

Документ подписан простой электронной подписью.
Информация о владельце:
ФИО: Костина Лариса Николаевна
Должность: проректор
Дата подписания: 25.06.2025 22:50:30
Уникальный программный ключ:
1800f7d89cf4ea7507265ba593fe87537eb15a6c

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ"

Факультет

Факультет государственной службы и управления

Кафедра

Информационных технологий

"УТВЕРЖДАЮ"

Проректор

_____ Л.Н. Костина

24.04.2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.01

"Алгоритмизация игровых задач"

Направление подготовки 09.03.03 Прикладная информатика

Профиль "Прикладная информатика в управлении корпоративными информационными системами"

Квалификация

Бакалавр

Форма обучения

очная

Общая трудоемкость

2 ЗЕТ

Год начала подготовки по учебному плану

2025

Донецк
2025

Составитель(и):
, ст.препод.

_____ Лебезова Э.М.

Рецензент(ы):
канд. экон. наук, доцент

_____ Стешенко И.В.

Рабочая программа дисциплины (модуля) "Алгоритмизация игровых задач" разработана в соответствии с:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Рабочая программа дисциплины (модуля) составлена на основании учебного плана Направление подготовки 09.03.03 Прикладная информатика Профиль "Прикладная информатика в управлении корпоративными информационными системами", утвержденного Ученым советом ФГБОУ ВО "ДОНАУИГС" от 24.04.2025 протокол № 12.

Срок действия программы: 2025-2029

Рабочая программа рассмотрена и одобрена на заседании кафедры
Информационных технологий

Протокол от 02.04.2025 № 9

Заведующий кафедрой:
канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2026 - 2027 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2026 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2027 - 2028 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2027 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2028 - 2029 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2028 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2029 - 2030 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2029 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент, Брадул Н.В.

(подпись)

РАЗДЕЛ 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ

1.1. ЦЕЛИ ДИСЦИПЛИНЫ

Основная цель освоения дисциплины - научить обучающихся осуществлять разработку простых игровых приложений на языке C# и платформе Unity.

1.2. УЧЕБНЫЕ ЗАДАЧИ ДИСЦИПЛИНЫ

- научить понимать как организовывать взаимодействие объектов игрового мира;
- изучить механизмы использования сторонних библиотек для собственных приложений;
- освоить основные конструкции и методы языка C#;
- изучить основы создания приложения в Unity;
- писать работающий код и создавать собственные проекты GameDev.

1.3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОПОП ВО: Б1.В

1.3.1. Дисциплина "Алгоритмизация игровых задач" опирается на следующие элементы ОПОП ВО:

Информатика и программирование

1.3.2. Дисциплина "Алгоритмизация игровых задач" выступает опорой для следующих элементов:

Теория алгоритмов

Разработка программных приложений

1.4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ:

ПКс-2.2: Разрабатывает и адаптирует прикладное программное обеспечение

Знать:

Уровень 1	Знать на уровне представлений алгоритмы, методы и средства обработки информации, применяемые при разработке игровых программ
Уровень 2	Знать на уровне представлений синтаксис языка программирования C#, основы объектно-ориентированного программирования
Уровень 3	Знать на уровне воспроизведения инструментальные средств разработки, доступные для платформы Unity

Уметь:

Уровень 1	Уметь практически реализовывать базовые алгоритмические конструкции на языке программирования C#
Уровень 2	Уметь создавать игровые сценарии с помощью обработки событий
Уровень 3	Уметь настраивать основные свойства стандартных элементов Unity

Владеть:

Уровень 1	Владеть навыками работы с интегрированной средой разработки, которая позволяет реализовывать разработку проекта.
Уровень 2	Владеть навыками детализации решения при проектировании программных продуктов
Уровень 3	Владеть навыками адаптации алгоритмов для решения конкретных задач

В результате освоения дисциплины "Алгоритмизация игровых задач" обучающийся должен:

3.1	Знать:
	Знать основные принципы геймификации, стандартные алгоритмы для реализации игровых сценариев в прикладном программном обеспечении
3.2	Уметь:
	Уметь разрабатывать игровые сценарии и создавать игровые программы на языках высокого уровня
3.3	Владеть:
	Иметь навыки владения программными средами, специализированными библиотеками и принципами GUI для разработки и адаптации игровых сценариев в прикладное программное обеспечение

1.5. ФОРМЫ КОНТРОЛЯ

Текущий контроль успеваемости позволяет оценить уровень сформированности элементов компетенций (знаний, умений и приобретенных навыков), компетенций с последующим объединением оценок и проводится в форме: устного опроса на лекционных и семинарских/практических занятиях (фронтальный, индивидуальный, комплексный), письменной проверки (тестовые задания, контроль знаний по разделу, ситуационных заданий и т.п.), оценки активности работы обучающегося на занятии, включая задания для самостоятельной работы.

Промежуточная аттестация

Результаты текущего контроля и промежуточной аттестации формируют рейтинговую оценку работы студента. Распределение баллов при формировании рейтинговой оценки работы студента осуществляется в соответствии с действующим локальным нормативным актом. По дисциплине "Алгоритмизация игровых задач" видом промежуточной аттестации является Зачет

РАЗДЕЛ 2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**2.1. ТРУДОЕМКОСТЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ**

Общая трудоёмкость дисциплины "Алгоритмизация игровых задач" составляет 2 зачётные единицы, 72 часов.

Количество часов, выделяемых на контактную работу с преподавателем и самостоятельную работу обучающегося, определяется учебным планом.

2.2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
Раздел 1. Алгоритмы и подходы геймификации в индустрии цифровых развлечений						
Тема 1.1. Геймификация в современном мире. /Пр/	1	2	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э7	0	
Тема 1.1. Геймификация в современном мире /Ср/	1	2	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э7	0	
Тема 1.2. Язык C# как инструмент разработки игр разной сложности /Пр/	1	6	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 1.2. Язык C# как инструмент разработки игр разной сложности /Ср/	1	2	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 1.3. Использование графических примитивов и библиотек персонажей /Пр/	1	2	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 1.3. Использование графических примитивов и библиотек персонажей /Ср/	1	2	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 1.4. Библиотека SmallBasic как тренажёр алгоритмизации игровых задач /Пр/	1	6	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э7	0	
Тема 1.4. Библиотека SmallBasic как тренажёр алгоритмизации игровых	1	6	ПКс-2.2		0	

задач /Ср/						
Консультация по С# /Конс/	1	1	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2	0	
Раздел 2. Алгоритмы и подходы геймификации в индустрии цифрового обучения						
Тема 2.1. Использование фреймворка .Net для создания игровых тренажёров /Пр/	1	4	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 2.1. Использование фреймворка .Net для создания игровых тренажёров /Ср/	1	6	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 2.2. Графические возможности проектов WindowsForm /Пр/	1	4	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 2.2. Графические возможности проектов WindowsForm /Ср/	1	4	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э1 Э2 Э3	0	
Тема 2.3. Использование игрового движка Unity для разработки приложений /Пр/	1	4	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э4 Э5 Э6 Э8	0	
Тема 2.3. Использование игрового движка Unity для разработки приложений /Ср/	1	6	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э4 Э5 Э6 Э8	0	
Тема 2.4. Визуализация моделей с помощью Unity /Пр/	1	4	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э4 Э5 Э6 Э8	0	
Тема 2.4. Визуализация моделей с помощью Unity /Ср/	1	8	ПКс-2.2	Л1.1Л2.1Л3 .1 Л3.2 Э4 Э5 Э6 Э8	0	
Консультация по Unity /Конс/	1	1			0	

РАЗДЕЛ 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе освоения дисциплины используются следующие образовательные технологии: практические занятия (ПР), самостоятельная работа студентов (СР) по выполнению различных видов заданий.

1. При изложении учебного материала используются такие методы:

- монологический;
- показательный;
- диалогический;
- эвристический;
- исследовательский;
- проблемное изложение.

2. Используются следующие принципы дидактики высшей школы:

- последовательность обучения;
- систематичность обучения;
- доступность обучения;
- принцип научности;
- принципы взаимосвязи теории и практики;
- принцип наглядности и др.

3. Практические занятия проходят в компьютерных классах для приобретения и закрепления практических навыков для освоения дисциплины в полной мере. Выполняя задания для практических работ, студенты осваивают новые технологии, изучают современные языки программирования и накапливают опыт разработки программного обеспечения для реализации подходов геймификации в различные сферы человеческой жизнедеятельности.

4. Самостоятельная работа предназначена для внеаудиторной работы студентов, связанной с конспектированием источников, учебного материала, изучением дополнительной литературы по дисциплине, подготовкой к текущему и семестровому контролю, а также выполнением индивидуального задания в форме реферата, эссе, презентации, эмпирического исследования.

РАЗДЕЛ 4. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

4.1. Рекомендуемая литература

1. Основная литература

	Авторы,	Заглавие	Издательство, год
Л1.1	Н. А. Тюкачев, В. Г. Хлебостроев	С#. Основы программирования: учебное пособие для вузов (272 с.)	4-е изд., стер. — Санкт-Петербург :Лань : электронно-библиотечная система, 2021

2. Дополнительная литература

	Авторы,	Заглавие	Издательство, год
Л2.1	Д. В. Денисов	Разработка игры в Unity. С нуля и до реализации: Справочник по разработке игр на движке Unity с применением языка программирования C# (272 с.)	Санкт-Петербург : Наука и Техника, 2021

3. Методические разработки

	Авторы,	Заглавие	Издательство, год
Л3.1	Лебезова Э.М.	Методические рекомендации для проведения практических занятий по учебной дисциплине «Алгоритмизация игровых задач» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (25)	Донецк : ДОНАУИГС, 2025
Л3.2	Лебезова Э.М.	Методические рекомендации по организации самостоятельной работы обучающихся по учебной дисциплине «Алгоритмизация игровых задач» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная	Донецк : ДОНАУИГС, 2025

Авторы,	Заглавие	Издательство, год
	информатика) / сост. Лебезова Э.М. : учебное пособие (22)	

4.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Изучение языка C#	https://www.tutorialspoint.com/csharp/index.htm
Э2	Изучение языка C#	http://www.tutorialsteacher.com/csharp/csharp-tutorials
Э3	Изучение языка C#	https://codescracker.com/c-sharp/
Э4	Изучение игрового движка Unity	http://unity3d.com/ru/community
Э5	Изучение игрового движка Unity	https://unity3d.com/learn
Э6	Изучение игрового движка Unity	https://mva.microsoft.com/ru/training-courses/-unity-8635?l=8oK9dYF1_8804984382
Э7	Справочник по работе с библиотекой SmallBasic	https://social.technet.microsoft.com/wiki/contents/articles/16369.small-basic-curriculum-lesson-2-2-turtle-graphics.aspx
Э8	Руководство по движку Unity	https://unityhub.ru/manual/index
Э9	ЭБС «ЗНАНИУМ»	https://znanium.ru
Э10	Научная электронная библиотека «КиберЛенинка»	https://cyberleninka.ru/
Э11	ЭБС «ЛАНЬ»	https://e.lanbook.com
Э12	ЭБС «SOCHUM»	https://sochum.ru

4.3. Перечень программного обеспечения

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства:

- Libre Office (лицензия Mozilla Public License v2.0.)
- 7-Zip (лицензия GNU Lesser General Public License)
- AIMP (лицензия LGPL v.2.1)
- STDU Viewer (freeware for private non-commercial or educational use)
- GIMP (лицензия GNU General Public License)
- Inkscape (лицензия GNU General Public License).

4.4. Профессиональные базы данных и информационные справочные системы

Не используются

4.5. Материально-техническое обеспечение дисциплины

Для проведения учебных занятий, предусмотренных образовательной программой, закреплены аудитории согласно расписанию учебных занятий:
рабочее место преподавателя, посадочные места по количеству обучающихся, доска меловая, персональный компьютер с лицензированным программным обеспечением общего назначения, мультимедийный проектор, экран, интерактивная панель.

РАЗДЕЛ 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

Задание к зачёту.

Разработать игру-тренажёр для изучения алгоритма бинарного поиска.

Требования к программе:

- Программа задумывает число в диапазоне заданным параметрами, пользователь угадывает число за N попыток, рассчитываемых согласно заданному диапазону. Если количество попыток превышает число N, игра считается проигранной.

- Программа должна представлять из себя многооконное приложение (заставка игры, основное окно тренажера, окно настройки параметров) с графическим интерфейсом (кнопки, метки, поля для ввода, изображения). Реализовать на языке C# в виде проекта Windows Form.
- Программа должна контролировать исключительные ситуации при вводе значений (отсутствие значения, неверный тип данных) и реагировать на них в виде сообщений для повторного ввода.
- Основное окно тренажера должно отображать количество удачных и неудачных попыток прохождения игры.
- Разместить код в репозитории GitHub и создать документацию в виде файла Markdown.

Вопросы к зачёту:

1. Приведите примеры геймификации в сфере отличной от развлекательной.
2. Принципы геймификации в образовании.
3. Возможности среды Visual Studio для разработчика игр и игровых алгоритмов
4. Основные простые типы C#
5. Операторы принятия решений в C#
6. Операторы цикла в C#
7. Ввод-вывод на консоль в C#
8. Графические возможности проектов Windows Form в C#
9. Работа с ассетами. Примитивные модели Unity.
10. Что такое спрайты, полигоны и коллайдеры?
11. Что такое физические материалы в Unity, как настроить гравитацию?
12. Как добавить в игру несколько уровней и переключаться между ними.
13. Как в Unity создать главное меню с выбором уровней.
14. Для чего нужен Менеджер пакетов Unity?
15. Как управлять объектом Turtle для реализации игровых алгоритмов?

5.2. Темы письменных работ

Не предусмотрены программой

5.3. Фонд оценочных средств

Фонд оценочных средств дисциплины "Алгоритмизация игровых задач" разработан в соответствии с локальным нормативным актом ФГБОУ ВО "ДОНАУИГС".

Фонд оценочных средств дисциплины "Алгоритмизация игровых задач" в полном объеме представлен в виде приложения к данному РПД.

5.4. Перечень видов оценочных средств

Тестирование (проводится на практических занятиях; контроль знаний раздела учебной дисциплины)

Практические задания (выполняются на практических занятиях за компьютером)

Индивидуальные задания (выполняются самостоятельно)

РАЗДЕЛ 6. СРЕДСТВА АДАПТАЦИИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ К ПОТРЕБНОСТЯМ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

1) с применением электронного обучения и дистанционных технологий.

2) с применением специального оборудования (техники) и программного обеспечения, имеющихся в ФГБОУ ВО "ДОНАУИГС".

В процессе обучения при необходимости для лиц с нарушениями зрения, слуха и опорно-двигательного аппарата предоставляются следующие условия:

- для лиц с нарушениями зрения: учебно-методические материалы в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные задания и консультации.

- для лиц с нарушениями слуха: учебно-методические материалы в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: учебно-методические материалы в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

РАЗДЕЛ 7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО УСВОЕНИЮ ДИСЦИПЛИНЫ

К числу важнейших умений, являющихся неотъемлемой частью успешного учебного процесса, относится

умение работать с различными литературными источниками, содержание которых так или иначе связано с изучаемой дисциплиной.

Подготовку к любой теме курса рекомендуется начинать с изучения презентационных материалов или учебной литературы, в которых дается систематизированное изложение материала, разъясняется смысл разных терминов и сообщается об изменениях в подходах к изучению тех или иных проблем данного курса.

В курсе широко используются актуальные и эффективные техники для более качественного обучения, социализации:

- обсуждения тем в специальных группах популярных социальных сетей;
- командная работа;
- удаленные технические консультации и видеоконференции;
- наглядные демонстрации современных интернет технологий.

Методические указания по организации самостоятельной работы.

Самостоятельная работа по дисциплине организована в следующих видах:

1. изучение теоретического материала по заданной теме;
2. анализ методов решения поставленной задачи;
3. выполнение индивидуальных заданий;
4. оценка достоверности полученных результатов;
5. отчет перед преподавателем по теоретической и практической части индивидуальной работы;
6. самостоятельная работа над индивидуальными заданиями представлены в виде элементов электронного курса в системе elearn

<http://elearn.dsum.org/course/view.php?id=2744>

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ»**

**Факультет государственной службы и управления
Кафедра информационных технологий**

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

«Алгоритмизация игровых задач»

Направление подготовки	09.03.03 Прикладная информатика
Профиль	«Прикладная информатика в управлении корпоративными информационными системами»
Квалификация	бакалавр
Форма обучения	очная

Донецк
2025

Фонд оценочных средств по дисциплине «Алгоритмизация игровых задач» для обучающихся 1 курса образовательной программы бакалавриата направления подготовки 09.03.03 Прикладная информатика (профиль: «Прикладная информатика в управлении корпоративными информационными системами») очной формы обучения

Автор,
разработчик:

стар. преподаватель Лебезова Э. М.

ФОС рассмотрен на заседании
кафедры

информационных технологий

Протокол заседания кафедры от

02.04.2025 г.

№

№ 9

Заведующий кафедрой

Н.В. Брадул

РАЗДЕЛ 1.
ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине «Алгоритмизация игровых задач»

1.1. Основные сведения об дисциплине

Таблица 1

Характеристика дисциплины
(сведения соответствуют разделу РПУД)

Образовательная программа	бакалавриат
Направление подготовки Профиль	09.03.03 Прикладная информатика «Прикладная информатика в управлении корпоративными информационными системами»
Количество разделов учебной дисциплины	2
Часть образовательной программы	Б1.В.01
Формы текущего контроля	Индивидуальные задания, практические задания.
<i>Показатели</i>	Очная форма обучения
Количество зачетных единиц (кредитов)	2
Семестр	1
Общая трудоемкость (академ. часов)	72
Аудиторная контактная работа:	34
Практические занятия	32
Консультации	2
Самостоятельная работа	36
<i>Форма промежуточной аттестации</i>	зачёт

1.2. Перечень компетенций с указанием этапов формирования в процессе освоения образовательной программы.

Таблица 2

Перечень компетенций и их элементов

Компетенция	Индикатор компетенции и его формулировка	Элементы индикатора компетенции	Индекс элемента
ПК-2.2	Применяет современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знать:	
		1. Знать на уровне представлений алгоритмы, методы и средства обработки информации, применяемые при разработке игровых программ	ПК 2.2 З-1
		2. Знать на уровне представлений синтаксис языка программирования С#, основы объектно-ориентированного программирования	ПК 2.2 З-2
		3. Знать на уровне воспроизведения инструментальные средств разработки, доступные для платформы Unity	ПК 2.2 З-3
		Уметь:	
		1. Уметь практически реализовывать базовые алгоритмические конструкции на языке программирования С#	ПК 2.2 У-1
		2. Уметь создавать игровые сценарии с помощью обработки событий	ПК 2.2 У-2

Компетенция	Индикатор компетенции и его формулировка	Элементы индикатора компетенции	Индекс элемента
		3. Уметь настраивать основные свойства стандартных элементов Unity	ПК 2.2 У-3
		Владеть:	
		1. Владеть навыками работы с интегрированной средой разработки, которая позволяет реализовывать разработку проекта.	ПК 2.2 В-1
		2. Владеть навыками детализации решения при проектировании программных продуктов	ПК 2.2 В-2
		3. Владеть навыками адаптации алгоритмов для решения конкретных задач	ПК 2.2 В-3

Таблица 3

Этапы формирования компетенций в процессе освоения основной образовательной программы

№ п/п	Контролируемые разделы (темы) дисциплины	Номер семестра	Код индикатора компетенции)	Наименование оценочного средства
Алгоритмы и подходы геймификации в индустрии цифровых развлечений				
1.	Тема 1.1 Геймификация в современном мире	1	ПК-2 3-1 ПК-2 У-1 ПК-2 В-1	Индивидуальные задания, практические задания, тест
2.	Тема 1.2 Язык C# как инструмент	1	ПК-2 3-1 ПК-2 У-1 ПК-2 В-1	Индивидуальные задания, практические задания, тест

	разработки игр разной сложности		ПК-2 З-2	
3.	Тема 1.3 Использование графических примитивов и библиотек персонажей	1	ПК-2 У-1 ПК-2 В-1 ПК-2 З-1 ПК-2 В-2	Индивидуальные задания, практические задания, тест
4.	Тема 1.4 Библиотека SmallBasic как тренажёр алгоритмизации игровых задач	1	ПК-2 У-1 ПК-2 В-2 ПК-2 З-2 ПК-2 В-2	Индивидуальные задания, практические задания, тест
Раздел №2. Алгоритмы и подходы геймификации в индустрии цифрового обучения				
5.	Тема 2.1 Использование фреймворка .Net для создания игровых тренажёров	1	ПК-2 У-2 ПК-2 В-2 ПК-2 З-2 ПК-2 У-3 ПК-2 В-3	Индивидуальные задания, практические задания, тест
6.	Тема 2.2. Графические возможности проектов WindowsForm	1	ПК-2 З-2 ПК-2 У-3 ПК-2 У-2 ПК-2 В-3	Индивидуальные задания, практические задания, тест
7.	Тема 2.3. Использование игрового движка Unity для разработки приложений	1	ПК-2 У-2 ПК-2 В-3 ПК-2 В-3	Индивидуальные задания, практические задания, тест
8.	Тема 2.4. Визуализация моделей с помощью Unity	1	ПК-2 У-3 ПК-2 В-3 ПК-2 В-3	Индивидуальные задания, практические задания, тест

РАЗДЕЛ 2.
ТЕКУЩИЙ КОНТРОЛЬ ПО ДИСЦИПЛИНЕ
«Алгоритмизация игровых задач»

Текущий контроль знаний используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной работой) обучающихся.

В условиях балльно-рейтинговой системы контроля результаты текущего оценивания используются в качестве показателя текущего рейтинга обучающегося. Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе преподавателя. Данный вид контроля стимулирует у обучающегося стремление к систематической самостоятельной работе по изучению учебной дисциплины.

Таблица 2.1.

Распределение баллов по видам учебной деятельности
(балльно-рейтинговая система)

Наименование Раздела/Темы	Вид задания						
		ПЗ		Всего за тему	КЗР	Р (СР)	ИЗ
	ЛЗ	УО	ТЗ				
P.1.T.1.1		5		5	5		20
P.1.T.1.2		5		5			
P.1.T.1.3		5		5			
P.1.T.1.4		5	5	10			
P.2.T.2.1		5		5	5		20
P.2.T.2.2		5		5			
P.2.T.2.3		5		5			
P.2.T.2.4		5	5	10			
Итого: 100б		40	10	50	10		40

ЛЗ – лекционное занятие;

УО – устный опрос;

ТЗ – тестовое задание;

ПЗ – практическое занятие;

КЗР – контроль знаний по Разделу;

Р – реферат.

СР – самостоятельная работа обучающегося

ИЗ – индивидуальное задание

2.1. Рекомендации по оцениванию индивидуальных и практических работ обучающихся

Максимальное количество баллов*	Критерии
Отлично	выставляется обучающемуся: если выполнены все пункты работы самостоятельно, без ошибок,

	если предложен более рациональный алгоритм решения задачи.
Хорошо	выставляется обучающемуся: если самостоятельно выполнены все пункты работы, допущены незначительные ошибки, если предложен более рациональный алгоритм решения задачи.
Удовлетворительно	выставляется обучающемуся: если самостоятельно (или с помощью преподавателя) выполнены все пункты работы, допущены грубые ошибки.
Неудовлетворительно	выставляется обучающемуся: если с помощью преподавателя выполнены не все пункты работы, допущены грубые ошибки.

* Представлено в таблице 2.1.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

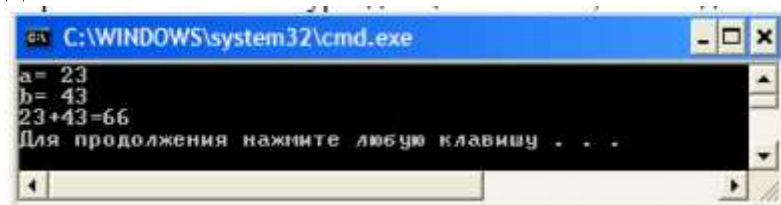
Индивидуальная работа

Раздел №1. Алгоритмы и подходы геймификации в индустрии цифровых развлечений

Тема 1.2. Язык C# как инструмент разработки игр разной сложности

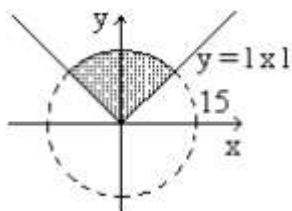
Вариант 1.

1. Написать программу, которая, реализует диалог с пользователем: запрашивает с клавиатуры два целых числа, и выводит на экран сумму данных чисел:



2. Написать программу, которая подсчитывает: площадь равностороннего треугольника, периметр которого равен p .
3. Написать программу, которая определяет: наибольшую цифру в натуральном двухзначном числе.

4. Дана точка на плоскости с координатами (x, y) . Составить программу, которая выдает одно из сообщений «Да», «Нет», «На границе» в зависимости от того, лежит ли точка внутри заштрихованной области, вне заштрихованной области или на ее границе. Области задаются графически следующим образом:



5. Составить программу. Замечание. При решении данных задач можно использовать как оператор *switch*, так и вложенные операторы *if*. Свой выбор обоснуйте.

Дан год. Вывести на экран название животного, символизирующего этот год по восточному календарю.

6. Вывести на экран: Замечание. При решении задачи можно использовать любой из операторов цикла *while*, *do while* и *for*. Свой выбор обоснуйте

Все целые числа из диапазона от A до B ($A \leq B$), оканчивающиеся на любую четную цифру.

7. Вывести на экран числа следующим образом:

```

0  1  2  3  4
0  1  2  3
0  1  2
0  1
0

```

8. Постройте таблицу значений функции $y=f(x)$ для $x \in [a, b]$ с шагом h .

$$y = \begin{cases} -4, & \text{если } x < 0; \\ x^2 + 3x + 4, & \text{если } 0 \leq x < 1; \\ 2, & \text{если } x \geq 1. \end{cases}$$

Индивидуальная работа

Раздел №1. Алгоритмы и подходы геймификации в индустрии цифровых развлечений

Тема 1.4. Библиотека SmallBasic как тренажёр алгоритмизации игровых задач

1. С помощью графического примитива (черепашки) напишите своё имя и слово ПРИВЕТ (с новой строчки) цветными карандашами. Пример



2. Реализуйте несколько сценариев, не меньше пяти, кроме 4-х стрелок (лево, право, верх, низ) управления черепашкой с помощью клавиатуры.
3. Нарисуйте с помощью объекта GraphicsWindow черепашкин домик.
4. В цикле игры задайте проверку для движения по 4-м направлениям. Необходимо, чтобы черепашка не могла уйти за границы окна.
5. Введите новый объект-приз, за которым бежит черепашка.
6. Придумайте и реализуйте сценарий изменений параметров игры при получении приза.
7. Выводите сведения о уровне игры в игровом поле.

Практическая работа

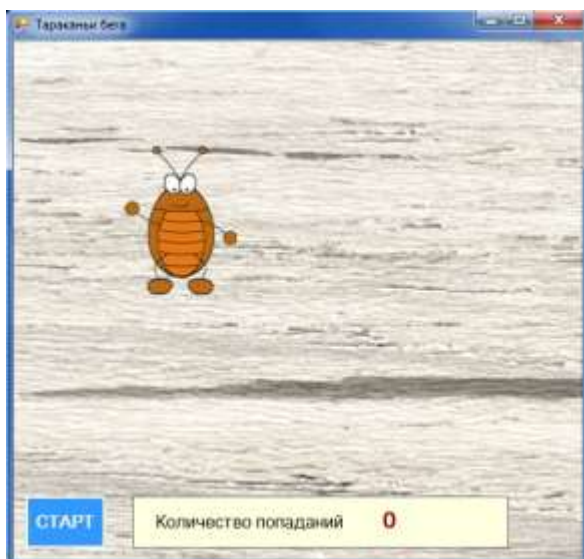
Раздел №2. Алгоритмы и подходы геймификации в индустрии цифрового обучения

Тема 2.2. Графические возможности проектов WindowsForm

Реализуйте игру "Погоня за тараканом".

Требования:

1. Используйте проект Windows Form для создания GUI игры. Используйте элементы Button, Label, PictureBox и др.



2. Объект случайным образом меняет свою локацию в поле игры с помощью элемента Timer. Задача игрока кликнуть по объекту мышкой. Подумайте, какое событие и какого объекта необходимо обработать.
3. Во время попадания объект должен изменить на короткое время свой вид (например, перевернуться или упасть) с помощью свойства Image.
4. После определённого времени (например, 1 мин.) игру остановить и вывести сообщение о количестве попаданий в диалоговом окне MessageBox.

2.2. Рекомендации по оцениванию тестовых заданий обучающихся

Максимальное количество баллов*	Критерии
Отлично	Выставляется обучающемуся, если обучающийся представил 90-100% правильных ответов.
Хорошо	выставляется обучающемуся, если обучающийся представил 75-89% правильных ответов.
Удовлетворительно	выставляется обучающемуся, если обучающийся представил 60-74% правильных ответов.
Неудовлетворительно	выставляется обучающемуся, если обучающийся представил менее 59% правильных ответов.

* Представлено в таблице 2.1.

ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ КОМПЕТЕНЦИЙ

Раздел №1. Алгоритмы и подходы геймификации в индустрии цифровых развлечений

Темы 1.1 – 1.3

1. Как называется среда программирования, в которой обычно разрабатывают программы на C#?

- a) Visual Studio
- b) Microsoft Studio
- c) Visual Express
- d) C# Studio

2. Как называется программа, преобразующая программу на языке высокого уровня в машинный код?

- a) Студия
- b) Библиотекарь
- c) Преобразователь
- d) Компилятор

3. Где хранится программа, выполняемая в данный момент центральным процессором?

- a) В оперативной памяти
- b) В программной памяти
- c) В самом процессоре
- d) На жестком диске

4. Какие приложения можно разрабатывать на Visual Studio ?

- a) Всё вышеперечисленное
- b) Только для мобильных устройств
- c) Только приложения для рабочего стола Windows
- d) Только веб-приложения

5. Какие команды может выполнять центральный процессор компьютера?

- a) Команды всех языков программирования (Ассемблер, C#, Basic и др.)
- b) Команды языка C#
- c) Машинный код
- d) Команды языка ассемблера

6. Если в программе описана переменная `var x = 10`, то какие значения она сможет принимать?

- a) Только 10 (значение изменить нельзя)

- b) Только целые числа
- c) Любые значения (числа, строки и т.д.)
- d) Только числа

7.Что нужно сделать, чтобы в программе использовать команды черепашки?

- a) Всё вышеперечисленное
- b) Написать `using Microsoft.SmallBasic.Library;` в начале программы
- c) Подключить к программе библиотеку `SmallBasicLibrary.dll`

8.Какую команду нужно выполнить, чтобы черепашка двигалась, не оставляя следа?

- a) `Turtle.Color \ Background.Color`
- b) `Turtle.Color\”White”`
- c) `Turtle.PenDown()`
- d) `Turtle.PenUp()`

9.Какой тип будет у переменной `x`, описанной как `var x = “Привет”`?

- a) `string`
- b) Константа
- c) `int`
- d) `var`

10.Где надо писать команды программы на `C#`, чтобы они начали выполняться при запуске программы?

- a) С первой строки файла `Program.cs`
- b) Сразу после команд `using`
- c) Внутри фигурных скобок в функции `Main`
- d) В любом из файлов проекта

11.Какие фрагменты программы лучше выносить в отдельные функции?

- a) Те, которые выполняют некоторое законченное осмысленное действие
- b) Любые
- c) Те, которые повторяются более одного раза

12.Что позволяют делать функции?

- a) Сокращать объем программы

- b) Сводить сложную задачу к более простым
- c) Описывать «новую команду», которой потом можно пользоваться несколько раз

13. Сколько параметров может быть у функции?

- a) От 0 до 3
- b) Ни одного или один
- c) Любое количество параметров

14. Какая функция в модуле GraphicsWindows служит для рисования прямоугольника?

- a) DrawMultiLine
- b) DrawSquare
- c) Draw
- d) DrawRectangle

15. Как будет описываться заголовок функции для рисования n-угольника с заданной длиной стороны?

- a) static void Draw(int n, int len)
- b) static int len Draw(int n)
- c) static void Draw()
- d) static void Draw(int len)

16. Необходимо написать программу рисования машинки. Какие функции имеет смысл выделить?

- a) Рисование контура, рисование отдельного колеса, рисование окон
- b) Рисование контура, рисование колес, рисование окон
- c) Нет нужны выделять функции
- d) Рисование отдельного колеса, рисование отдельного окна

17. Что такое формальный параметр?

- a) Переменная, указанная при вызове функции как параметр
- b) Переменная, указанная при описании функции в заголовке как параметр
- c) Выражение, указанное при вызове функции как параметр

18. К вещественным типам данных относятся типы:

- a) char

- b) long
- c) float
- d) int
- e) double
- f) bool

19. Тип int является:

- a) размерным типом, определенным пользователем
- b) базовым и размерным типом данных
- c) ссылочным типом, определенным пользователем
- d) базовым и ссылочным типом данных

20. Результат работы фрагмента программы

```
int x = 10;
while (x < 0)
{
    Console.WriteLine('{0} ',x);
    x = x - 3;
}
```

- a. тело цикла не выполнится ни разу
- b. 10 7 4
- c. фрагмент кода содержит ошибку
- d. 10 7 4 1

21. Для какого числа X истинно выражение $(X > 1) \ \&\& \ (! (X < 5) \ || \ (X < 3))$

- a) 2
- b) 1
- c) 4
- d) 3

22. В результате работы программного фрагмента:

```
int c;
Console.WriteLine(c);
```

переменная c примет значение:

- a) 0
- b) никакое, т.к. произойдет ошибка этапа выполнения программы
- c) неопределённое значение
- d) никакое, т.к. произойдет ошибка этапа компиляции

23. В результате работы программного фрагмента:

```
const int c=1;
```

```
c++;
```

константа c примет значение:

- a) 1
- b) 2
- c) никакое, т.к. произойдет ошибка этапа выполнения программы
- d) никакое, т.к. произойдет ошибка этапа компиляции

24. Определите значение целочисленной переменной a после выполнения фрагмента программы:

```
a \ 3 + 8*4;
```

```
b \ a / 10 + 14;
```

```
a \ 2 + b \ % 10;
```

- a) 8
- b) 10
- c) 9
- d) 11

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТ)

Задание к зачёту.

Разработать игру-тренажёр для изучения алгоритма бинарного поиска.

Требования к программе:

- Программа задумывает число в диапазоне заданным параметрами, пользователь угадывает число за N попыток, рассчитываемых согласно заданному диапазону. Если количество попыток превышает число N, игра считается проигранной.

- Программа должна представлять из себя многооконное приложение (заставка игры, основное окно тренажера, окно настройки параметров) с графическим интерфейсом (кнопки, метки, поля для ввода, изображения). Реализовать на языке C# в виде проекта Windows Form.

- Программа должна контролировать исключительные ситуации при вводе значений (отсутствие значения, неверный тип данных) и реагировать на них в виде сообщений для повторного ввода.

- Основное окно тренажера должно отображать количество удачных и неудачных попыток прохождения игры.

- Разместить код в репозитории GitHub и создать документацию в виде файла Markdown.

Вопросы к зачёту:

1. Приведите примеры геймификации в сфере отличной от развлекательной.
2. Принципы геймификации в образовании.
3. Как управлять объектом Turtle для реализации игровых алгоритмов?
4. Каким образом организовать процесс игры в цикле?
5. Каким образом организовать пользовательское управление объектом Turtle в игре?
6. Возможности среды Visual Studio для разработчика игр и игровых алгоритмов
7. Основные простые типы C#
8. Операторы принятия решений в C#
9. Операторы цикла в C#
10. Ввод-вывод на консоль в C#
11. Графические возможности проектов Windows Form в C#
12. Основные графические элементы проектов Windows Form в C#
13. Свойства и события элементов. Обработка событий.
14. Работа с ассетами. Примитивные модели Unity.
15. Что такое спрайты, полигоны и коллайдеры?
16. Что такое физические материалы в Unity, как настроить гравитацию?
17. Как добавить в игру несколько уровней и переключаться между ними.
18. Как в Unity создать главное меню с выбором уровней.
19. Для чего нужен Менеджер пакетов Unity?
20. Основные типы игр и тренажёров.
21. Как работать с циклами в SmallBasic?
22. Как создать подпрограмму в SmallBasic?
23. Как вводить данные с клавиатуры в SmallBasic?
24. Как выводить данные на экран в SmallBasic?
25. Как работать с графикой в SmallBasic?
26. Как работать с аудио в SmallBasic?
27. Как создать пользовательский интерфейс в SmallBasic?
28. Как настроить окно программы в SmallBasic?
29. Как работать со встроенной документацией в SmallBasic?
30. Как управлять объектом Turtle для реализации игровых алгоритмов?
31. Какой язык программирования используется в SmallBasic?
32. Как создать новый проект в SmallBasic?
33. Как создать новую переменную в SmallBasic?
34. Как выполнить арифметические операции в SmallBasic?
35. Как создать условную конструкцию в SmallBasic?