, основное *пространство* занято окном редактора кода.

Напомним, что *SolutionExplorer* используется для просмотра и модификации файлов, которые входят в проект, для добавления новых файлов в проект, окно Properties применяется для модификации свойств выделенного объекта, окно *ErrorList* содержит сообщения об ошибках, которые генерируются при выполнении программы, а окно редактора кода применяется при редактировании текстов программ. Основная *деятельностьпо* созданию игры ведется именно с использованием вышеупомянутых инструментов.

Для тестового запуска игры служат команды **Debug⟹ StartDebugging и Debug⟹ StartWithoutDebugging**.

Если сейчас мы попытаемся запустить игру, мы увидим лишь пустое окно (рис. 1.3.).

Рис. 1.3. Окно шаблонного проекта
Это окно готово для ваших игровых экспериментов.
Рассмотрим важнейшие элементы содержимого панели *SolutionExplorer*(рис. 5.4.).

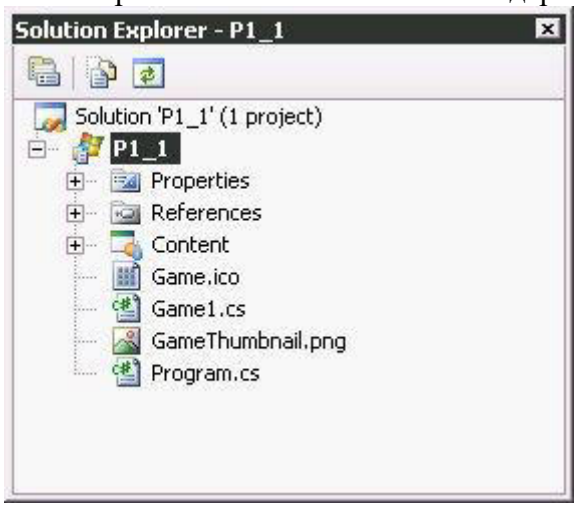

Рис. 1.4. Панель SolutionExplorer
*Файл*Program.cs – это обычный *файл* программы на **C#**. Точно такие же файлы обычно генерируются при создании консольных *Windows*-приложений на *C#*. Данный *файл* играет важную роль – в нем содержится точка входа в программу, с него начинается *исполнение* игры, и именно в нем содержится *команда*, которая запускает игру.
Файл Game1.cs – это *файл*, в котором хранится программный код игры. Как правило, вышеописанный Program.cs в модификации не нуждается, а вот Game1.cs – это *файл*, с

которым приходится работать в процессе создания игры. Причем, игры обычно включают в себя множество файлов с программным кодом.
ПапкаContent содержит игровой *контент* – то есть – файлы различных форматов, которые используются в игре.
Рассмотрим код стандартного проекта. Нам придется модифицировать этот код в процессе создания собственных игр.

Разбор кода стандартного игрового проекта

В листинге 1.1. приведеносодержимоефайла Program.cs.

```
using System;

namespace P1_1
{
static class Program
   {
      /// <summary>
      /// The main entry point for the application.
      /// </summary>
static void Main(string[] args)
      {
using (Game1 game = new Game1())
         {
game.Run();
         }
      }
   }
}
```

Листинг 5.1. КодфайлаProgram.cs

Как видите, выглядит он вполне стандартно. В теле статического метода Main содержатся команды создания нового объекта класса Game1. Объекту присваивается имя game, после чего выполняется метод Run объекта Game, который и запускает игру.
Упрощенно структуру игрового проекта можно представить так (Табл. 1.1.).

Таблица 1.1. Структура игрового проекта

Название	Описание
Инициализация	На этапе инициализации проекта проводится инициализация графической и звуковых подсистем, системы ввода данных.
Загрузка ресурсов	Здесь загружаются игровые ресурсы, такие, как текстуры и трехмерные модели, звуки, шрифты
Начало игрового цикла	Основные действия, которые выполняются при работе игры, организованы в виде игрового цикла. Фактически, игровой цикл – это обязательный элемент любой игры.
Считывание данных пользовательского ввода	Пользователь управляет игрой с помощью различных устройств ввода данных. В нашем случае это – клавиатура и мышь. Здесь так же может обрабатываться информация, поступающая от других устройств ввода. Например – от джойстика.
Игровые вычисления	Игровые вычисления включают в себя все вычисления, которые необходимы для организации игрового процесса. В частности, здесь обсчитываются позиции игровых объектов, проводятся вычисления, связанные с работой искусственного интеллекта, вычисления, связанные с проверкой столкновений игровых объектов, игровая физика, подсчет очков, набранных игроком и так далее. Если речь идет об оптимизации быстродействия игры,

	то такая оптимизация обычно касается именно игровых вычислений, так как на них тратится большая часть времени выполнения игрового цикла.
Проверка критерия прекращения игры	Игровой цикл продолжается до тех пор, пока игра не будет прекращена. Например, критерием остановки может быть истечение времени, выделенного игроку на выполнение игровой задачи, набор определенного количества очков, какое-то событие, произошедшее в процессе игры, принудительная остановка игры, если пользователь решил прекратить играть и так далее.
Вывод изображений, проигрывание звуков	После завершения этапа игровых вычислений в программе имеются данные для визуализации. В частности, имеются данные о новых позициях объектов, о сообщениях, которые нужно вывести на игровое поле, о звуках, которые нужно воспроизвести. На данном этапе проводится перемещение объектов по игровому полю, вывод сообщений, проигрывание звука.
Конец игрового цикла	Игровой цикл заканчивается после выполнения условия останова игры
Освобождение ресурсов	На последнем этапе работы игры проводится освобождение системных ресурсов, занятых игрой и выход из игры.

В стандартном игровом проекте все эти этапы представлены с помощью специальных методов. Эти методы унаследованы созданным игровым проектом от класса Game, который служит основой для проекта. Итак, это следующие методы (табл. 1.2.).

Таблица 1.2. Методы стандартного игрового проекта

Название	Описание
Game1()	Общая инициализация. Game1() – это конструктор класса Game1, который выполняется при создании объекта Game в процедуре Main файла Program.cs
Initialize()	Инициализация игры
LoadContent()	Загрузка игровых ресурсов
Run()	Запуск игрового цикла. Как было показано выше, метод Run() вызывается в процедуре Main файла Program.cs
Update()	Игровой цикл в стандартном проекте состоит из двух методов. Первый из них – это метод Update(), в котором выполняется прием пользовательского ввода, все необходимые вычисления, проверка критериев останова игры.
Draw()	Второй метод игрового цикла – это метод Draw(), который содержит код для визуализации игровой графики
UnloadContent()	По окончании игрового цикла этот метод освобождает системные ресурсы.

Теперь, когда мы ознакомились с типичной структурой игры и назначением стандартных методов, рассмотрим код класса Game1(Листинг 1.2.). В стандартном игровом проекте можно найти англоязычные комментарии. Здесь, для большей наглядности кода, они опущены. Комментарии, ключевые для понимания работы отдельных методов, приведены на русском языке.

```
using System;
usingSystem.Collections.Generic;
usingMicrosoft.Xna.Framework;
usingMicrosoft.Xna.Framework.Audio;
usingMicrosoft.Xna.Framework.Content;
usingMicrosoft.Xna.Framework.GamerServices;
usingMicrosoft.Xna.Framework.Graphics;
usingMicrosoft.Xna.Framework.Input;
```

```
using Microsoft.Xna.Framework.Net;
usingMicrosoft.Xna.Framework.Storage;

namespace P1_1
{

public class Game1 : Microsoft.Xna.Framework.Game
   {
GraphicsDeviceManager graphics;
SpriteBatchspriteBatch;

public Game1()
       {
graphics = new GraphicsDeviceManager(this);
Content.RootDirectory = "Content";
       }


protected override void Initialize()
       {
          // Сюдаследуетдобавитькодинициализации

base.Initialize();
       }


protected override void LoadContent()
       {
spriteBatch = new SpriteBatch(GraphicsDevice);

          // Сюданадодобавитькодзагрузкиигровыхресурсов
       }

protected override void UnloadContent()
       {
          // Кодвыгрузкиигровыхресурсов
       }


protected override void Update(GameTimegameTime)
{
          // Эта последовательность команд обрабатывает пользовательский ввод и позволяет
завершать игру,
 она рассчитана на использование джойстика
if (GamePad.GetState(PlayerIndex.One).Buttons.Back == ButtonState.Pressed)
this.Exit();

          // Сюда следует добавить код игровых вычислений

base.Update(gameTime);
       }
```

```
protected override void Draw(GameTimegameTime)
        {
graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

// Сюда следует добавить код визуализации игровых объектов

base.Draw(gameTime);
        }
    }
}
```
Листинг 5.2. Код файла Game1.cs

Рассмотрим каждый из приведенных выше методов. В листинге 1.3. приведен код метода Game1().

```
public Game1()
        {
graphics = new GraphicsDeviceManager(this);
Content.RootDirectory = "Content";
        }
```
Листинг 5.3. Кодметода Game1()

Метод Game1() – это *конструктор класса* Game1. *Команда*graphics = newGraphicsDeviceManager(this) ; создает новое игровое окно, а *объектgraphics* используется для проведения различных графических операций. Например, с помощью этого объекта можно настраивать ширину и высоту окна, переводить игру в полноэкранный режим.

Помимо описанных команд, автоматически включаемых в стандартный игровой проект, *конструктор* может содержать и другие команды. Например, команды дополнительной настройки объекта *graphics*.

В листинге 1.4. приведен код метода Initialize().

```
protectedoverridevoidInitialize()
        {
            // Сюда следует добавить код инициализации

base.Initialize();
        }
```
Листинг 5.4. Кодметодаinitialize()

*Команда*base.Initialize() ; вызывает метод Initialize() базового класса. Сюда же добавляют команды для инициализации звуковой подсистемы игры, здесь же выполняются игровые настройки. Надо отметить, что в этом методе инициализируется все, кроме графических ресурсов игры. ДлянихсуществуетметодLoadContent(), листинг 1.5.

```
protectedoverridevoidLoadContent()
        {
spriteBatch = newSpriteBatch(GraphicsDevice);

            // Сюданадодобавитькодзагрузкиигровыхресурсов
}
```
Листинг 5.5. Код метода LoadContent()

SpriteBatch – это *класс*, который позвляет выводить изображения на устройство, задаваемое параметром GraphicsDevice. В этот же метод добавляют команды для загрузки

графических ресурсов, шрифтов. Метод UnloadContent()(листинг. 1.6.) предназначен для освобождения системных ресурсов.

```
protectedoverridevoidUnloadContent()
        {
            // Код выгрузки игровых ресурсов
        }
```

Листинг 5.6. Код метода UnloadContent()

Здесь размещают команды для выгрузки игровых ресурсов. В частности, в данном методе можно использовать команду base.UnloadContent() ; - это метод базового класса, выгружающий ресурсы. Так же в этот метод добавляют команды для выгрузки шрифтов и других ресурсов.

Игровой цикл состоит из двух методов, выполняющихся циклически это методы Update() и Draw() – листинг 1.7.

```
protected override void Update(GameTimegameTime)
{
      // Сюда следует добавить код игровых вычислений

base.Update(gameTime);
   }


protected override void Draw(GameTimegameTime)
   {
graphics.GraphicsDevice.Clear(Color.CornflowerBlue);

// Сюда следует добавить код визуализации игровых объектов

base.Draw(gameTime);
   }
```

Листинг 5.7. Код методов Update() и Draw()

Сначала выполняется метод Update(), потом – метод Draw().

*Команда*base.Update(gameTime) ; метода Update() ответственна за организацию игрового *цикла*. Она снова и снова вызывает метод Update() базового класса. Именно эта *команда* является "двигателем" игры. Если ее убрать – игровой цикл не будет работать.

*Параметр*gameTime – представляет собой *объект* класса GameTime, который содержит информацию об игровом времени. *Знание* игрового времени нужно для организации взаимодействия внутри игры. *По* отношению к игровому времени можно выделить два типа игр. Первые – это игры реального времени, такие, как симуляторы. Вторые – это пошаговые игры, где игроку дается некоторое время – фиксированное либо нет – для обдумывания следующего хода. *Знание* игрового времени нужно для многих игровых вычислений. *Класс*GameTime имеет несколько важных свойств.

Свойство ElapsedGameTime позволяет узнать время, которое прошло после последнего вызова игрового *цикла*. TotalGameTime – игровое время, прошедшее с момента старта игры.

ElapsedRealTime – *реальное время*, прошедшее после последнего вызова игрового *цикла*. TotalRealTime – *реальное время*, прошедшее с момента старта игры.

*Команда*graphics.GraphicsDevice.Clear(Color.CornflowerBlue) ; метода Draw() очищает игровой экран, заливая его цветом CornflowerBlue. А *команда*base.Draw(gameTime) ; вызывает метод Draw() базового класса, который выводит на экран изображения.

Поговорим о пространствах имен (компонентах), которые подключены к стандартному проекту *по* умолчанию. Влистинге 1.8. приведеныихвызовы.

```
using System;
usingSystem.Collections.Generic;
usingMicrosoft.Xna.Framework;
usingMicrosoft.Xna.Framework.Audio;
usingMicrosoft.Xna.Framework.Content;
usingMicrosoft.Xna.Framework.GamerServices;
usingMicrosoft.Xna.Framework.Graphics;
usingMicrosoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Net;
usingMicrosoft.Xna.Framework.Storage;
```

Листинг 5.8. Подключаемые компоненты

Пространство именAudio содержит инструменты для работы со звуком. Для работы со звуком в XNA существует *утилита* XACT (MicrosoftCross-Platform*Audio*CreationTool). Подготовка музыки и звуков осуществляется с помощью этой утилиты, создается специальный XACT-проект, который загружается в игру. Применение XACT позволяет использовать одно и то же звуковое оформление как в *Windows*-играх, так и в Xbox360-играх.

Пространство именGraphics представляет собой графическую часть XNA и содержит в себе средства для работы с графикой.

Пространство именInput поддерживает работу со средствами ввода данных, в частности, это клавиатура, *мышь* и игровой *манипулятор* – джойстик. Как правило, при создании *Windows*-игр используют *мышь* и клавиатуру, при создании Xbox-игр – джойстик.

Пространство именStorage содержит средства для работы с файловой системой. Именно инструменты из *Storage* позволяют, например, сохранять игровые данные в файлы, читать их и выполнять другие файловые *операции*.

Пространство именNet поддерживает создание сетевых игр с помощью XNA. Реализация сетевого взаимодействия в XNA-играх осуществляется с помощью *WindowsLIVE*.

Пространство именContent содержит средства, обеспечивающие работу *ContentPipeline*. *ContentPipeline* – это очень важный механизм XNA Framework.

*Пространство имен*GamerService обеспечивает функции работы с сервисом *LIVE* – для игр под Xbox и *Windows*. Эти функции необходимы для создания сетевых игр. В частности, в GamerService находятся функции, позволяющие игроку регистрироваться для участия в игре, работать с информацией об игроке, хранить локальные пользовательские настройки для сетевых игр.

В разделе объявления свойств класса Game1 есть две переменных (листинг 1.9.).

```
GraphicsDeviceManagergraphics;
SpriteBatchspriteBatch;
```

Листинг 5.9. Переменные в разделе свойств класса

*Переменная*graphics имеет тип GraphicsDeviceManager – она предназначена для управления графическим устройством, которое используется для вывода информации.

*Переменная*spriteBatch имеет тип SpriteBatch – она нужна для вывода спрайтов на экран.

Критерии и шкала оценивания типовых практических работ

отлично	студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя понятия дисциплины.
хорошо	студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя понятия дисциплины.

удовлетворительно	студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном понятия дисциплины.
неудовлетворительно	ставится, если: студент не решил учебно-профессиональную задачу.

Типовые задания для промежуточной аттестации (зачет)

Перечень типовых контрольных вопросов для устного опроса на промежуточной аттестации (зачет)

1. Создание пустого игрового проекта "Chess" и знакомство с ним. Введение в XNA GameStudio 2.0.
2. Разработка двумерных изображений для игры – доска, фигуры, вывод их на экран. 2D-графика в XNA GameStudio 2.0.
3. Устройства ввода, перемещение объектов
4. Взаимодействие объектов
5. Игровая физика
6. Спрайтовая анимация
7. Озвучивание игр
8. Методы искусственного интеллекта (ИИ) в компьютерных играх
9. Оформление игры
10. Работа с файлами, сериализация
11. Организация многоуровневых игр, конструктор уровней
12. Сетевые игры

Критерии и шкала оценки зачета по дисциплине

Оценка	**Характеристики ответа студента**
Зачтено	Оценка «зачтено» выставляется, если студент успешно ответил на контрольный вопрос.
Не зачтено	Оценка «не зачтено» выставляется, если студент не ответил на контрольный вопрос.

7.2.МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ

Практическое задание	Оценочное средство, включающее совокупность условий, направленных на выполнение практического задания с целью формирования компетенций, соответствующих основным типам профессиональной деятельности. Процедура проведения данного оценочного

	мероприятия включает в себя: оценку правильности выполнения практического задания
Устный опрос	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п. Развернутый ответ обучающегося должен представлять собой связное, логически последовательное сообщение на заданную тему, показывать его умение применять определения, правила в конкретных случаях. Показатели для оценки устного ответа: 1) знание материала; 2) последовательность изложения; 3) владение речью и профессиональной терминологией; 4) применение конкретных примеров; 5) знание ранее изученного материала; 6) уровень теоретического анализа; 7) степень самостоятельности; 8) степень активности в процессе; 9) выполнение регламента. Уровень знаний обучающегося определяется оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Критерии и шкала оценки приведены в п. 3. Фонда оценочных средств.

Методические материалы, определяющие процедуры оценивания в рамках промежуточной аттестации

Зачет – это форма промежуточной аттестации, задачей которой является комплексная оценка уровней достижения планируемых результатов обучения по дисциплине.

Зачет по дисциплине проводится за счет часов, отведённых на изучение дисциплины.

Зачет по дисциплине включает в себя: собеседование преподавателя со студентами по контрольным вопросам.

Контрольные вопросы	Контрольный вопрос — это средство контроля усвоения учебного материала дисциплины. Процедура проведения данного оценочного мероприятия включает в себя: беседу преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме дисциплины.

После окончания ответа преподаватель объявляет обучающемуся оценку по результатам зачета, а также вносит эту оценку в зачетно-экзаменационную ведомость, зачетную книжку. Уровень знаний, умений и навыков обучающегося определяется оценками «зачтено», «не зачтено».

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Никитин, Б. Е. Теория игр, эконометрика: модели, алгоритмы,компьютерная реализация : учебное пособие / Б. Е. Никитин, М. Н. Ивлиев. — Воронеж :Воронежский государственный университет инженерных технологий, 2019. — 92 с. —ISBN 978-5-00032-433-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : — URL: https://www.iprbookshop.ru/95379.html

2. Шиловская, Н. А. Теория игр : учебник и практикум для вузов / Н. А. Шиловская. — Москва : Издательство Юрайт, 2021. — 318 с. — (Высшее образование). — ISBN 978-5-9916-8264-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/470213.

8.2. Дополнительная литература

Боресков, А. В. Основы компьютерной графики : учебник и практикум для вузов / А. В. Боресков, Е. В. Шикин. — Москва : Издательство Юрайт, 2021. — 219 с. — (Высшее образование). — ISBN 978-5-534-13196-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/468914

8.3. Программное обеспечение

MicrosoftWindows, Яндекс 360, MicrosoftOfficeProfessionalPlus 2019, GoogleChrome, Яндекс.Браузер.

8.4. Информационно-справочные системы

1. . База данных IT специалиста» [Электронный ресурс] – Режим доступа: http://info-comp.ru/

2. База данных программного обеспечения Oracle [Электронный ресурс] – Режим доступа: https://www.oracle.com/ru/inde x.html

8.5. Информационные справочные системы

1С: Библиотека - https://www.sksi.ru/environment/eor/library/

Справочно-правовая система «КонсультантПлюс» - http://www.consultant.ru/

Поисковые системы

Поисковая система Yandex- https://www.yandex.ru/

Поисковая система Rambler – https://www.rambler.ru/

8.6. Интернет-ресурсы

1. Интернет университет информационных технологий [Электронный ресурс] – Режим доступа : http://www.intuit.ru/

2. Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: http://www.iprbookshop.ru/

8.7. Методические указания по освоению дисциплины

Методические указания для подготовки к лекции

Аудиторные занятия планируются в рамках такой образовательной технологии, как проблемно-ориентированный подход с учетом профессиональных и личностных особенностей обучающихся. Это позволяет учитывать исходный уровень знаний обучающихся, а также существующие технические возможности обучения.

Методологической основой преподавания дисциплины являются научность и объективность.Лекция является первым шагом подготовки обучающихся к практическим занятиям. Проблемы, поставленные в ней, на практическом занятии приобретают конкретное выражение и решение.

Преподаватель на вводной лекции определяет структуру дисциплины, поясняет цели и задачи изучения дисциплины, формулирует основные вопросы и требования к результатам освоения. При проведении лекций, как правило, выделяются основные понятия и определения. При описании закономерностей обращается особое внимание на сравнительный анализ конкретных примеров.

На первом занятии преподаватель доводит до обучающихся требования к текущей и промежуточной аттестации, порядок работы в аудитории и нацеливает их на проведение самостоятельной работы с учетом количества часов, отведенных на нее учебным планом по направлению подготовки 09.03.03 Прикладная информатика и рабочей программой по дисциплине (п. 5.5).

Рекомендуя литературу для самостоятельного изучения, преподаватель поясняет, каким образом максимально использовать возможности, предлагаемые библиотекой АНО ВО СКСИ, в том числе ее электронными ресурсами, а также сделает акцент на привлечение ресурсов сети Интернет и профессиональных баз данных для изучения практики.

Выбор методов и форм обучения по дисциплине определяется:

– общими целями образования, воспитания, развития и психологической подготовки обучающихся;

– особенностями учебной дисциплины и спецификой ее требований к отбору дидактических методов;

– целями, задачами и содержанием материала конкретного занятия;

– временем, отведенным на изучение того или иного материала;

– уровнем подготовленности обучающихся;

–уровнем материальной оснащенности, наличием оборудования, наглядных пособий, технических средств.

Лекции дают обучающимся систематизированные знания по дисциплине, концентрируют их внимание на наиболее сложных и важных вопросах.

Лекции обычно излагаются в традиционном или в проблемном стиле (интерактивном). Интерактивный стиль позволяет стимулировать активную познавательную деятельность обучающихся и их интерес к дисциплине, формировать творческое мышление, прибегать к противопоставлениям и сравнениям, делать обобщения, активизировать внимание обучающихся путем постановки проблемных вопросов, поощрять дискуссию. Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления или процессов, выводы и практические рекомендации.

В конце лекции делаются выводы и определяются задачи на самостоятельную работу. Во время лекционных занятий рекомендуется вести конспектирование учебного материала, обращать внимание на формулировки и категории, раскрывающие суть того или иного явления или процессов, научные выводы и практические рекомендации. В случае недопонимания какой-либо части предмета следует задать вопрос в установленном порядке преподавателю.

Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов:

План-конспект – это развернутый детализированный план, в котором достаточно подробные записи приводятся по тем пунктам плана, которые нуждаются в пояснении.

Текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника.

Свободный конспект – это четко и кратко сформулированные (изложенные) основные положения в результате глубокого осмысливания материала. В нем могут

присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом.

Тематический конспект – составляется на основе изучения ряда источников и дает более или менее исчерпывающий ответ по какой-то схеме (вопросу).

Подготовленный конспект и рекомендуемая литература используются при подготовке к и практическим занятиям. Подготовка сводится к внимательному прочтению учебного материала, к выводу с карандашом в руках всех утверждений, к решению примеров, задач, к ответам на вопросы. Примеры, задачи, вопросы по теме являются средством самоконтроля.

Методические указания по подготовке к практическим работам

Целью практических работ является углубление и закрепление теоретических знаний, полученных обучающимися на лекциях и в процессе самостоятельного изучения учебного материала, а, следовательно, формирование у них определенных умений и навыков.

В ходе подготовки к практическим работам необходимо прочитать конспект лекции, изучить основную литературу, ознакомиться с дополнительной литературой, выполнить выданные преподавателем задания. При этом учесть рекомендации преподавателя и требования программы. Дорабатывать свой конспект лекции, делая в нем соответствующие записи из литературы. Желательно при подготовке к практическим работам по дисциплине одновременно использовать несколько источников, раскрывающих заданные вопросы.

Методические указания для выполнения самостоятельной работы

Самостоятельная работа обучающихся заключается:

В целях наиболее эффективного изучения дисциплины подготовлены различные задания, различающиеся по преследуемым целям.

Задания представлены – 1) контрольными вопросами, предназначенными для самопроверки; 2) письменными заданиями, включающими задачи и задание.

Задачи самостоятельной внеаудиторной работы обучающихся заключаются в продолжении изучения теоретического материала дисциплины и в развитии навыков самостоятельного анализа литературы.

I. Самостоятельное теоретическое обучение предполагает освоение студентом во внеаудиторное время рекомендуемой преподавателем основной и дополнительной литературы. С этой целью обучающимся рекомендуется постоянно знакомиться с классическими теоретическими источниками по темам дисциплины, а также с новинками литературы, статьями в периодических изданиях, справочных правовых системах.

Для лучшего понимания материала целесообразно осуществлять его конспектирование с возможным последующим его обсуждением на практических занятиях, на научных семинарах и в индивидуальных консультациях с преподавателем. Формы конспектирования материала могут быть различными:

1) обобщение – при подготовке такого конспекта студентом осуществляется анализ и обобщение всех существующих в доктрине подходов по выбранному дискуссионному вопросу раздела, в том числе, дореволюционных ученых, ученых советского и современного периода развития. Основная задача обучающегося заключается не только в изложении точек зрения по исследуемому вопросу, но и в выражении собственной позиции с соответствующим развернутым теоретическим обоснованием.

2) рецензия – при подготовке такого конспекта студентом осуществляется рецензирование выбранного источника по изучаемому дискуссионному вопросу, чаще всего, статьи и периодическом издании, тезисов выступления на конференции либо главы из монографии. Для этого студентом дается оценка содержанию соответствующего

источника по следующим параметрам: актуальность выбранной темы, в том числе убедительность обоснования актуальности исследования автором; соответствие содержания работы ее названию; логичность, системность и аргументированность (убедительность) выводов автора; научная добросовестность (наличие ссылок на использованные источники, самостоятельность исследования, отсутствие фактов недобросовестных заимствований текстов, идей и т.п.); научная новизна и др.

Формами контроля за самостоятельным теоретическим обучением являются теоретические опросы, которые осуществляются преподавателем на практических занятиях в устной форме, преследующие цель проверки знаний обучающихся по основным понятиям и терминам по теме дисциплины. В случае представления студентом выполненного им в письменном виде конспекта по предложенным вопросам темы, возможна его защита на практическом занятии или в индивидуальном порядке.

II. Ключевую роль в планировании индивидуальной траектории обучения по дисциплине играет *опережающая самостоятельная работа* (ОПС). Такой тип обучения предлагается в замену традиционной репродуктивной самостоятельной работе (самостоятельное повторение учебного материала и рассмотренных на занятиях алгоритмов действий, выполнение по ним аналогичных заданий). ОПС предполагает следующие виды самостоятельных работ:

познавательно-поисковая самостоятельная работа, предполагающая подготовку докладов, выступлений на практических занятиях, подбор литературы по конкретной проблеме, написание рефератов и др.;

творческая самостоятельная работа, к которой можно отнести выполнение специальных творческих и нестандартных заданий. Задача преподавателя на этапе планирования самостоятельной работы – организовать ее таким образом, чтобы максимально учесть индивидуальные способности каждого обучающегося, развить в нем познавательную потребность и готовность к выполнению самостоятельных работ все более высокого уровня. Студенты, приступая к изучению тем, должны применить свои навыки работы с библиографическими источниками и рекомендуемой литературой, умение четко формулировать свою собственную точку зрения и навыки ведения научных дискуссий. Все подготовленные и представленные тексты должны являться результатом самостоятельной информационно-аналитической работы обучающихся. На их основе студенты готовят материалы для выступлений в ходе практических занятий.

Подготовка к устному опросу

Самостоятельная работа обучающихся включает подготовку к устному опросу на практических занятиях. Для этого студент изучает лекции, основную и дополнительную литературу, публикации, информацию из Интернет-ресурсов.Кроме того, изучению должны быть подвергнуты различные источники информации.

Тема и вопросы к практическим занятиям по дисциплине доводятся до обучающихся заранее. Эффективность подготовки обучающихся к устному опросу зависит от качества ознакомления с рекомендованной литературой. Для подготовки к устному опросу студенту необходимо ознакомиться с материалом, посвященным теме практического занятия, в рекомендованной литературе, записях с лекционного занятия, обратить внимание на усвоение основных понятий дисциплины, выявить неясные вопросы и подобрать дополнительную литературу для их освещения, составить тезисы выступления по отдельным проблемным аспектам. В среднем, подготовка к устному опросу по одному практическому занятию занимает от 2 до 4 часов в зависимости от сложности темы и особенностей организации студентом своей самостоятельной работы.

Методические указания по подготовке к промежуточной аттестации

Промежуточная аттестация по дисциплине проводится в форме зачета.

Зачет – это форма промежуточной аттестации, задачей которой является комплексная оценка уровней достижения планируемых результатов обучения по дисциплине.

При подготовке к зачету необходимо повторить конспекты лекций по всем разделам дисциплины. На зачете студент должен подтвердить усвоение учебного материала, предусмотренного рабочей программой дисциплины, а также продемонстрировать приобретенные навыки адаптации полученных теоретических знаний к своей профессиональной деятельности. Дифференцированный зачет проводится в форме устного собеседования по контрольным вопросам, а также обучающемуся необходимо решить ситуационную задачу.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины требуется следующее материально-техническое обеспечение (специальные помещения):

- для проведения занятий лекционного типа

учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.

- для проведения занятий семинарского типа, практических занятий

учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.

- для проведения, текущего контроля и промежуточной аттестации

учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.

- для групповых и индивидуальных консультаций

учебная аудитория, оснащенная учебной мебелью, оборудованная проектором, ПК, экраном, доской.

- для самостоятельной работы:

помещение, оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Института

10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.Организация обеспечивает печатными и/или электронными образовательными ресурсами в формах адаптированных к ограничениям их здоровья.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения высшего образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.