

**Министерство образования и науки Смоленской области
Муниципальное бюджетное общеобразовательное учреждение
«Центр образования № 1 «Академия детства» города Смоленска**

Принята на заседании
педагогического совета
Протокол № 7 от 23.05.2025

УТВЕРЖДАЮ
Директор Центра
И.В. Новикова
Приказ № 309-ОД от 26.05.2025

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
«С# для школьников.
От простых программ до Unity3D»**

Направленность: **техническая**

Уровень освоения: **базовый, продвинутый**

Возраст обучающихся: **12-16 лет**

Срок реализации: **1 учебный год (72 часа)**

Автор – составитель:
Айрапетян Артур Робертович,
педагог дополнительного
образования структурного
подразделения детский технопарк
«Кванториум»

Смоленск, 2025 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «С# для школьников. От простых программ до Unity3D» базового и продвинутого уровней (далее – программа) имеет техническую направленность, разработана в соответствии с основными нормативными правовыми актами Российской Федерации, Смоленской области, общеобразовательной организации:

1. Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

2. Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

3. Приказ Министерства труда и социальной защиты Российской Федерации от 22 сентября 2021 г. № 652-н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых».

4. Концепция развития дополнительного образования детей до 2030 года (утверждена Распоряжением Правительства Российской Федерации от 31 марта 2022 г. № 678-р).

5. Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства Российской Федерации от 29.05.2015 г. №996-р).

6. Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (утверждены Постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28).

7. Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Департамента государственной политики в сфере воспитания детей и молодежи Министерства образования и науки Российской Федерации от 18 ноября 2015 г. № 09-3242).

8. Приказ Министерства образования и науки Российской Федерации России от 23.08.2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».

9. Постановление Администрации города Смоленска от 29 января 2019 г. № 193-адм «Об утверждении Положения об организации предоставления дополнительного образования детей в муниципальных бюджетных учреждениях дополнительного образования, подведомственных управлению образования и молодежной политики Администрации города Смоленска».

10. Устав МБОУ «ЦО № 1 «Академия детства».

11. Положение о детском технопарке «Кванториум».

12. Программа развития общекультурных компетенций обучающихся детского технопарка «Кванториум».

Актуальность программы.

Введение в изучение промышленных языков программирования является важнейшим этапом в становлении юных программистов. Под понятием промышленный язык программирования имеется ввиду языки, на которых пишут профессиональные программисты, создавая различные приложения: десктопные, мобильные, Web – приложения, серверы, искусственный интеллект и т.д. Все современное оборудование абсолютно бесполезно, если к нему не написано программное обеспечение, которое определяет его работу, его функционал. Без программного обеспечения любой станок, компьютер, современный автомобиль, космические технологии и т.д. представляют собой не более чем груды железа. Если железо – это тело, то программное обеспечение – это душа. Всё программное обеспечение для вышеперечисленного оборудования пишется на промышленных языках программирования (высокого или низкого уровня).

В связи с этим изучение основ данных языков является неотъемлемой частью становления будущих профессиональных программистов. На сегодняшний день умение программировать также важно, как уметь читать. Именно такие требования устанавливает современный мир, поскольку жизнь человека теснейшим образом связана с использованием компьютерных технологий, которые развиваются стремительными темпами. Войти в мир современных компьютерных технологий намного проще и эффективнее, когда человек молод, поэтому дети являются основным субъектом, когда речь заходит об обучении основ компьютерной грамотности и программирования.

На сегодняшний день существует множество промышленных языков программирования, на которых создаётся программное обеспечение различного назначения. В качестве примеров наиболее популярных языков программирования можно привести такие языки, как: Python, C/C++, Java, C#, JavaScript, Perl, Kotlin, XAML и т.д. Все эти языки отличаются друг от друга, поэтому знать в совершенстве их все не представляется возможным. Однако, всем этим языкам присуще общие черты, а у некоторых из этих языков даже общий синтаксис. Поэтому детальное изучение одного языка позволит будущему программисту перейти без проблем на другой язык или изучить дополнительный. Многие языки настолько похожи друг на друга (например, C# и Java), что изучение одного языка практически влечет за собой знание второго языка программирования.

Особо следует отметить, что не существует идеального или самого лучшего языка программирования. Не правильно утверждать, что эту программу можно написать только, например на C++. В принципе любой из вышеперечисленных языков программирования может быть использован для выполнения любой задачи. Поэтому выбор языка — это задача, которую должен для себя решить каждый из будущих программистов.

В рамках настоящей программы рассматривается введение в язык программирования общего назначения C#. Актуальность изучения данного языка очевидна. Язык программирования C# был разработан компанией Microsoft в 2001 году. Изначально данный язык предназначался для созданий

приложений исключительно для ОС Windows, однако на сегодняшний день данный язык программирования стал действительно универсальным. На нём можно разрабатывать приложения для ОС Windows, MacOS, Android. Поэтому на сегодняшний день программисты, владеющие данным языком программирования очень востребованы на рынке труда. Более того в России на данном языке разрабатывают 3D приложения для Сбербанка. Мировые компании, такие как Tesla также активно используют этот язык программирования. На C# создаются различные приложения и симуляторы 3D для обучения и образовательного процесса. Ну и главное – это игровая индустрия! Один из самых мощных и продвинутых движков для создания 2D и 3D игр – Unity3D – использует именно C# для написания скриптов для объектов игр. C# активно используется для создания компьютерных игр класса AAA, то есть игр с огромным бюджетом и командой создателей.

Изучение языка программирования C# позволит обучающимся освоить важнейший принцип современного программирования – объектно – ориентированное программирование (ООП), поскольку C# является исключительно объектно-ориентированным языком. К тому же C# - строго типизированный язык, что позволяет писать программы, содержащие малое количество багов. Код на C# является безопасным и надежным.

Ещё один важный момент, определяющий актуальность изучения языка программирования C#. Это язык, который создан одной из самых крупных IT компаний в мире – Microsoft, а потому этот язык постоянно обновляется и имеет поддержку. Плюс данный язык ориентирован на ОС Windows, а как известно на сегодняшний день Windows самая популярная ОС в мире. Доля компьютеров с ОС Windows составляет порядка 60%. Сообщество программистов, использующих язык программирования C#, составляет более 1 миллиона человек по всему миру. Это означает, что существует много литературы, посвященной данному языку, форумов и сообществ.

И последнее. Так как C# является языком программирования, который много унаследовал от C/C++ и очень похож на язык Java, то изучив C# можно с небольшими усилиями перейти на данные языки программирования. Или, наоборот, с них перейти на C#.

Отличительные особенности программы, новизна.

Особенность данной программы является её стопроцентная практичность. Принцип, который использовался при составлении программы очень прост – **меньше теории и больше практики**. Более того, обучение основам языка программирования начинается сразу с написания первой программы. В данной методике ученик совместно с педагогом в первую очередь пишет работающую программу. И по ходу написания кода педагог разъясняет те или иные особенности языковых конструкций.

Следует отметить, что данный подход применим для изучения любого другого языка программирования. К примеру, эта методика была с успехом применена при изучении курса визуального программирования Scratch 3.

Ещё одной особенностью данной программы является её ориентация именно на детей. Практика преподавания языков программирования показала,

что никто не любит сортировать массивы, создавать списки, решать непонятные логические задачи с помощью языков программирования. Но все дети (абсолютно все) любят игры, любят программы с визуальным интерфейсом. А потому данная программа разработана с учетом пожеланий именно детей. Достаточно создать одну компьютерную игру с нуля (даже простую), чтобы ребёнок смог освоить основные языковые конструкции. Причем освоить их в интересной для себя форме.

Особенностью и новизной в данной программе является также ее ориентация на математику. Программа составлена таким образом, что ученик при изучении курса программирования и написания программы волей – неволей сталкивается с множеством математических понятий. Причем **важно отметить**, что со многими математическими понятиями он еще никогда не встречался. Принцип очень прост – хочешь написать свою собственную игру, изучи понятия координат, 2D и 3D пространства, понятие векторов, вещественных чисел, комбинаторики и многое другое. И главное – это не помеха. Ребенок ради своей игры готов изучить основы этих сложнейших понятий.

Социальная значимость программы.

Реализация настоящей программы обеспечит решение важнейших задач в воспитании подрастающего поколения. На сегодняшний день Россия нуждается в многочисленных технических кадрах. Инженеры и программисты нужны практически во всех сферах деятельности нашей огромной страны: это и космическая сфера, и военная сфера, научно-техническая, авиационная промышленность, банковский сектор, безопасность государства и многое другое. Закладывать основы технического мышления нужно как можно раньше и именно с программ, которые будут восприниматься ребёнком с большим интересом и энтузиазмом.

Постепенно осваивая данную программу, ученик будет все больше и больше ощущать свою значимость в огромном мире IT-технологий. Разработка реальных проектов и отправка их на конкурсы позволит повысить самооценку ребёнка и придаст ему уверенности в дальнейшем обучении и освоении более сложных программ. Кроме того, уже с самого начала освоения данной программы обучающийся сможет начать разрабатывать проект, который реально можно использовать в жизни (к примеру, учебная программа «Словарный диктант»).

Также полное освоение данной программы позволит ребёнку определиться, подходит для него профессия программиста, разработчика, а также определиться с направлением деятельности (программирование игр, разработка системных программ, учебных и т.д.).

Адресат программы.

Возраст детей, участвующих в реализации программы, составляет 12-16 лет. В данном возрасте дети имеют высокие способности к быстрому овладению теми или иными видами деятельности, включая программирование. Тем самым это определяет большие потенциальные возможности их всестороннего (в большей части технического) развития. Однако не следует

забывать, что данный курс отличается, к примеру, от курса визуального программирования на Scratch. Здесь предполагается более рутинная работа с написанием кода (реального) на английском языке, что может вызвать некоторые трудности у обучающихся. В связи с этим предполагается задействовать в данной программе детей более старшего возраста (12-16 лет). В этом возрасте дети более осознанно выбирают темы и программы для обучения и способны изначально настроить себя на выполнения сложных и однообразных задач. При должном занятии результат не заставит себя ждать. Если ученик готов к трудной, сложной и кропотливой работе, то в скором времени он научится создавать приложения простой и средней сложности, а также самостоятельно изучать более подробно дополнительный материал для совершенствования своих умений и навыков.

Возможность использования программы для детей с ОВЗ и инвалидностью: общие заболевания (нарушение дыхательной системы, пищеварительной, эндокринной систем, сердечно-сосудистой системы и т.д.), нарушение опорно-двигательного аппарата (НОДА). Возможно одновременное участие детей с инвалидностью и ОВЗ, и детей без инвалидности и ОВЗ.

Обучение по данной программе доступно детям, находящимся в трудной жизненной ситуации (например, из многодетных и малоимущих семей, из семей беженцев, вынужденных переселенцев), так как не требует финансовых затрат.

Объем и срок реализации программы.

Срок освоения программы – 1 год. Общее количество учебных часов за период обучения – 72 часа.

Формы и режим занятий.

Основная форма обучения – очная. При необходимости программу (или часть программы) можно реализовать с применением дистанционных образовательных технологий, используя платформу «Сферум».

Занятия проводятся 1 раз в неделю по 2 академических часа (академический час – 40 минут) с перерывом между занятиями в 10 минут. Учитываются нормы СанПиН. Занятия проводятся в соответствии с установленным расписанием.

Для образовательного процесса используются:

- *групповые занятия*: как правило для разработки крупного проекта (к примеру, учебной программы или компьютерной игры). Целесообразно использовать небольшие группы по 5 – 6 человек;
- *индивидуальные формы обучения*. Необходимы, когда преподаватель дает ученикам задание по разработке индивидуального проекта (в частности разработка проекта на конкурсы).

Основные формы работы:

- практические работы по созданию реальных приложений для ОС Windows;
- беседы;
- конкурсы, олимпиады;
- презентации, защита проектов;
- онлайн мероприятия.

ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель программы – формирование у обучающихся алгоритмических и креативных способностей через изучение промышленного языка программирования общего назначения C#.

Реализация данной цели предполагает решение следующих задач:

Образовательные задачи:

- изучить главные принципы современного программирования;
- изучить основы блочного программирования и создания алгоритмов для решения поставленных задач;
- формировать навыки проектирования и разработки цифровых продуктов (игр, анимации, презентаций и т.д.);
- развивать критическое, алгоритмическое, творческое мышление;
- научить работать в паре, команде, малой группе, выполнять коллективный проект и т.д.;
- освоить интегрированную среду разработки Sharp Develop и Visual Studio;
- формировать уважение и любовь к техническим профессиям.

Развивающие задачи:

- научить ставить перед собой цели (например, создать игру);
- научить планировать и систематизировать свои действия и знания;
- развивать аккуратность, внимательность к деталям, ответственность;
- развивать коммуникативные способности (умение работать и общаться в коллективе);
- научить планировать свои действия и работу в целом, развить в ребенке самостоятельность;
- научить применять полученные знания в решении конкретных практических задач, определять подходы и методы для достижения поставленной цели, отбирать необходимые средства для достижения поставленной цели;
- научить осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.

Воспитательные задачи:

- создать условия для развития общекультурных компетенций, формирования информационной и технологической культуры обучающихся, представлений о целостности картины мира и способах ее отражения посредством IT-технологий;
- содействовать формированию ценностного отношения к компьютерным играм и потребности создания игр с учетом традиционных российских ценностей и традиций;
- способствовать формированию детского коллектива как средства развития личности;
- создать условия для самоопределения учащихся в профессиональном выборе;
- повысить уровень любознательности и самостоятельности в решении задач.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

В ходе обучения детьми основ программирования на языке C# обучающиеся смогут улучшить свои математические способности, развить логическое мышление, проявить свои творческие способности. Ребята освоят базовые принципы объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм), которые являются основой современной парадигмы программирования. К концу данного курса обучающиеся научатся создавать простые консольные приложения и классические Windows приложения.

Пользу программирования для детского мышления трудно переоценить. При написании программ (даже простейших) ребенок будет сталкиваться с логическими операциями и математическими действиями, принципом работы компьютера. Это поможет ему при изучении школьного курса математики и информатики в дальнейшем.

Также предполагается, что по мере изучения языка программирования на C# ребенок морально и психологически будет подготовлен к дальнейшему освоению профессии программиста в будущем.

Предметные результаты:

В результате освоения программы обучающиеся будут уметь:

- составлять несложные линейные, разветвляющиеся, циклические алгоритмы на языке программирования C#;
- использовать логические значения, операции и выражения с ними;
- оперировать понятиями класс, объект, обработка событий;
- создавать и выполнять несложные программы (игры, презентации, анимацию и т.д.) на языке программирования C#;
- пользоваться принципами модульного (функционального) программирования;
- строить алгоритмы блочного программирования.

Метапредметные результаты:

В результате прохождения начального курса программирования на C# дети будут уметь:

- оформлять свои мысли, высказывать свою точку зрения;
- слушать других, пытаться принимать другую точку зрения;
- уважительно относиться к позиции другого человека, уметь договариваться;
- участвовать в общей беседе, соблюдая правила речевого поведения;
- владеть способами позитивного взаимодействия со сверстниками.

Личностные результаты:

В результате освоения программы обучающиеся научатся:

- осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности;
- проводить рефлекссию своей учебно-познавательной деятельности;
- продуктивно использовать коммуникативные навыки для установления дружеских отношений с окружающими и решения поставленных задач;
- выстраивать взаимоотношения с окружающими людьми на основе

принципов толерантности с учетом традиционных российских ценностей;

- ориентироваться в мире культуры, науки и техники, осознавать их значимость в своей жизни и развитии инновационного потенциала России;

- учитывать аксиологические принципы при разработке своих собственных образовательных продуктов;

- определять свои профессиональные предпочтения и стратегии их дальнейшего развития.

УЧЕБНЫЙ ПЛАН

№	Раздел, тема	Количество часов			Форма контроля
		Всего	Теория	Практика	
1.	Модуль 1. Знакомство интегрированной средой программирования Sharp Develop/Visual Studio Community. Основы языка программирования C#	16	2	14	Педагогическое наблюдение, выполнение практических заданий
2.	Модуль 2. Основы программирования классических приложений Windows (оконные приложения Windows Form)	10	0	10	Педагогическое наблюдение, выполнение практических заданий
3.	Модуль 3. Прикладное программирование на языке C#	8	0	8	Педагогическое наблюдение, выполнение практических заданий
4.	Модуль 4. Основы технологии программирования WPF	16	2	14	Педагогическое наблюдение, тестирование
5.	Модуль 5. Основы Unity3D	20	2	18	Педагогическое наблюдение, выполнение практических заданий
	Итоговое занятие	2	0	2	Защита проекта
	ВСЕГО:	72	6	66	

СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

Курс начального программирования на языке C# целесообразно разбить на модули. В каждом модуле обучающиеся будут получать соответствующие знания и навыки, и отрабатывать их при выполнении практических заданий. Каждая тема модуля использует и закрепляет знания, полученные из предыдущих тем.

Модуль 1 – включает в себя 16 часов занятий. Сюда входит знакомство с интегрированной средой разработки программного обеспечения Sharp Develop или Visual Studio Community. Также в данный модуль входят основные занятия по изучению основ программирования на высокоуровневом языке общего назначения C#. Освоение данного модуля осуществляется путем практических занятий (создания несложных программ, презентаций, игр и т.д.). Итог освоения данного модуля – обучающиеся свободно ориентируются в интегрированной среде программирования Sharp Develop (или Visual Studio) и умеют использовать основные ключевые слова и конструкции языка программирования C#. В конце данного модуля ученик должен уметь создавать простые консольные приложения для решения тех или иных практических задач, а также приложения средней сложности, имеющие реальное практическое применение, например в учебных целях. Ниже на рисунках 1,2,3,4 показан пример учебной программы «Словарный диктант», разработанной в рамках индивидуального проекта при изучении модуля 1.

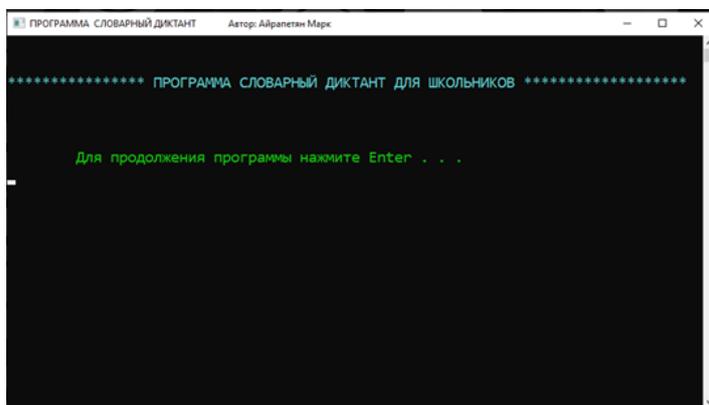


Рис.1 Стартовое окно учебного консольного приложения «Словарный диктант»

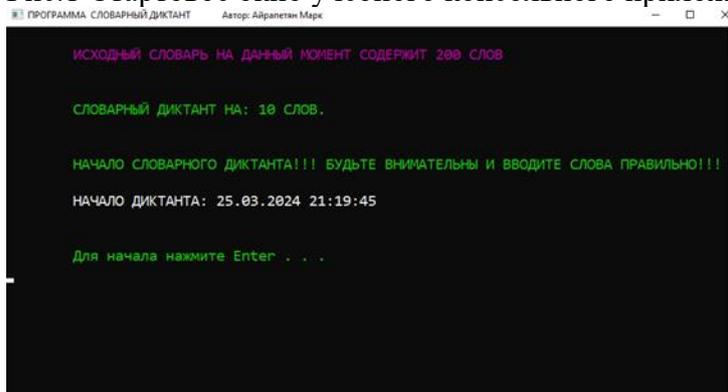


Рис2. Диалоговое окно для взаимодействия с пользователем

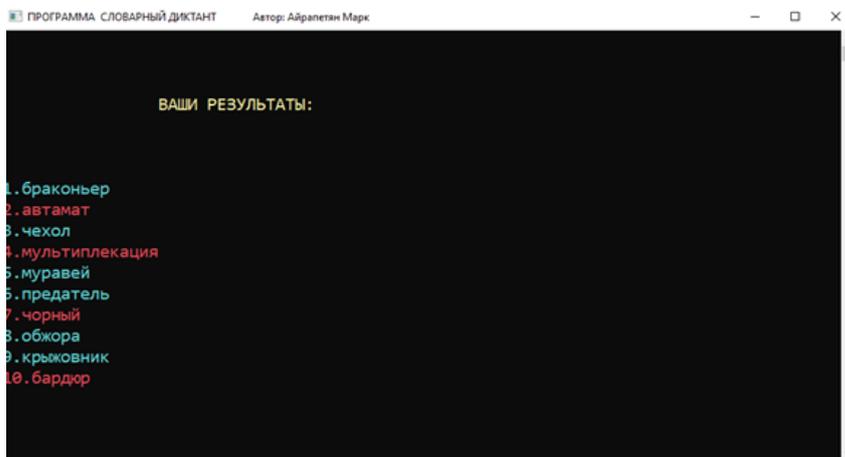


Рис3. Окно результата.

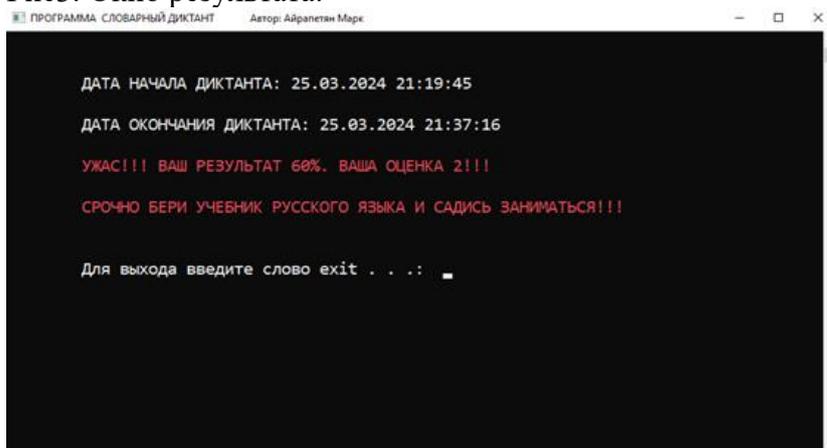


Рис4. Окно оценок и рекомендаций

Модуль 2 – включает в себя 10 часов занятий. В данном модуле рассматриваются основы программирования классических графических приложений Windows (оконные приложения). Обучающиеся учатся создавать привлекательные приложения с графическим интерфейсом, подробнее знакомятся с понятием формы, компонентов формы и их свойств.

Ниже на рисунке 5 показан пример учебной программы «Windows Калькулятор», разработанной в рамках индивидуального проекта при изучении модуля 2.



Рис5. Windows приложение «Калькулятор»

Модуль 3 – включает в себя 8 часов занятий. Является продолжением Модуля 2, в котором обучающиеся учатся программировать сложные алгоритмы, а также создавать практические программы на языке программирования C# с использованием более сложных конструкций языка и математических понятий.

Ниже на рисунках 6 и 7 представлена маленькая часть индивидуального проекта «Графики функций». Программа позволяет строить математические кривые любой сложности в 2D пространстве.

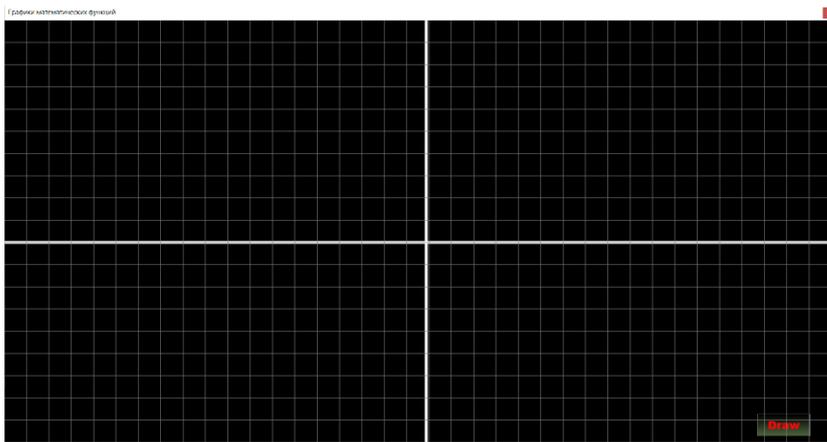


Рис6. Оси декартовой системы координат и координатная сетка.

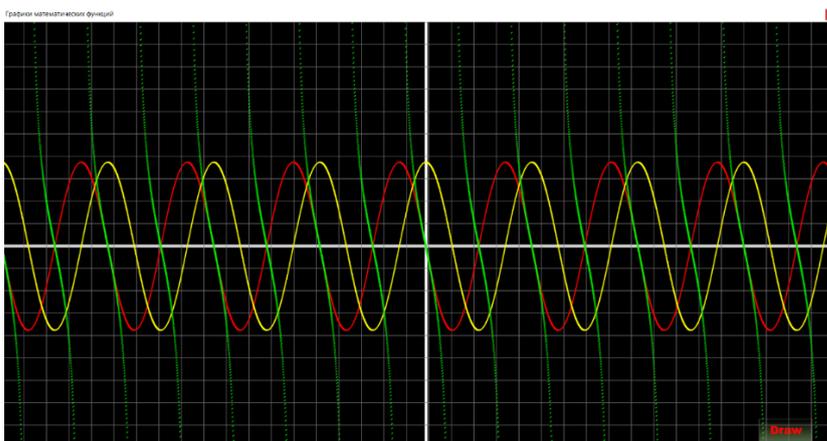


Рис7. Графики функций Sin, Cos, Tan

Модуль 4 – включает в себя 16 часов занятий. В данном модуле обучающиеся, которые освоили предыдущие модули на достаточно высоком уровне начинают осваивать новую технологию Windows программирования, а именно создания WPF приложений на языке C#. WPF (Windows Presentations Foundations) – это современный стандарт Windows приложений, позволяющий создавать привлекательные современные программы, которые можно использовать в различных сферах, включая образовательную. По сути, модуль 4 – это логическое продолжение модуля 2. Поэтому освоение ранних модулей на высоком уровне имеет ключевое значение для изучения WPF технологии программирования.

Ниже на рисунках 8,9,10 показан пример программы написанной с использованием технологии WPF при изучении модуля 4.



Рис 8. Заставка к игре «Крестики-Нолики»



Рис9. Основное окно игры «Крестики-Нолики»



Рис10. Окно настроек

Модуль 5 – самый длительный модуль, который включает в себя 20 часов. В данном модуле рассматриваются основы Unity3D - современного фреймворка для разработки 2D и 3D приложений. В качестве приложений могут быть игры для настольных компьютеров, мобильных устройств и консолей, а также социально значимые программы (различные учебные симуляторы, медицинские программы и т.д.). Предполагается, что данный модуль будет восприниматься детьми с особым энтузиазмом, поскольку данный модуль – это практически полностью творческая работа, где дети могут полностью реализовать свои творческие способности и применить фантазию.

Поскольку рабочим языком программирования для Unity3D является язык C#, то глубокое его освоение из предыдущих модулей позволит обучающимся использовать весь его потенциал для создания своих собственных приложений. Естественно предполагается, что дети захотят создать свою собственную компьютерную игру (ведь основное предназначение Unity3D именно разработка игровых приложений).

На рисунках 11 и 12 показан 3D мир, созданный в Unity3D в рамках данной программы.

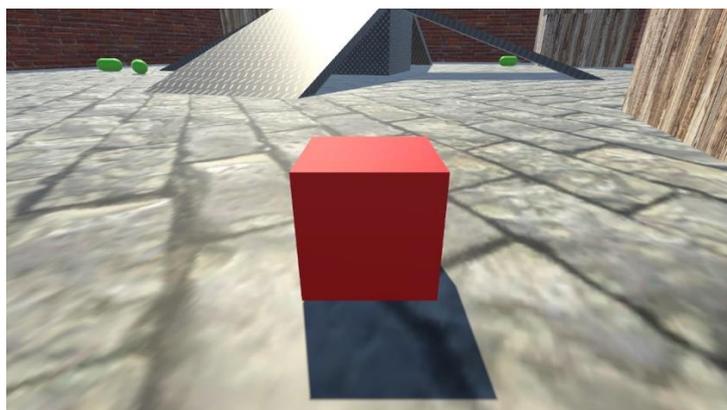


Рис.11 3D игровой мир

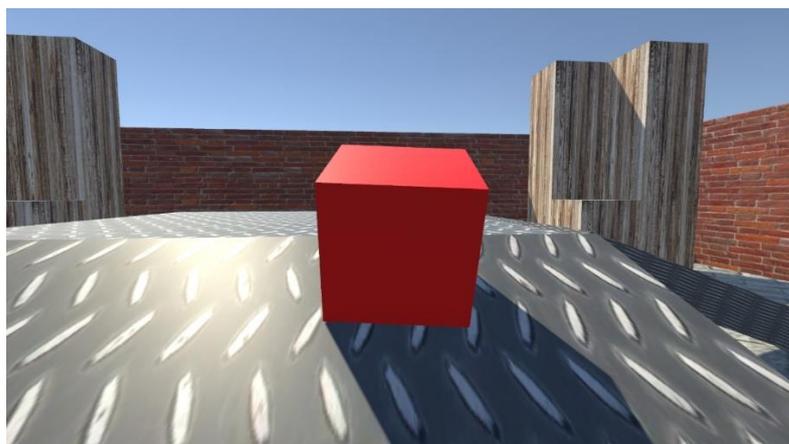


Рис.12 3D игровой мир

ПОЧАСОВОЕ ПЛАНИРОВАНИЕ ПРОГРАММЫ

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		всего	теория	практика	
1.	Модуль 1: «Знакомство интегрированной средой программирования Sharp Develop/Visual Studio Community. Основы языка программирования C#»	16	2	14	
1.1	Вводное занятие, знакомство с учениками, объяснение правил использования компьютерной техники и правил техники безопасности	1	1	0	Беседа, педагогическое наблюдение
1.2	Установка IDE Sharp Develop на компьютер. Русификация программы.	1	1	0	Проверка работоспособности редактора.
1.3	Знакомство с интегрированной средой разработки программного обеспечения Sharp Develop/Visual Studio Community. Настройка редактора, основные инструменты главного меню.	1	0	1	Практическое задание: настроить интегрированную среду разработки под себя. Выбрать размер шрифта, окон, подсветку синтаксиса кода и т.д.
1.4	Изучение строения минимальной программы на C#. структура программы, описание основных элементов кода. Знакомство с ключевыми словами языка C#.	1	0	1	Практическое задание: Пишем на C# первую программу «Hello World!!!»
1.5	Виды проектов C#. Особенности создания каждого проекта. Виды приложений на C#. Знакомство с консольным приложением. Понятие командной строки.	2	0	2	Практическое задание: Создать проект консольного приложения. Сохранение и открытие проекта. Работа с командной строкой. <u>Домашнее задание:</u> создать простейшую программу в консольном режиме. Программа спрашивает вас «Как вас зовут». Вы вводите свое имя, и программа отвечает вам: «Привет Имя! Как у тебя дела?»
1.6	Изучение переменных. Объявление переменных в C#. Основные типы данных. Понятие и применение автозаполнения кода в IDE Sharp Develop/Visual Studio. Последовательный алгоритм программы в действии. Организация ввода/вывода в C#.	2	0	2	Практическое задание: Пишем программу калькулятор в консольном приложении. Выполняем основные математические действия (сложение, вычитание, умножение, деление).

1.7	Ветвление алгоритмов в языке C#. Проверка логических условий. Понятие и значение булевых переменных в программировании.	2	0	2	Практическое задание: Создаем простейшую программу для вывода четных и нечетных чисел, с применением условия ветвления алгоритма.
1.8	Циклы в языке программирования C#. Основные виды циклов и их применение при создании приложений. Понятие счетчика цикла.	2	0	2	Практическое задание: Создаем программу для демонстрации работы циклов C#.
1.9	Основы структурного (модульного) программирования. Понятие функции в языке программирования на примере языка C#. Значение функций в структурировании программ.	2	0	2	Практическое задание: Создаем функции, каждая из которых выполняет основные математические действия (сложение, вычитание, умножение, деление).
1.10	Понятие параметров функций и возвращение результата функцией. Основы программирования параметризованных функций в C#.	1	0	1	Практическое задание: Создаем четыре параметризованные функции, выполняющие основные математические действия (сложение, вычитание, умножение, деление).
1.11	Оформление программ консольного типа. Работа с цветами, шрифтами, фонами консоли.	1	0	1	Практическое задание: Разрабатываем программу для демонстрации возможности консольного приложения. Домашнее задание: разработать и оформить программу в консольном режиме используя различные цвета, фоны и шрифты.
2.	Модуль 2: Основы программирования классических приложений Windows (оконные приложения Windows Form)	10	0	10	
2.1	Создание проекта Windows Form в Sharp Develop/Visual Studio Community. Создание проекта Windows приложения. Строение стандартного Windows приложения.	1	0	1	Практическое задание: Создаем простейшую программу, состоящую из Windows окна.
2.2	Изучение основных компонентов формы: button, label, textbox, Form, Panel, Image Box и т.д. Свойства компонентов формы. Задания и изменения свойств компонентов формы.	1	0	1	Практическое задание: Создаем программу, состоящую из формы с несколькими компонентами (button, label, picture Box). Меняем свойства компонентов, их расположение на форме.

2.3	Понятие обработчика событий компонента формы. Виды обработчиков событий и их свойства. Значение обработчика событий для формирования логики программы.	2	0	2	Практическое задание: Пишем обработчики событий для различных компонентов формы.
2.4	Мышь – как основной инструмент для работы с Windows приложениями. Понятие обработчика событий мыши. Понятие обработчика событий клавиатуры.	2	0	2	Практическое задание: Пишем обработчики событий для мыши и клавиатуры. Привязываем данные события к конкретным объектам формы.
2.5	Понятие многопоточности программы. Реализация многопоточности в Windows приложениях при помощи компонента Timer.	2	0	2	Практическое задание: Пишем программу, выполняющую одновременно два события при помощи компонента Timer
2.6	Краткий обзор других компонентов формы. Их применение при написании Windows приложений	2	0	2	Практическое задание: Изучить некоторые специфические свойства компонентов формы (прозрачность, стиль, цвет и т.д.)
3.	Модуль 3: Прикладное программирования на языке C#	8	0	8	
3.1	Введение в коллекции языка программирования C# (массивы, списки, стеки, словари)	2	0	2	Практическое задание: Создание программы «Грамматический словарь» для проверки знаний грамматики у учащихся. <u>Домашнее задание:</u> добавь в созданный словарь как можно больше слов. Придумай свою систему оценок (систему проверки).
3.2	Комплексное применение полученных знаний для создания полноценного Windows приложения	2	0	2	Практическое задание: Создаем шуточную программу «Вам нравится учиться в школе?» <u>Домашнее задание:</u> придумайте и реализуйте свой шуточный проект в Windows приложении.
3.3	Введение в компонент Picture Box. Создаем простейшие анимации в виде Windows приложений.	2	0	2	Практическое задание: Пишем программу, имитирующую полет космического корабля.

3.4	Комплексное использование полученных знаний для применения в программировании игр простой и средней сложности	2	0	2	Практическое задание: Создаем программу основу к знаменитой 2D игре «Зомби против растений» <u>Домашнее задание:</u> добавь ещё несколько компонентов в игровое поле.
4.	Модуль 4: Основы технологии программирования WPF	16	2	14	
4.1	Что такое WPF приложение. Примеры реальных программ, написанных с использованием данной технологии. Особенности построения WPF приложения.	2	2	0	Беседа, педагогическое наблюдение
4.2	Структура WPF приложения. Язык разметки XAML. Параметры главной формы WPF приложения.	2	0	2	Практическое задание: Разработка формы для классической игры «Крестики- Нолики»
4.3	Основные компоненты WPF приложения (Rectangle, Button, Textbox, Label, Slider и т.д.). Способы расположения компонентов на форме.	2	0	2	Практическое задание: Наполнение формы компонентами (интерфейсная часть игры «Крестики-Нолики»)
4.4	Свойства компонентов формы WPF приложения. Задание свойств через конструктор и через язык разметки XAML. Основные свойства компонентов формы.	2	0	2	Практическое задание: Подбор текстур, графических объектов. Привязка данных объектов и текстур к элементам формы
4.5	Работа с классами MediaPlayer и SoundPlayer. Использование медиа в приложениях WPF. Обработка исключений при открытии файла.	2	0	2	Практическое задание: Звуковое сопровождение игры «Крестики-Нолики»
4.6	Основы написания функциональной части WPF приложения. Понятие обработчика событий элементов формы.	2	0	2	Практическое задание: Написание логической части игры «Крестики -Нолики» (логика игрока)
4.7	Основы многопоточности. Класс Thread и Task. Использование многопоточности в WPF приложениях. Практическое значение многопоточности.	2	0	2	Практическое задание: Написание логической части игры «Крестики-Нолики» (логика компьютера)
4.8	Основы отладки готового приложения. Проверка корректной работы приложения. Средства отладки IDE Visual Studio.	2	0	2	Практическое задание: Тестирование готового проекта. Исправление ошибок. Отладка.

5.	Модуль 5: Основы Unity3D	20	2	18	
5.1	Знакомство с Unity3D. Место данного фреймворка и мировой игровой индустрии. Связь Unity с языком программирования C#	2	2	0	Беседа, педагогическое наблюдение
5.2	Порядок установки бесплатной версии фреймворка Unity3D. Системные требования. Выбор компонентов установки. Регистрация и создание учетной записи.	2	0	2	Практическое задание: Установка Unity3D на компьютер. Регистрируемся. Создаем учётную запись.
5.3	Виды проектов в Unity3D. Понятие кроссплатформенных приложений. Порядок создания проекта в Unity3D.	2	0	2	Практическое задание: Создаем первый проект 3D в Unity3D. Выбираем место сохранения проекта и название проекта. Изучаем состав созданного проекта.
5.4	Особенности интерфейса редактора Unity3D. Правила настройки редактора. Назначение окон редактора.	2	0	2	Практическое задание: Настраиваем редактор под себя. Изучаем расположение и функционал основных окон Unity3D.
5.5	Примитивы в Unity3D (куб, сфера, капсула, цилиндр, плоскость, квад). Основные свойства примитивов. Назначение и мощь стандартных примитивов.	2	0	2	Практическое задание: Добавляем на сцену примитивы. Используя примитив куба строим стены будущего лабиринта. Используя примитив плоскость добавляем на сцену пол.
5.6	Основы работы с текстурами в Unity3D. Добавление текстур в проект. Применение текстур к объекту сцены.	2	0	2	Практическое задание: Оживляем стены будущего лабиринта. Добавляем текстуру стен и пола. Устанавливаем цвет неба и силу освещения.
5.7	Добавление объектов на сцену. Размещение объектов в трехмерном пространстве. Особенности 3D пространства в Unity. Трехмерная система координат.	2	0	2	Практическое задание: Размещаем объекты на сцене. Практикуемся, изменяя координаты, размер и угол поворота объекта по различным осям. Привыкаем к 3D пространству.
5.8	Назначение языка программирования C# в Unity3D. Синхронизация работы редактора Unity3D и Visual Studio.	2	0	2	Практическое задание: Настраиваем редактор Unity3D для работы с Visual Studio.

5.9	Элементарные основы скриптинга в Unity3D. Правила написания скриптов. Добавление скрипта к объектам сцены.	2	0	2	Практическое задание: Оживляем наши объекты сцены с помощью скриптов. Добавляем управление объекта с помощью клавиш стрелок.
5.10	Основы работы с камерой в Unity3D. Компоненты и свойства камеры. Компиляция готового проекта. Получение исполняемого (exe) файла.	2	0	2	Практическое задание: Привязываем камеру к главному игровому объекту. Пишем скрипт для управления камерой. Тестируем готовый проект – перемещение главного героя по сцене. Сохраняем и компилируем проект в готовый файл.
6	Итоговое занятие	2	0	2	Презентация проектов
	Всего часов:	72	6	66	

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Месяц	Форма занятия	К-во часов	Тема занятия	Место проведения	Форма контроля
1.	Сентябрь	Лекция. Практическая работа	1	Вводное занятие, знакомство с учениками, объяснение правил использования компьютерной техники и правил техники безопасности	Кванториум	Беседа
		Лекция. Практическая работа	1	Установка IDE Sharp Develop на компьютер. Русификация программы.	Кванториум	Проверка работоспособности редактора
2.	Сентябрь	Практическая работа	1	Знакомство с интегрированной средой разработки программного обеспечения Sharp Develop. Настройка редактора, основные инструменты главного меню.	Кванториум	Практическое задание: настроить интегрированную среду разработки под себя. Выбрать размер шрифта, окон, подсветку синтаксиса кода и т.д.
		Практическая работа	1	Изучение строения минимальной программы на C#. структура программы, описание основных элементов кода. Знакомство с ключевыми словами языка C#	Кванториум	Практическое задание: Пишем на C# первую программу «Hello World!!!»
3.	Сентябрь	Практическая работа	2	Виды проектов C#. Особенности создания каждого проекта. Виды приложений на C#. Знакомство с консольным приложением. Понятие командной строки	Кванториум	Практическое задание: Создать проект консольного приложения. Сохранение и открытие проекта. Работа с командной строкой. <u>Домашнее задание:</u> создать простейшую программу в консольном режиме. Программа спрашивает вас «Как вас зовут». Вы вводите свое имя, и программа отвечает вам: «Привет Имя! Как у тебя дела?»
4.	Сентябрь	Практическая работа	2	Изучение переменных. Объявление переменных в C#. Основные типы данных. Понятие и применение автозаполнения кода в IDE Sharp Develop. Последовательный алгоритм программы в	Кванториум	Практическое задание: Пишем программу калькулятор в консольном приложении. Выполняем основные математические действия (сложение, вычитание, умножение, деление).

				действии. Организация ввода/вывода в C#		
5.	Октябрь	Практическая работа	2	Ветвление алгоритмов в языке C#. Проверка логических условий. Понятие и значение булевых переменных в программирование	Кванториум	Практическое задание: Создаем простейшую программу для вывода четных и нечетных чисел, с применением условия ветвления алгоритма
6.	Октябрь	Практическая работа	2	Циклы в языке программирования C#. Основные виды циклов и их применение при создании приложений. Понятие счетчика цикла	Кванториум	Практическое задание: Создаем программу для демонстрации работы циклов C#
7.	Октябрь	Практическая работа	2	Основы структурного (модульного) программирования. Понятие функции я языке программирования на примере языка C#. Значение функций в структурировании программ	Кванториум	Практическое задание: Создаем функции, каждая из которых выполняет основные математические действия (сложение, вычитание, умножение, деление).
8.	Октябрь	Практическая работа	1	Понятие параметров функций и возвращение результата функцией. Основы программирования параметризованных функций в C#	Кванториум	Практическое задание: Создаем четыре параметризованные функции, выполняющие основные математические действия (сложение, вычитание, умножение, деление).
		Практическая работа	1	Оформление программ консольного типа. Работа с цветами, шрифтами, фонами консоли	Кванториум	Практическое задание: Разрабатываем программу для демонстрации возможности консольного приложения. <u>Домашнее задание:</u> разработать и оформить программу в консольном режиме используя различные цвета, фоны и шрифты.
9.	Ноябрь	Практическая работа	1	Создание проекта Windows Form в Sharp Develop/Visual Studio Community. Создание проекта Windows приложения. Строение стандартного Windows приложения.	Кванториум	Практическое задание: Создаем простейшую программу, состоящую из Windows окна.
		Практическая работа	1	Изучение основных компонентов формы: button, label, textbox, Form, Panel, Image Box и т.д. Свойства компонентов формы.	Кванториум	Практическое задание: Создаем программу, состоящую из формы с несколькими компонентами (button, label,

				Задания и изменения свойств компонентов формы.		picture Box). Меняем свойства компонентов, их расположение на форме.
10.	Ноябрь	Практическая работа	2	Понятие обработчика событий компонента формы. Виды обработчиков событий и их свойства. Значение обработчика событий для формирования логики программы.	Кванториум	Практическое задание: Пишем обработчики событий для различных компонентов формы.
11.	Ноябрь	Практическая работа	2	Мышь – как основной инструмент для работы с Windows приложениями. Понятие обработчика событий мыши. Понятие обработчика событий клавиатуры.	Кванториум	Практическое задание: Пишем обработчики событий для мыши и клавиатуры. Привязываем данные события к конкретным объектам формы.
12.	Ноябрь	Практическая работа	2	Понятие многопоточности программы. Реализация многопоточности в Windows приложениях при помощи компонента Timer.	Кванториум	Практическое задание: Пишем программу, выполняющую одновременно два события при помощи компонента Timer
13.	Декабрь	Практическая работа	2	Краткий обзор других компонентов формы. Их применение при написании Windows приложений	Кванториум	Практическое задание: Изучить некоторые специфические свойства компонентов формы (прозрачность, стиль, цвет и т.д.)
14.	Декабрь	Практическая работа	2	Введение в коллекции языка программирования C# (массивы, списки, стеки, словари)	Кванториум	Практическое задание: Создание программы «Грамматический словарь» для проверки знаний грамматики у учащихся. <u>Домашнее задание:</u> добавь в созданный словарь как можно больше слов. Придумай свою систему оценок (систему проверки).
15.	Декабрь	Практическая работа	2	Комплексное применение полученных знаний для создания полноценного Windows приложения	Кванториум	Практическое задание: Создаем шуточную программу «Вам нравится учиться в школе?» <u>Домашнее задание:</u> придумайте и реализуйте свой шуточный проект в Windows приложении.

16.	Декабрь	Практическая работа	2	Введение в компонент Picture Box. Создаем простейшие анимации в виде Windows приложений.	Кванториум	Практическое задание: Пишем программу, имитирующую полет космического корабля.
17.	Январь	Практическая работа	2	Комплексное использование полученных знаний для применения в программировании игр простой и средней сложности	Кванториум	Практическое задание: Создаем программу основу к знаменитой 2D игре «Зомби против растений» <u>Домашнее задание:</u> добавь ещё несколько компонентов в игровое поле.
18.	Январь	Лекция	2	Что такое WPF приложение. Примеры реальных программ, написанных с использованием данной технологии. Особенности построения WPF приложения.	Кванториум	Беседа, педагогическое наблюдение
19.	Январь	Практическая работа	2	Структура WPF приложения. Язык разметки XAML. Параметры главной формы WPF приложения.	Кванториум	Практическое задание: Разработка формы для классической игры «Крестики- Нолики»
20.	Январь	Практическая работа	2	Основные компоненты WPF приложения (Rectangle, Button, Textbox, Label, Slider и т.д.). Способы расположения компонентов на форме.	Кванториум	Практическое задание: Наполнение формы компонентами (интерфейсная часть игры «Крестики-Нолики»)
21.	Февраль	Практическая работа	2	Свойства компонентов формы WPF приложения. Задание свойств через конструктор и через язык разметки XAML. Основные свойства компонентов формы.	Кванториум	Практическое задание: Подбор текстур, графических объектов. Привязка данных объектов и текстур к элементам формы
22.	Февраль	Практическая работа	2	Работа с классами MediaPlayer и SoundPlayer. Использование медиа в приложениях WPF. Обработка исключений при открытии файла.	Кванториум	Практическое задание: Звуковое сопровождение игры «Крестики-Нолики»
23.	Февраль	Практическая работа	2	Основы написания функциональной части WPF приложения. Понятие обработчика событий элементов формы.	Кванториум	Практическое задание: Написание логической части игры «Крестики - Нолики» (логика игрока)
24.	Февраль	Практическая работа	2	Основы многопоточности. Класс Thread и Task. Использование многопоточности в	Кванториум	Практическое задание: Написание логической части игры «Крестики-

				WPF приложениях. Практическое значение многопоточности.		Нолики» (логика компьютера)
25.	Март	Практическая работа	2	Основы отладки готового приложения. Проверка корректной работы приложения. Средства отладки IDE Visual Studio.	Кванториум	Практическое задание: Тестирование готового проекта. Исправление ошибок. Отладка.
26.	Март	Лекция	2	Знакомство с Unity3D. Место данного фреймворка и мировой игровой индустрии. Связь Unity с языком программирования C#	Кванториум	Беседа, педагогическое наблюдение
27.	Март	Практическая работа	2	Порядок установки бесплатной версии фреймворка Unity3D. Системные требования. Выбор компонентов установки. Регистрация и создание учетной записи.	Кванториум	Практическое задание: Установка Unity3D на компьютер. Регистрируемся. Создаем учётную запись.
28.	Март	Практическая работа	2	Виды проектов в Unity3D. Понятие кроссплатформенных приложений. Порядок создания проекта в Unity3D.	Кванториум	Практическое задание: Создаем первый проект 3D в Unity3D. Выбираем место сохранения проекта и название проекта. Изучаем состав созданного проекта.
29.	Апрель	Практическая работа	2	Особенности интерфейса редактора Unity3D. Правила настройки редактора. Назначение окон редактора.	Кванториум	Практическое задание: Настраиваем редактор под себя. Изучаем расположение и функционал основных окон Unity3D.
30.	Апрель	Практическая работа	2	Примитивы в Unity3D (куб, сфера, капсула, цилиндр, плоскость, квад). Основные свойства примитивов. Назначение и мощь стандартных примитивов.	Кванториум	Практическое задание: Добавляем на сцену примитивы. Используя примитив куба строим стены будущего лабиринта. Используя примитив плоскость добавляем на сцену пол.
31.	Апрель	Практическая работа	2	Основы работы с текстурами в Unity3D. Добавление текстур в проект. Применение текстур к объекту сцены.	Кванториум	Практическое задание: Оживляем стены будущего лабиринта. Добавляем текстуру стен и пола. Устанавливаем цвет неба и силу освещения.

32.	Апрель	Практическая работа	2	Добавление объектов на сцену. Размещение объектов в трехмерном пространстве. Особенности 3D пространства в Unity. Трехмерная система координат.	Кванториум	Практическое задание: Размещаем объекты на сцене. Практикуемся, изменяя координаты, размер и угол поворота объекта по различным осям. Привыкаем к 3D пространству.
33.	Май	Практическая работа	2	Назначение языка программирования C# в Unity3D. Синхронизация работы редактора Unity3D и Visual Studio.	Кванториум	Практическое задание: Настраиваем редактор Unity3D для работы с Visual Studio.
34.	Май	Практическая работа	2	Элементарные основы скриптинга в Unity3D. Правила написания скриптов. Добавление скрипта к объектам сцены.	Кванториум	Практическое задание: Оживляем наши объекты сцены с помощью скриптов. Добавляем управление объекта с помощью клавиш стрелок.
35.	Май	Практическая работа	2	Основы работы с камерой в Unity3D. Компоненты и свойства камеры. Компиляция готового проекта. Получение исполняемого (exe) файла.	Кванториум	Практическое задание: Привязываем камеру к главному игровому объекту. Пишем скрипт для управления камерой. Тестируем готовый проект – перемещение главного героя по сцене. Сохраняем и компилируем проект в готовый файл.
36.	Май	Итоговое занятие	2	Презентация и защита проектов	Кванториум	Защита проекта

ВОСПИТАТЕЛЬНАЯ РАБОТА

Воспитательная работа в рамках реализации настоящей программы строится в соответствии с программой развития общекультурных компетенций обучающихся структурного подразделения детский технопарк «Кванториум», под которыми понимают способности ребенка ориентироваться в пространстве культуры, а именно:

- способность ориентироваться в первоисточниках культуры (произведениях литературного, музыкального, изобразительного, театрального искусства, музейных экспозициях) в целях максимального погружения в проблему и поиска оптимального пути ее решения;

- способность ориентироваться в источниках информации, отбирать адекватные источники для выполнения познавательных задач, постижения и построения научной картины мира;

- способность объяснять явления действительности, с которыми сталкиваются люди в повседневной жизни, с позиций науки и техники;

- способность ориентироваться в актуальных проблемах общественной жизни, определять причины их возникновения, характеризовать и обосновывать мнения о путях их решения с проекцией на собственную деятельность;

- способность ориентироваться в мире социальных, нравственных и эстетических ценностей: уметь различать факты, суждения, оценки, устанавливать их связь с определенной системой ценностей, определять собственное аксиологическое поле.

Воспитательная работа организуется в рамках 5 тематических модулей, а именно:

- модуль «Учебное занятие», который является основным и направлен на формирование технологической грамотности обучающихся в контексте реализации содержания программы;

- модуль «Ключевые образовательные события», который предполагает проведение воспитательных мероприятий преимущественно в форматах мастер-классов, квестов, конструкторских лабораторий, являющихся своеобразными профессиональными пробами, позволяющими обучающимся познакомиться с образовательными возможностями определенного оборудования, программного обеспечения, цифрового сервиса или платформы в практической деятельности;

- модуль «Проектная деятельность», который предусматривает выбор, разработку, реализацию и защиту итогового проекта, которые в дальнейшем могут быть представлены за пределами детского технопарка «Кванториум» (на муниципальной неделе школьных наук, региональных и межрегиональных научно-практических конференциях и соревнованиях, таких как «Шаг в науку» и «Шаг в будущее», отборочных этапах Всероссийских технологических конкурсов и хакатонов «Большие вызовы», «Инженерные кадры России», «Первому разработчику приготовиться» и т.п.);

- модуль «Социальные инициативы», который направлен на вовлечение обучающихся в различные формы шефства и наставничества, волонтерской и самоуправленческой деятельности в ходе проведения учебных занятий и образовательных событий, подготовке индивидуальных и групповых проектов, участии в конкурсных мероприятиях разного уровня;

- модуль «Инфо-контент», который ориентирован на подбор информации, способствующей формированию целостной картины мира, развитию практических компетенций по использованию технического оборудования и цифровых лабораторий, расширению представлений об актуальных на сегодняшний день профессиях посредством ее размещения на каналах и видеохостингах детского технопарка «Кванториум».

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение:

- учебный кабинет;
- ноутбуки на каждого обучающегося и преподавателя с доступом к сети Интернет;
- проекционное оборудование (экраны);
- маркерная доска.

Информационное обеспечение:

- интегрированная среда разработки Sharp Develop или Visual Studio Community;
- графический редактор Point;
- платформа «Сферум».

Кадровое обеспечение:

Программу реализует педагог дополнительного образования.

Педагог умеет:

- учитывать уровень знаний учащихся при постановке задач;
- оказывать поддержку в поиске различных видов источников информации для решения той или иной задачи;
- помогать формировать образовательный маршрут, если это вызывает трудности у самого учащегося

ФОРМЫ АТТЕСТАЦИИ И ТЕКУЩЕГО КОНТРОЛЯ

Текущая аттестация проводится в форме выполнения практических заданий, итоговая аттестация предусматривает выполнение индивидуальных и (или) групповых проектов по пройденному материалу. Отметочная форма контроля отсутствует.

Оценивание развития учащихся проводится на основе следующего перечня компетенций:

Технические: инженерно-пространственное, конструкторское, алгоритмическое и логическое мышление.

Гибкие: творческое мышление, умение работать в коллективе, эффективная коммуникация, контроль эмоционально-волевой сферы.

Текущий контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов на нескольких уровнях: на каждом занятии: беседа с учениками, постановка задачи, совместное обсуждение и планирование будущего мини проекта, выполнение заданий, самоконтроль ученика; выполнение поставленных задач, взаимоконтроль учеников, мини соревнования.

Показатели выполнения практических заданий:

- решают практические задачи по образцу, следуя прямым указаниям педагога;
- умеют выполнять задания, внося изменения в образец, манипулируя изученным материалом, но обращаются за помощью к педагогу;
- самостоятельно формируют алгоритм, применяя все ранее изученные алгоритмические конструкции;
- применяют творческие способности для разработки собственных проектов;
- умеют находить, подбирать, адаптировать объекты, необходимые для создания собственного проекта.

Критерии оценивания выполнения практических заданий:

Критерий оценивания практических заданий не носит бальный характер, поскольку обучающиеся только вступают в огромный мир программирования и система оценок может ослабить их мотивацию к дальнейшему обучению этой дисциплины.

Педагог должен индивидуально оценить способность каждого обучающегося и в случае необходимости уделить больше времени тому ученику, у которого процесс освоения дисциплины более трудный нежели у остальных учеников.

В любом случае программирование – это творческий процесс, и работу обучающегося следует оценивать исходя из его заинтересованности в данной дисциплине, отношению к ней, степени ответственности при выполнении задания и потенциала самого ученика.

Бальную систему оценки целесообразно применить лишь при презентации итогового проекта (система оценивания описана выше).

Оценка защиты проекта осуществляется по **накопительной системе** в соответствии со следующей таблицей:

№	Виды работ	Оценка в баллах	Кто оценивает
1	Презентация проекта. Умение обосновать выбранную тему или жанр (например игры)	0-10	Преподаватель
2	Сложность модели проекта (количество фонов, спрайтов, звуков и т.д.)	0 - 10	Преподаватель
3	Сложность программы (наличие алгоритмов, разветвлений, циклов, переходов, число скриптов и т.д.)	0-10	Преподаватель
4.	Бесперебойность в работе программы (отсутствие сбоев при нештатных ситуациях)	0-10	Преподаватель
5.	Оригинальность сценария (авторство)	0-10	Преподаватель
	ИТОГО:	50 баллов	

Результаты освоения программы (высокий, средний и низкий уровни)

Высокий уровень освоения программы	Учащийся демонстрирует высокую заинтересованность в учебной и творческой деятельности, которая является содержанием программы; показывает широкие возможности практического применения в собственной творческой деятельности приобретенных знаний умений и навыков, проявляет самостоятельность и высокий уровень готового продукта (практические задания, проекты и т.д.)
Средний уровень освоения программы	Учащийся демонстрирует достаточную заинтересованность в учебной и творческой деятельности, которая является содержанием программы; может применять на практике в собственной творческой деятельности приобретенные знания умения и навыки, выполнение работ под контролем или небольшой помощью педагога.
Низкий уровень освоения программы	Учащийся демонстрирует слабую заинтересованность в учебной и творческой деятельности, которая является содержанием программы; не стремится самостоятельно применять на практике в своей деятельности приобретенные знания умения и навыки, работы выполняет с помощью педагога.

СПИСОК ЛИТЕРАТУРЫ

1. Банкрашков, А.В. Программирование для детей на языке С# /А.В. Банкрашков. - М.: АСТ, 2018. - 288 с.
2. Блиновская, Я.Ю. Введение в информатику: Учебное пособие /Я.Ю. Блиновская, Д.С. Задоя. - М.: Форум, НИЦ ИНФРА-М, 2016. - 112 с.
3. Бриггс, Джейсон С# для детей. Самоучитель по программированию / Джейсон Бриггс. - Москва: Огни, 2019. - 177 с.
4. Бэрри, Пол. Изучаем программирование на С# / Пол Бэрри. - М.: Эксмо, 2017. - 332 с.
5. Васильев, А. Н. С# на примерах. Практический курс по программированию / А.Н. Васильев. - М.: Наука и техника, 2017. - 432 с.
6. Васильев, Александр Николаевич С# на примерах. Практический курс по программированию. Руководство / Васильев Александр Николаевич. - М.: Наука и техника, 2018. - 752 с.
7. Велихов, А. С. Основы информатики и компьютерной техники: учебное пособие / А. С. Велихов. – Москва: СОЛОН-Пресс, 2017. – 539 с.
8. Вордерман, К. Программирование на С#. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. - М.: Манн, Иванов и Фербер, 2017. - 346 с.
9. Гуриков, С.Р. Основы алгоритмизации и программирования на С#. Учебное пособие. Гриф МО РФ / С.Р. Гуриков. - М.: Инфра-М, Форум, 2020. - 707 с.
10. МакГрат, Майк С#. Программирование для начинающих / Майк МакГрат. - М.: Эксмо, 2019. - 727 с.
11. Саммерфилд, М. Программирование на С# 3. Подробное руководство / М. Саммерфилд. - М.: Символ, 2016. - 608 с.
12. Семакин И.Г., Залогова Л.А. Информатика: Базовый курс 7-9 кл. / И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова – М.: Бином. Лаборатория Знаний, 2020.
13. Семакина И.Г., Хеннера Е.К. Информатика. Задачник-практикум: в 2т./ Под ред. И.Г.Семакина, Е.К. Хеннера: Т.1. М.: БИНОМ. Лаборатория базовых знаний, 2017.