

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЛЕСОСИБИРСКИЙ ПЕДАГОГИЧЕСКИЙ ИНСТИТУТ –  
филиал Сибирского федерального университета

Высшей математики, информатики, экономики и естествознания  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

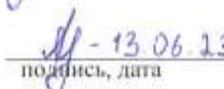
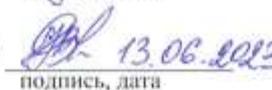
 Л.Н. Храмова  
подпись      инициалы, фамилия

« 13 » 06 2023 г.

### БАКАЛАВРСКАЯ РАБОТА

44.03.05 Педагогическое образование (с двумя профилями подготовки)  
код-наименование направления

ТЕХНОЛОГИИ ГЕЙМИФИКАЦИИ В ФОРМИРОВАНИИ ИГРОВОЙ  
СРЕДЫ ДЛЯ ОБУЧЕНИЯ ОСНОВАМ ПРОГРАММИРОВАНИЯ НА  
PYTHON НА УРОКАХ ИНФОРМАТИКИ В 9 КЛАССЕ

Руководитель	 13.06.2023 подпись, дата	доцент, канд. пед. наук должность, ученая степень	Е. В. Киргизова инициалы, фамилия
Выпускник	 13.06.23 подпись, дата		А. П. Зернин инициалы, фамилия
Нормоконтролер	 13.06.2023 подпись, дата		А. В. Фирер инициалы, фамилия

Лесосибирск 2023

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Технологии геймификации в формировании игровой среды для обучения основам программирования на Python на уроках информатики в 9 классе» содержит 68 страницы текстового документа, 43 использованных источника, 13 таблиц, 13 рисунков, 1 приложение.

ТЕХНОЛОГИЯ ГЕЙМИФИКАЦИИ, ИГРОВАЯ СРЕДА, ПРОГРАММИРОВАНИЕ, МЕТОДИКА ОБУЧЕНИЯ ИНФОРМАТИКЕ.

В условиях развития информационного общества происходит трансформация образовательных подходов: уходят в прошлое традиционные подходы, основанные на трансляции знания и появляются новые подходы, основанные на использовании информационно-коммуникационных технологий, в том числе и игрового антуража (геймификации).

Цель исследования – теоретически обосновать и разработать методическое сопровождение игровой среды, реализованной на основе технологии геймификации, для изучения основ языка программирования Python в 9 классе.

Основные задачи исследования:

– провести анализ научной и учебно-методической литературы по теме исследования, определить понятийно-категориальный аппарат, сущность технологии геймификации;

– разработать методическое сопровождение игровой среды, реализованной на основе технологии геймификации, для изучения основ языка программирования Python в 9 классе;

– организовать и провести опытно-экспериментальную работу по применению игровой среды на основе технологии геймификации для обучения основам языка программирования Python на уроках информатики в 9 классе.

В результате исследования рассмотрено применение технологии геймификации в формировании игровой среды и на её основе разработан электронный ресурс для обучения основам языка программирования Python.

## СОДЕРЖАНИЕ

Введение.....	4
1 Теоретические аспекты применения технологии геймификации в формировании игровой среды для обучения основам программирования .....	10
1.1 Сущность технологии геймификации и её использование в обучении информатике .....	10
1.2 Игровые среды и возможности их применения для изучения основ программирования .....	19
1.3 Дидактический потенциал технологии геймификации в формировании игровой среды при обучении основам программирования .....	23
2 Реализация технологии геймификации в формировании игровой среды для обучения основам программирования в 9 классе .....	33
2.1 Анализ содержания линии «Алгоритмизация и программирование» в школьном курсе информатики.....	33
2.2 Разработка учебно-методического сопровождения для изучения основ языка программирования Python с использованием игровой среды .....	42
2.3 Результаты опытно-экспериментальной работы по организации обучения основам языка программирования Python с использованием игровой среды, реализованной на основе технологии геймификации.....	53
Заключение .....	59
Список использованных источников .....	61
Приложение А Тестовые задания для проведения контрольного этапа опытно-экспериментальной работы .....	67

## ВВЕДЕНИЕ

Современный мир, развиваясь в направлении информационных технологий, заставляет общество и школьников приобретать новые знания и использовать новые методы обучения. Федеральный проект «Цифровая образовательная среда» предусматривает изменения образовательных подходов в связи с цифровизацией общества. Традиционные подходы, основанные на передаче знаний, уходят в прошлое, а на их место приходят новые методы, основанные на использовании информационно-коммуникационных технологий и игрового антуража (геймификации).

В настоящее время особое внимание уделяется рассмотрению вопросов, направленных на оценку возможности применения различных игровых сред в процессе обучения. Некоторые разделы школьного курса информатики можно эффективно изучать с применением технологий геймификации в ходе организации процесса обучения на уроках информатики [16].

В современном образовательном пространстве преподавание информатики является одной из ключевых задач, которая требует внимания и комплексного подхода. Для эффективного решения этой задачи необходимо учитывать общие методические проблемы, которые возникают при преподавании данного предмета. Представителями, занимающимися анализом обучения информатике в контексте общих методических проблем, являются Л. Л. Босова [6], Г. А. Краснова [16], М. П. Лапчик [17], К. Ю. Поляков [29], И. Г. Семакин [35], А. Г. Сиденко [36] и другие ученые. Их работы отражают потребность в разработке новых подходов к преподаванию курса информатики, а также в оценке эффективности существующих методик.

Вопросам применения геймификации в образовании посвящены работы С. С. Белоконовой [1], Н. Г. Борисова [4], К. М. Каппа [15], А. Л. Мазелиса [19], О. В. Орловой [26], М. В. Эйхорна [40] и других.

Актуальность исследования технологии геймификации в образовании информатики проявляется в существующем противоречии между требованием

увеличения эффективности обучения с помощью средств информатизации образования и недостаточностью проработки методологии и подходов в обучении с использованием технологии геймификации. Для решения данного противоречия необходимо внимательно изучить методы применения геймификации при создании игровой среды для процесса обучения информатике. Введение технологии геймификации в образовательный процесс может привести к улучшению мотивации и усвоению материала учащимися. Поэтому поиск устранения данного противоречия свидетельствует об актуальности выбранной темы исследования.

Цель исследования – теоретически обосновать и разработать методическое сопровождение игровой среды, реализованной на основе технологии геймификации, для изучения основ языка программирования Python в 9 классе.

Объект исследования – процесс обучения информатике в основной школе.

Предмет исследования – технологии геймификации в формировании игровой среды для обучения основам языка программирования Python в 9 классе.

Для достижения поставленной перед нами цели необходимо решить следующие задачи исследования:

- провести анализ научной и учебно-методической литературы по теме исследования, определить понятийно-категориальный аппарат, сущность технологии геймификации;

- разработать методическое сопровождение теоретической и практической составляющих игровой среды, реализованной на основе технологии геймификации, для изучения основ языка программирования Python в 9 классе;

- организовать и провести опытно-экспериментальную работу по применению игровой среды на основе технологии геймификации для обучения основам языка программирования Python на уроках информатики в 9 классе.

Теоретико-методологическую основу исследования составили:

– технология геймификации (С. Детергин [42], К. Вербах [7], К. Н. Мацуца [20], С. И. Никитин [23], П. В. Храмкин [39], М. В. Эйхорн [40] и др.), как новый способ организации процесса обучения;

– научные исследования ученых в области применения геймификации в процессе обучения (С. С. Белоконова [1], Н. Г. Борисов [4], К. М. Капп [15], А. Л. Мазелис [19], О. В. Орлова [26], М. В. Эйхорн [40]);

– игровые среды (Е. К. Герасимова [8], Г. А. Краснова [16], В. В. Певзнер [28]), как средство для изучения основ программирования;

– теоретико-методические основы изучения общих методических проблем обучения информатике в основной школе (Л. Л. Босова [6], Г. А. Краснова [16], М. П. Лапчик [17], К. Ю. Поляков [29], И. Г. Семакин [35], А. Г. Сиденко [36]);

– программирование (Л. Л. Босова [5], А. П. Ершов [10], М. П. Лапчик [17], К. Ю. Поляков [29], И. Г. Семакин [35] и Н. Д. Угринович [38]), как один из ключевых раздела предмета «Информатики и ИКТ».

Методы исследования:

– теоретические – анализ, синтез отечественной и зарубежной научной и учебно-методической литературы по проблеме исследования, изучение нормативных документов в сфере образования и обучения информатике;

– эмпирические – педагогическое наблюдение, тестирование, опытно-экспериментальное исследование;

– статистические – количественный и качественный анализ данных, интерпретация результатов.

Экспериментальная база исследования: Муниципальное бюджетное образовательное учреждение «Средняя общеобразовательная школа № 1 города Лесосибирска» (далее МБОУ «СОШ № 1»). В апробации приняли участие ученики 9 классов, объем выборки – 9А (20 человек) и 9Б (20 человек).

Исследование проводилось в соответствии со следующими этапами:

– констатирующий этап (сентябрь 2023 – январь 2023) – анализ научных источников по теме исследования; определение понятийно-категориального аппарата и методологической основы исследования; обоснование актуальности темы исследования, степени ее научной разработанности; постановка цели, определение объекта, предмета и задач исследования; выбор наиболее подходящих методов исследования, направленных на достижение заявленной цели;

– формирующий этап (февраль 2023 – апрель 2023) – теоретическая разработка методического сопровождения и апробация практической составляющей игровой среды для изучения основ языка программирования Python в 9 классе, реализованной на основе технологии геймификации; формирование системы критериев и показателей для контроля степени обученности класса;

– контрольный этап (май 2023 – июнь 2023) – анализ и обобщение результатов опытно-экспериментальной работы по обучению основам языка программирования Python с использованием игровой среды, реализованной на основе технологии геймификации, подведение итогов исследования.

Практическая значимость исследования:

– разработано и внедрено в учебный процесс по информатике в основной школе методическое сопровождение игровой среды, реализованной на основе технологии геймификации, для изучения основ языка программирования Python в 9 классе;

– разработанное методическое сопровождение игровой среды для изучения основ языка программирования Python в 9 классе – может быть использовано учителями информатики в своей педагогической деятельности, а также студентами – будущими педагогами, при подготовке к практическим занятиям, прохождению практики и написании научных работ.

Результаты исследования докладывались и обсуждались на конференциях:

1. Всероссийский молодежный научный форум «Современное педагогическое образование: теоретический и прикладной аспекты» в секции «Актуальные проблемы преподавания дисциплин естественнонаучного цикла» (Лесосибирск, ЛПИ – филиал СФУ, 7–12 ноября 2022 г., диплом 1 степени).

2. Международная научно-практическая конференция памяти академика РАО М. П. Лапчика «Информатизация образования: теория и практика» (Омск, ОмГПУ, 18–19 ноября 2022 г., участник).

3. Международная научно-практическая конференция «Исследования в современной науке» (Краснодар, КЦНТИ, 30 марта 2023 г., участник).

4. II Всероссийский молодежный научный форум «Современное педагогическое образование: теоретический и прикладной аспекты» в секции «Информатика, информационные технологии и экономика» (Лесосибирск, ЛПИ – филиал СФУ, 10–15 апреля 2023 г., участник).

5. Международная научно-практическая интернет-конференция «Актуальные проблемы методики обучения информатике и математике в современной школе» (Москва, МПГУ, 24–28 апреля 2023 г., участник).

6. VIII Всероссийская научно-практическая конференции с международным участием «Педагогика и психология: проблемы развития мышления. Развитие личности в изменяющихся условиях» (Красноярск, СибГУ им. М. Ф. Решетнева, 31 мая 2023 г., участник).

По результатам исследования опубликованы статьи в сборниках конференций:

1. Зернин, А. П. Технологии геймификации в обучении основам программирования на Python / Современное педагогическое образование : теоретический и прикладной аспекты : сборник научных статей Всероссийского молодежного научного форума (7–12 ноября 2022 г.) / Сибирский федеральный университет, Лесосибирский педагогический институт – филиал СФУ. – Красноярск, 2022. – С. 23–27.

2. Зернин, А. П. Использование веб-сервисов геймификации в организации образовательного процесса по информатике / А. П. Зернин,

Е. В. Киргизова // Информатизация образования : теория и практика : сборник материалов Международной научно-практической конференции памяти академика РАО М. П. Лапчика (Омск, 18–19 ноября 2022 г.) / Омский Государственный педагогический университет. – Омск, 2022. – С. 261–263.

3. Зернин, А. П. Использование технологии геймификации в обучении основам программирования на уроках информатики / А. П. Зернин, Е. В. Киргизова // Исследования в современной науке : материалы Международной научно-практической конференции (Краснодар, 30 марта 2023 г.) / Краснодарский ЦНТИ. – Краснодар, 2023. – С. 81–85.

Структура работы – работа состоит из введения, двух глав, заключения, списка использованных источников, включающего 43 наименования. Результаты работы представлены в 13 таблицах, 13 рисунках. В 1 приложении представлены материалы диагностики. Общий объем работы – 68 печатных листов.

# **1 Теоретические аспекты применения технологии геймификации в формировании игровой среды для обучения основам программирования**

## **1.1 Сущность технологии геймификации и её использование в обучении информатике**

В настоящее время возрастает значимость проблемы, связанной с качеством образования. Следовательно, необходимо проведение поиска нестандартных подходов для повышения эффективности управления образовательным процессом. Для решения данного вопроса, требуется разработка и внедрение новых методов и практик в образовательные учреждения.

В распоряжении Министерства просвещения РФ от 18.05.20 № Р-44 [32], геймификация учения рассматривается как возможное решение для преодоления трудностей индустриальной системы образования. Геймификация представляет собой процесс включения цифровых игровых форм в процессы формирования компетенций обучающихся и их мотивации. Этот метод активно используется в передовом российском и международном опыте в качестве одного из ключевых образовательных трендов. Использование геймификации в организации учебного процесса является новейшей технологией, позволяющей решать проблемы традиционной системы образования [32].

С древних времен игровые элементы были включены в образовательный процесс для формирования и развития самоконтроля, нестандартного мышления и выявления индивидуального творческого потенциала. В настоящее время игровая деятельность успешно внедряется в учебный процесс, поскольку игра является уникальной средой для обучения, предоставляющей широкий спектр вариантов «ходов» и дающей возможность ошибаться. Один из основных преимуществ игровой деятельности в обучении состоит в том, что игра способствует развитию концентрации внимания и повышению мотивации участников. Ученики часто становятся более заинтересованными в теме, когда они могут получать знания через игру [26].

В настоящее время геймификация является одной из новых технологий, которая пока недостаточно изучена, но имеет широкий потенциал в образовании. Последние десять лет геймификация рассматривается как один из наиболее перспективных инновационных подходов к проведению занятий в школе. Его применение позволяет создавать интересные и увлекательные задания для детей, которые заинтересованы в играх и охотно принимают участие в них. Геймификация обогащает образовательный процесс, создает более эффективную и интерактивную атмосферу на уроках, помогает улучшить мотивацию учеников и снизить уровень стресса. В связи с этим, доступность геймификации и ее использование в качестве образовательного инструмента должно получить более глубокое изучение, чтобы эффективно интегрировать в образовательный процесс и реализовывать все эти преимущества на практике.

Термин «геймификация» был введен в 2003 году британским программистом Ником Пеллингом. Он предложил вводить в программные инструменты элементы игры и использовать их в сферах, не относящихся к играм [23].

Термин «геймификация» в контексте образования появился недавно, благодаря немецкому исследователю С. Детердингу [42], который рассматривал геймификацию как применение элементов игры в неигровом контексте. Для педагогического процесса геймификация – это средство образовательной деятельности, которое включает в себя игру, имеющую свое содержание [42].

Чтобы понять место геймификации в образовательном процессе, нужно обратиться к схеме С. Детердинга [42], представленной на рисунке 1, которая отражает место геймификации в образовательном процессе.



Рисунок 1 – Схема отражения места геймификации в образовательном процессе

Игровые практики могут нести за собой цель, не связанную с самой игрой. «Организация» и «Содержание» являются главными осями, между которыми проходит вся игровая практика. Игра относится к содержанию, в то время как геймификация – к организации. Использование педагогом игры, как учебного инструмента, приводит к остановке учебного процесса, который возобновляется после проведения игры. Геймификация, в отличие от игры, не препятствует образовательному процессу, а наоборот, мотивирует учащихся к активности, позволяет параллельно вести процесс обучения и достигать образовательных целей [42].

Геймификация в обучении – это использование игровых правил, которые применяются в онлайн-играх, с целью мотивировать учащихся и достигать реальных образовательных целей [26].

В педагогической литературе рассмотрено большое количество определений понятия «геймификация», представленных на рисунке 2.

К. Вербах [7]	<ul style="list-style-type: none"> <li>• применение приемов, характерных для компьютерных игр, в неигровых процессах. Геймификация делает скучные задания интересными, избегаемое – желанным, а сложное – простым</li> </ul>
К. М. Капп [15]	<ul style="list-style-type: none"> <li>• использование игровых механик, эстетика и игровое мышление для вовлечения людей, мотивируют действия, способствуют обучению и решают проблемы</li> </ul>
А. Л. Мазелис [19]	<ul style="list-style-type: none"> <li>• применение элементов игры и игровых принципов в неигровых контекстах</li> </ul>
К. Н. Мацуца [20]	<ul style="list-style-type: none"> <li>• адаптация и повсеместное использование (видео) игр в повседневной жизни</li> </ul>
С. И. Никитин [23]	<ul style="list-style-type: none"> <li>• применение элементов дизайна игры в неигровом контексте</li> </ul>
П. В. Храбкин [39]	<ul style="list-style-type: none"> <li>• использование игровых правил, используемых в современных онлайн-играх, для мотивации учащихся и достижения реальных образовательных целей в курсе изучения учебного предмета</li> </ul>
М. В. Эйхорн [40]	<ul style="list-style-type: none"> <li>• применение игровых методик в неигровых ситуациях</li> </ul>
С. Детердинг [42]	<ul style="list-style-type: none"> <li>• применение элементов игры в неигровом контексте</li> </ul>

Рисунок 2 – Поле мнений на понятие геймификации

Анализируя данные определения, раскроем сущность термина «геймификация». Элементы игры – это те ее части, которые можно выделить и с которыми можно работать в дальнейшем (например, правила, очки, немедленная обратная связь, уровни, знаки достижений и т.д.). Как утверждает Кевин Вербах, участник проекта онлайн-образования Coursera, определяет три категории игровых элементов: динамики, механики и компоненты геймификации в этом случае заключается в том, что возникает возможность заново использовать эти элементы при создании новых работ, не имеющих прямого отношения к играм. Под неигровыми контекстами понимается цель, отличная от задачи достижения успеха в самой игре [7].

С. С. Белоконова и В. В. Назарова [1], рассматривая возможность применения игровых технологии в современном образовательном процессе, выделяют основную цель геймификации – организовать учебную деятельность посредством игры, т.е. «геймифицировать» образовательный процесс, мотивировать обучающихся на своевременное и качественное выполнение заданий игры и сформировать у них ощущение прогресса [1].

В настоящее время, геймификация признана инновационной технологией, способной преодолеть разрыв между поколениями учителей и учеников. Главным преимуществом геймификации является ее универсальность как вспомогательной учебной деятельностью на занятиях, в домашних заданиях, итоговых тестах, так и в качестве мотивации учеников и повышения качества обучения путем совершенствования их навыков. Общеизвестным фактом является тот факт, что геймификация может повысить эффективность учебного процесса и способствовать включению современных средств обучения в образовательную практику. В свою очередь, применение геймификации среди учеников способствует повышению интереса к учебному процессу и формированию позитивного настроения [12; 16].

Г. А. Краснова [16] определяет геймификацию, как «подход, предусматривающий применение в процессе обучения принципов <...> игровых сценариев и динамик в неигровых ситуациях с целью повышения

мотивации, вовлеченности в процесс решения учебных задач и достижения поставленных целей». И выделяет следующие формы геймификации:

Тяжелая геймификация – полная погруженность в мир игры с четко ограниченным временем, игровым пространством, ролями, сюжетной историей для каждого игрового персонажа;

Легкая геймификация, её суть заключается в заимствование игровых элементов в учебном и их применении в учебном или организационном процессе. Она помогает повышать уровень мотивации учащихся, индивидуализировать обучение и делать учебный процесс более интересным и увлекательным. В качестве примера можно привести виртуальные баллы, которые присваиваются детям за выполнение заданий и активное участие в выполнении заданий. Эти баллы не имеют реальной стоимости, но служат мотивационным фактором и помогают стимулировать учебную активность [16].

Интересный обзор зарубежной практики геймификации обучения содержится в монографии К. М. Каппа [15], в которой выделены два основных вида геймификации.

Структурная геймификация направлена на перемещение обучаемого по содержанию уже изученного материала без изменения его содержания. Этот подход подходит для научных дисциплин, где предполагается наличие большого количества практических заданий, которые можно оценить и ранжировать. При таком подходе дети могут работать вместе над выполнением задач, что способствует обмену знаниями и совместному усвоению материала.

Содержательная геймификация – заключается в использовании игровых элементов и мышления для того, чтобы изменить содержание изучаемого материала и превратить его в игру. Такой подход становится особенно полезным в области гуманитарных и социально-экономических дисциплин, когда необходимо запомнить большое количество теоретического материала [15].

В своей работе С. С. Белоконова [1] определяет два варианта геймификации при организации образовательного процесса.

Первый вариант – это сквозная геймификация, она проходит через всю тему или курс и помогает обучающимся на протяжении всего обучения. Данную геймификацию можно воплотить с помощью системы званий, например, в начале изучения темы все новички, потом после прохождения первый блока материала можно стать учеником, после второго блока продвинутым учеником и т.д.

Второй вариант – это точечная геймификация, она не проходит через всю тему или курс, а используется для отдельной части. Например, вместо теста для проверки знаний по определённой теме, можно создать квест или викторину с помощью какого-нибудь веб-сервиса. Так учебная задача превратится в соревнование, где рейтинг будет зависеть от правильности и скорости ответа [1].

В научных исследованиях по геймификации существует мнение о том, что правильное и стратегическое использование игровых элементов способно создать ситуацию обучения, которая повышает активность и мотивацию учащихся, приводя к положительным результатам в различных сферах. Несмотря на преимущества геймификации, необходимо учитывать результаты исследований, которые указывают на ограничения данного подхода: пренебрежение значением учебной темы, превращение учебной работы в игру, подходящие игры только для работы с концепциями и понятиями, некоторые игры могут не быть достаточно эффективными для обучения и невозможность преодолеть сложности материала с помощью игр [40].

Несмотря на то, что геймификация в обучении еще недостаточно исследована, результаты некоторых исследований указывают на высокий потенциал данного подхода, особенно в отношении мотивации, вовлеченности в процесс обучения и улучшения результатов обучения.

Сформулируем основные особенности геймификации в образовательном процессе, рассмотренные в авторском подходе Ю. П. Олейника [24], которые должны быть учтены при организации игрового процесса:

– игра должна иметь заданные цели, которые формируются исходя из желаемого результата;

– взаимодействие участников в рамках игры осуществляется на добровольной основе, так что они всегда могут прервать её в любой момент без негативных последствий;

– действия, совершенные в рамках игры, могут оказать влияние на ход событий и поведение других игроков в дальнейшем;

– когда игра используется как инструмент обучения, её суть раскрывается в игровой дидактике, которая позволяет эффективно и интерактивно передавать знания;

– игра в обучении должна удовлетворять принципу естественности и вызывать положительные эмоции и интерес у учеников;

– социальное взаимодействие между пользователями-игроками;

– эмоциональная вовлеченность игроков [24].

Учитывая выделенные особенности игровых механик, возможно, перевести обучение из плоскости многократного повторения в плоскость интересного занятия, достигая успеха усвоения новых знаний.

Формирование собственного игрового пространства на базе определённой платформы является эффективным инструментом в обучении и развитии личности. Для разработки игр необходимо понимать структурные компоненты, обеспечивающие их функционирование. Профессор Кевин Вербах [7] определяет три категории игровых элементов: динамики, механики и компоненты, представленные на рисунке 3. Каждая механика связана с одной или несколькими динамиками; каждый компонент связан с одним или несколькими элементами высокого уровня. Применение игровых элементов в геймификации позволяет создавать образовательные и развлекательные игры, стимулирующие развитие учащихся. Кроме того, разработка игр на основе собственного игрового пространства способствует развитию творческих и критических навыков, аналитического мышления и логической связности.

Таким образом, использование игровых элементов для создания шаблонов игр открывает широкие возможности в педагогическом процессе.

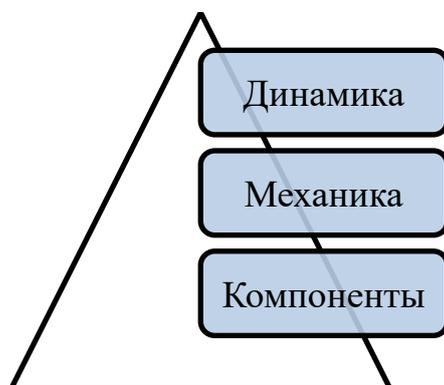


Рисунок 3 – Пирамида игровых элементов Кевина Вербаха

Рассмотрим более подробно игровые элементы пирамиды.

Одним из важнейших элементов игры является верхний уровень – слой «Динамика», в котором сосредоточены концептуальные элементы игры, представляющие ее скрытую структуру и «грамматику» игры.

К данному слою можно отнести:

- различные ограничения, которые присущи каждой игре, поскольку они необходимы для того, чтобы ограничить свободу игроков и создать определенную интригу в ходе игры;
- ограниченный спектр эмоций, доступный игрокам;
- структура хронологии, которая объединяет различные части игры в логическую цепочку;
- прогрессия в игре описывает ощущения игрока о том, что он имеет возможность улучшить свои результаты в сравнении с предыдущими уровнями;
- отношения между игроками или общие связи между людьми также входят в этот слой, поскольку они имеют важное значение для правильного восприятия игрового процесса.

В игровых процессах важную роль играет слой «Механика», состоящий из определенных действий, которые стимулируют и поддерживают игровую активность. Среди элементов данного слоя можно выделить такие важные

аспекты, как: определение целей игры, генерация случайных значений, соревнование, обратная связь, добыча ресурсов и вознаграждение. Без этих функционирующих элементов игра может не представлять для игрока большого интереса.

Однако помимо слоя «Механика» в игровом процессе играют важную роль компоненты, которые составляют слой «Компоненты». Среди данных компонентов можно выделить достижения, персонажей, уровни, социальные связи и виртуальные товары. Следует отметить, что взаимодействие между слоями «Механика» и «Компоненты» является ключевым моментом в формировании игрового опыта.

Важно понимать, что эти два слоя не являются независимыми. Игровые компоненты, находящиеся на верхних уровнях, должны быть опираются на функционирующие элементы нижних слоев. Таким образом, пирамида игровых компонентов формируется за счет взаимодействия и укрепления соединений между различными элементами игры. Удачное взаимодействие между данными слоями формирует привлекательную и интересную игровую среду, которая способна удерживать внимание игроков на протяжении продолжительного времени [7].

Геймификация – это метод мотивации пользователей, основанный на игровом подходе, который позволяет повысить эффективность использования различных функциональных возможностей, предоставляемых преподавателем. Основным принципом геймификации является обеспечение пользователей постоянной, измеримой обратной связью, которая обеспечивает динамическую корректировку пользовательского поведения и быстрое освоение функциональных возможностей игрового пространства. При применении геймификации используется поэтапное изменение и усложнение целей и задач в зависимости от приобретаемых пользователями навыков и умений. Данная стратегия позволяет повышать мотивацию пользователей за счет постоянных вызовов и увеличения сложности задач. Таким образом, пользователь ощущает рост своих способностей и достигает новых знаний в обучении [15].

Суть геймификации в организации учебного процесса состоит в том, чтобы использовать склонность человека к игре как ключ к получению знаний. Основными целями геймификации являются:

- сформировать образовательную деятельность через игру (геймифицировать учебное занятие);
- сформировать систему мотивации обучающихся на своевременное и качественное выполнение игровых заданий;
- сформировать систему роста прогресса в освоении знаний за счет использования игровых технологий.

Наряду с деловыми и ролевыми играми, геймификация учебного процесса способствует развитию коммуникабельности, целеустремленности, познавательной и интеллектуальной активности обучающихся [2].

Итак, геймификация – это использование игровых элементов в неигровой ситуации, т. е. процесс, когда элементы игры используют для достижения реальных целей. Важно, что геймификация – это не создание полноценной игры, а только использование определенных элементов. Целью этого процесса является привлечение и повышение внимания обучающихся, улучшения их мотивации при решении практических задач, обучения новым видам деятельности.

Таким образом, можно говорить о геймификации как о новом способе организации обучения, имеющем огромный педагогический потенциал. Использование технологий геймификации в образовательном процессе усиливает интерес обучающего к изучаемому предмету и повышается вероятность достижения поставленной цели, что положительно влияет на качестве образования.

## **1.2 Игровые среды и возможности их применения для изучения основ программирования**

Невозможно представить себе современный мир без цифровых технологий. Нас повсюду окружают гаджеты, компьютеры, бытовая техника с

интеллектуальным управлением, компоненты системы «Умный дом» – это лишь маленькая часть цифрового мира XXI века. Взрослые удивляются тому, как быстро дети всё это осваивают. Но мало кто задумывается о том, кем и как были созданы или разработаны такие привычные для нас вещи, в том числе и компьютерные игры [16].

Мы все любим играть, это заложено в нас с детства. Так мы развлекаемся и отдыхаем. Однако, игры могут нести в себе, кроме прочих удовольствий, еще и полезные знания. Можно обучать детей программированию через учебные игровые среды [28].

Учебные игровые среды программирования – это языки программирования и игровые среды программирования, предназначенные для введения в основы алгоритмизации, программирования и развития логико-алгоритмического (операционного) мышления у детей [8].

На сегодняшний день в сети Интернет представлено большое количество платформ, web-сервисов, игровых сред для организации процесса обучения программированию.

В условиях современного образования активное применение совместных ресурсов и средств обучения, особенно в контексте решения конкретных проблем, способствует не только формированию у учеников навыков самостоятельности, но и активному развитию поисковой, творческой и исследовательской деятельности. Одновременно с этим, педагог всегда находится в контакте с обучающимися и имеет возможность помочь им при необходимости, с доступом к заданиям обучающихся в любом месте и в любое время. Установление постоянной коммуникации позволяет понимать учащихся и направлять их мысли в нужное русло, гармонично влияя на их развитие и воспитание [8].

В таблице 1 рассмотрены основные игровые среды для изучения основ программирования.

Таблица 1 – Игровые среды для изучения основ программирования

Платформа	Описание	Пример
<b>Code.org</b> <a href="https://code.org/">https://code.org/</a>	<p>В проекте Code.org осуществляется обучение программированию, основное содержание которого охватывает базовые понятия языков программирования. Обучение реализуется в форме игрового квеста, включающего в себя множество заданий и головоломок различного уровня сложности, которые решаются с помощью программирования. Интерфейс программы является простым и понятным</p>	
<b>Blockly</b> <a href="http://blockly.ru/">http://blockly.ru/</a>	<p>Среда визуального программирования Blockly основанная в 2012 году предлагает освоить основы программирования через проект. На сайте представлены задания различных уровней (от простого к сложному) по обучению основам программирования. Данные задания могут быть использованы как в образовательных учреждениях, так и для самостоятельного обучения</p>	
<b>ColoBot</b> <a href="https://colobot.info/">https://colobot.info/</a>	<p>Трёхмерная обучающая стратегическая игра. Разработана компанией Epsitec SA. Игра призвана обучить детей навыкам программирования в игровой форме и позволяет разрабатывать дополнение к уже существующей компьютерной игре с использованием языка C++ и Java. Образовательный ресурс ColoBot русифицирован и имеет подробное описание используемого языка, и требования к разрабатываемым миссиям</p>	
<b>JavaRush</b> <a href="https://javarush.com/questions">https://javarush.com/questions</a>	<p>В настоящее время изучение языка программирования Java является одним из наиболее востребованных в IT-сфере. Для облегчения этого процесса, JavaRush предлагает новаторский подход – онлайн-игру, которая поможет пользователям изучать язык. Курс состоит из нескольких уровней, каждый из которых представляет собой отдельный мир – солнечную систему на звездной карте. Уровень проходится в течение нескольких дней, и во время его прохождения, ученики посещают планеты, являющиеся лекциями. Каждая лекция представляет собой интерактивное пространство, в котором дети будут изучать концепции Java. Таким образом, прохождение уровня ученикам воспроизводит опыт космического путешествия, где знания являются ключом для прохождения испытаний на каждой планете. Как только все лекции пройдены, пользователь переходит к следующей звездной системе и начинает новый уровень</p>	
<b>CodeCombat</b> <a href="https://codecombat.com/">https://codecombat.com/</a>	<p>В CodeCombat представлена очень симпатичная графическая оболочка для решения задач в виде набора небольших квестов. Она позволяет пользователям выбирать один из четырех языков программирования: Python, JavaScript, CoffeeScript или Lua. Основная цель игры заключается в том, чтобы позволить пользователям выступить в роли волшебника и с помощью программного кода менять окружающий игровой мир</p>	

## Окончание таблицы 1

Платформа	Описание	Пример
	Для каждого уровня в игре задается определенная задача, например, собрать самоцветы, которые помогут призвать союзников и найти выход из лабиринта	
<b>Learnis</b> <a href="https://www.learnis.ru/">https://www.learnis.ru/</a>	Разработан веб-сервис Learnis, предназначенный для создания квестов, викторин и игр путем использования набора электронных интерактивных приложений	
<b>Microsoft MakeCode</b> <a href="https://www.microsoft.com/en-us/makecode">https://www.microsoft.com/en-us/makecode</a>	Microsoft MakeCode предоставляет возможность детям и молодежи обучаться программированию, используя блочные текстовые редакторы для создания увлекательных проектов. Этот сервис бесплатен и может быть использован на всех платформах и браузерах. Основная цель сервиса – представить программирование доступным и привлекательным способом. Для этого, MakeCode использует модель программирования блоков, чтобы позволить пользователю изучать концепции кодирования более осязаемым образом. Как только пользователь освоится с элементами и структурой кодирования, он сможет перейти к созданию более сложных программ. Блоки сопоставляются непосредственно с реальными строками кода на языке программирования JavaScript и Python	
<b>Minecraft Education Edition (далее Minecraft EDU)</b> <a href="https://education.minecraft.net/ru-ru">https://education.minecraft.net/ru-ru</a>	Одной из веток сайта Code.Org, разработанной в сотрудничестве с Microsoft, стал проект Minecraft Education, выполненный по мотивам популярной среди подрастающего поколения игру Minecraft. Minecraft – это игра о размещении и разбивании блоков, где единственным ограничением является воображение игрока. Minecraft творческая платформа для обучения и изучения компьютерных наук, которая позволяет создавать программы или собственные мини-игры. Школьники решают задачи по добыче необходимых полезных ископаемых, автоматизации постройки объектов и изменения игрового поведения Minecraft. Для этих целей им предоставлен встроенный редактор Code Builder	
<b>CheckiO</b> <a href="https://checkio.org/">https://checkio.org/</a>	Check IO – это игровой ресурс, который позволяет наглядно изучать и применять TypeScript и Python. Начинающие программисты имеют возможность решать цепочки задач и получать оценки других игроков, а более опытные пользователи могут поделиться своими знаниями с другими пользователями	

Подводя итог вышеописанным игровым средам, которые позволяют выполнять и создавать различные по сложности задания для использования в их учебном процессе, следует отметить, что самым понятным интерфейсом

обладает платформа Code.org. Code Combat, хотя и обладает меньшим выбором форматов заданий, однако, предлагает уникальные способ по организации соревновательной работы в командах. Blockly предполагает наличие совместной деятельности и коммуникации, общий предмет и способ деятельности, иерархию внутриигровых мотивов, имитацию реальных процессов. Платформа Minecraft EDU вдохновляет на обучение и открывает новые возможности для решения любых задач и проблем.

Таким образом, при использовании игровых сред при обучении основам программирования, возможно повышение мотивации учащихся и вовлеченности их в учебный процесс. Это достигается за счет соответствующего методического сопровождения, которое позволяет использовать игровые элементы вне игрового контекста. Применение таких приемов не требует изменения самого процесса обучения, а лишь дополнения его инструментарием, состоящим из игровых элементов и приемов. Элементы игры не являются единственным методом обучения и должны использоваться органично вместе с традиционными методами обучения.

### **1.3 Дидактический потенциал технологии геймификации в формировании игровой среды при обучении основам программирования**

В процессе применения геймификации в обучении необходимо обеспечить выполнение образовательных задач. Внутренняя мотивация учеников способствует выполнению приоритетных образовательных задач в рамках игровых заданий. Геймификация отличается от других игровых практик, поскольку не имитирует деятельность, а качественно модифицирует способ организации учебной деятельности, сохранения содержания образовательного процесса. Для того чтобы избежать бессмысленного зарабатывания игровых баллов, необходимо установить соответствующие игровые и образовательные задачи на протяжении всего обучения [11].

Федеральный государственный образовательный стандарт основного общего образования (далее ФГОС ООО) фокусируются на практических навыках детей: они должны понимать, как связаны предметы и как знания помогают в реальной жизни [30].

В современном образовании наблюдается растущий интерес к использованию геймификации в учебном процессе. Геймификация является неотъемлемой частью цифровых технологий, обладая рядом свойств, таких как мультимедийность, интерактивность, нелинейность подачи информации и информативность. Геймификация позволяет оптимизировать учебный процесс, основываясь на применении следующих дидактических принципов, представленных в таблице 2 [19].

Таблица 2 – Реализация дидактических принципов посредством использования геймификации

<b>Дидактический принцип</b>	<b>Средства реализации дидактического принципа</b>
Принцип наглядности (возможность зрительно и динамически воспринимать информацию)	Мультимедийно-динамическая форма подачи материала (использование графических изображений, аудио, видеофайлов и других интерактивных элементов)
Принцип сознательности и активности обучаемых (ориентирован на результаты и на повышение личной ответственности учеников за свою работу)	Мгновенная обратная связь (позволяет получать информацию о том, насколько хорошо выполняются задачи) и возможность оценивания и самооценивания (анализ учениками собственных достижений и возможностей роста)
Принцип последовательности и систематичности (позволяет выстраивать логически связанные этапы практики)	Систематическое усложнение условий игры и игрового материала
Принцип индивидуализации обучения (предоставляет возможность ученику самому определить темп обучения, количество времени и способы получения знаний)	Возможность выбора каждым учащимся индивидуальной траектории обучения (учет потребностей и интересов каждого ученика и создании оптимальных условий для его обучения)
Принцип доступности и посильности (учитывая индивидуальные особенности каждого ребенка необходимо создавать условия для успешного обучения и развития)	Возможность самостоятельно выбирать сложность учебного материала (ясность и доступность изложения учебного материала, учитывается уровень подготовки каждого ребенка)
Принцип прочности (присутствие положительных моментов способствует активному и продуктивному усвоению материала)	Задействование эмоций обучающихся (на уровне психологии и эмоционального состояния обучающегося формируются механизмы, повышающие усвоение материала)

Для достижения эффективности обучения важно реализовывать основные дидактические принципы. Один из них – мгновенная обратная связь, позволяющая детям получать обратную связь на свой ответ в режиме реального времени. Кроме того, важным принципом является сознательность и активность обучаемых, достигаемые через возможность взаимооценивания и самооценивания. При проведении занятий в интерактивной форме создается более активная и взаимодействующая группа. Наглядность материала также является важным аспектом обучения, особенно если использовать мультимедийно-динамические формы подачи. Эффект визуализации может быть максимально усилен за счет геймификации и новых технологий, таких как 3D-технологии, дополненная реальность и технологии, обеспечивающие эффект присутствия. Они позволяют создавать более вовлекающие и захватывающие возможности для обучения и повышения эффективности.

Геймификация является эффективным инструментом обучения с использованием игровых элементов и принципов. Она позволяет организовать обучение по принципу последовательности и систематичности, при котором учащиеся постепенно овладевают учебным материалом от простого к сложному. Внедрение геймификации позволяет выстраивать процесс обучения для каждого ребенка с учетом скорости усвоения каждым ребенком [11].

Принцип доступности и посильности обучения предусматривает систематическое усложнение учебного материала, с выбором заданий, т.е. каждое задание должно быть адаптировано под уровень знаний ребенка в области программирования. Игровые элементы, способны достичь принципа прочности усвоения материала, благодаря улучшению эмоционального состояния ребенка.

Организация учебного процесса с применением геймификации при обучении основам программирования позволяет развивать не только навыки программирования и алгоритмического мышления, но и коммуникативные умения, критическое мышление, готовность к групповой работе и толерантность. Технологии геймификация позволяет использовать проблемные

методы обучения, поисковую деятельность и исследовательские проекты, стимулирует самооценивание и взаимооценивание, а также симулирует процесс обучения путем создания виртуального игрового пространства. Преимущество геймификации заключается в том, что она активизирует творческий потенциал личности участников образовательного процесса и позволяет не только эффективно обучать основам программирования, но и развивать личностные качества учеников [8; 11].

На современном этапе развития образования наиболее актуальной задачей является развитие системы самообразования. Однако для достижения этой цели необходимо стимулировать учебно-познавательную активность детей, которая становится возможной в условиях организации их самостоятельной деятельности. В этом контексте, геймификация может быть эффективным инструментом для развития системы самообразования. Создание различных платформ и сред, способствующих самостоятельной активности детей в направлении саморазвития и раскрытия творческих способностей, может обеспечить уровень внутренней мотивации и общей удовлетворенности процессом обучения. Важно отметить, что участие в игровых процессах не должно заменять традиционные методы обучения, а быть дополнением к ним.

С каждым годом увеличивается количество сервисов и платформ, использующих различные формы реализации технологии геймификации в образовании и обладающих дидактическими элементами обучения программированию.

В таблицах 3, 4, 5 и 6 рассмотрены несколько проектов уровня, реализованные на принципах структурной геймификации для сопоставления их дидактического потенциала.

Таблица 3 – Дидактические свойства и функции Code.org

Дидактические свойства	Дидактические функции
Разнообразные типы уровней для изучения заданий различных форматов	Развитие речевой деятельности и языковых навыков у учеников, развитие социокультурной компетенции и подготовка к жизни в обществе, повышение общей культуры учеников и усилению их позитивных качеств, создание заданий экзаменационного формата и организация самостоятельной и аудиторной работы

### Окончание таблицы 3

<b>Дидактические свойства</b>	<b>Дидактические функции</b>
Занятие «Предсказание» – геймификация вопросов с множественным выбором	Использование при промежуточном контроле, что позволяет оценить степень освоения материала на определённом этапе обучения. Самооценивание (оценка собственных знаний и навыков) и взаимооценивание (оценивание работы своих товарищей и корректировка для улучшения качества их работы)
Занятие «Выбор уровня» – геймификация мероприятий с множественным выбором проектов	Повышение мотивации и уровня вовлеченности детей через выбор уровней освоения материала, организация индивидуальной и парной работы, организация самостоятельной (активное участие в обучении, развитие навыков решения задания) и аудиторной работы (поддержка и контроль знаний учеников)
Занятие «Практика» – геймификация заданий на сопоставление	Формирование различных языковых навыков: лексических, грамматических, фонетических и орфографических
Инструмент «Свободная игра»	Развитие навыков письменной коммуникации через организацию индивидуальной и групповой работы, обеспечение эффективной коммуникации между участниками процесса обучения
Карта	Администрирование и мониторинг учебного процесса через отслеживание прогресса каждого ученика, организация самостоятельной и аудиторной работы детей
Коллекция созданных другими преподавателями проектов	Использование заданий для самообразования, восполнения пробелов учащихся, использованием преподавателями коллекций для создания новых проектов и уроков, опираясь на наработки других специалистов

Таблица 4 – Дидактические свойства и функции среды программирования Blockly

<b>Дидактические свойства</b>	<b>Дидактические функции</b>
Разнообразные типы уровней для изучения заданий и тестов различных форматов	Развитие речевой деятельности и языковых навыков у учеников, развитие социокультурной компетенции и подготовка к жизни в обществе, повышение общей культуры учеников и усилению их позитивных качеств, создание заданий экзаменационного формата и организация самостоятельной и аудиторной работы
Раздел «Задачник» – геймификация заданий с выбором уровней задач (включает задания из ОГЭ и ЕГЭ)	Использование при промежуточном контроле, что позволяет оценить степень освоения материала на определённом этапе обучения. Самооценивание (оценка собственных знаний и навыков) и взаимооценивание (оценивание работы своих товарищей и корректировка для улучшения качества их работы)
Раздел «Среда обучения» – геймификация знакомства школьников с процессом и ходом исполнения	Формирование различных языковых навыков: лексических, грамматических, фонетических и орфографических. Наличие блока пауза даёт возможность замедлить процесс выполнения программы в

#### Окончание таблицы 4

<b>Дидактические свойства</b>	<b>Дидактические функции</b>
созданных программ	определённом месте на нужное время. Это позволяет преподавателю во время паузы комментировать процесс выполнения программы, а также улучшает наглядность восприятия происходящих процессов. Имеются демоверсии ОГЭ и ЕГЭ
Раздел «СТЕР-обучение» – геймификация знакомства школьников с процессом пошагового исполнения созданных программ (трассировка)	Развитие навыков письменной коммуникации через организацию индивидуальной и групповой работы, обеспечение эффективной коммуникации между участников процесса обучения
Раздел «Черепашка» – геймификация знакомства школьников с графическими возможностями языка программирования	Повышение мотивации и уровня вовлеченности детей через выбор уровней освоения материала, организация индивидуальной и парной работы, организация самостоятельной (активное участие в обучении, развитие навыков решения задания) и аудиторной работы (поддержка и контроль знаний учеников)
Коллекция созданных другими преподавателями проектов	Использование заданий для самообразования, восполнения пробелов учащихся, использованием преподавателями коллекций для создания новых проектов и уроков, опираясь на наработки других специалистов

Таблица 5 – Дидактические свойства и функции CodeCombat

<b>Дидактические свойства</b>	<b>Дидактические функции</b>
Разнообразные типы уровней для изучения заданий и тестов различных форматов	Развитие речевой деятельности и языковых навыков у учеников, развитие социокультурной компетенции и подготовка к жизни в обществе, повышение общей культуры учеников и усилению их позитивных качеств, создание заданий экзаменационного формата и организация самостоятельной и аудиторной работы
Игра «Ozaria» – приключенческая игра с сюжетной линией	Повышение мотивации и уровня вовлеченности детей через выбор уровней освоения материала, организация индивидуальной и парной работы, организация самостоятельной (активное участие в обучении, развитие навыков решения задания) и аудиторной работы (поддержка и контроль знаний учеников)
Класс «CodeCombat» – оригинальная игра учит навыкам реального программирования через силу игры и встроенный элементы геймификации	Использование при формирующем контроле, для повышения мотивации учащихся на изучение учебной информации. Самооценивание (оценка собственных знаний и навыков), организация индивидуальной, парной и групповой работы, организация самостоятельной и аудиторной работы, формирование языковых навыков
Миры «Roblox» – творческое программирование игровых персонажей	Визуализация учебного материала, организация самостоятельной и аудиторной работы направленной на закрепление полученных знаний и навыков, повышение мотивации, формирование языковых навыков, развитие умений чтения и аудирования

## Окончание таблицы 5

Дидактические свойства	Дидактические функции
«Лига искусственного интеллекта» – киберспортивная лига соревновательного программирования, для изучения реального кода	Повышение мотивации и уровня вовлеченности, соревновательности
Онлайн-классы «Вживую»	Осуществление проектной деятельности через правильную организацию групповой и индивидуальной работы с учетом факторов влияющих на результативность работы
Коллекция созданных другими преподавателями проектов	Использование заданий для самообразования, восполнения пробелов учащихся, использованием преподавателями коллекций для создания новых проектов и уроков, опираясь на наработки других специалистов

Ещё одна платформа, которая предоставляет возможность преподавателям создавать задания с геймификацией – Minecraft EDU.

Таблица 6 – Дидактические свойства и функции Minecraft EDU

Дидактические свойства	Дидактические функции
Разнообразные типы уровней для изучения заданий и тестов различных форматов	Развитие речевой деятельности и языковых навыков у учеников, развитие социокультурной компетенции и подготовка к жизни в обществе, повышение общей культуры учеников и усилению их позитивных качеств, создание заданий экзаменационного формата и организация самостоятельной и аудиторной работы
Увлекательные задания и головоломки	Использование при промежуточном контроле, что позволяет оценить степень освоения материала на определённом этапе обучения. Самооценивание (оценка собственных знаний и навыков) и взаимооценивание (оценивание работы своих товарищей и корректировка для улучшения качества их работы)
Разнообразные типы уровней для изучения заданий и тестов различных форматов	Развитие речевой деятельности и языковых навыков у учеников, развитие социокультурной компетенции и подготовка к жизни в обществе, повышение общей культуры учеников и усилению их позитивных качеств, создание заданий экзаменационного формата и организация самостоятельной и аудиторной работы
Игровые персонажи	Повышение мотивации и уровня вовлеченности детей через выбор уровней освоения материала, организация индивидуальной и парной работы
Классные доски, таблички	Визуализация учебного материала, организация самостоятельной и аудиторной работы направленной на закрепление полученных знаний и навыков, повышение мотивации, формирование языковых навыков, развитие умений чтения и аудирования

## Окончание таблицы 6

<b>Дидактические свойства</b>	<b>Дидактические функции</b>
Чат	Обеспечение коммуникации участников процесса обучения через возможность ввода сообщений в чате, обеспечение четких правил коммуникации и контроля за их соблюдение
Режим класса (Classroom Mode)	Администрирование и мониторинг учебного процесса через отслеживание телепортация игроков, переключение различных игровых параметров, которые влияют на игровой мир
Шаблоны миров	Создание своего динамического мира на основе базовых биномов
Встроенный редактор Microsoft MakeCode (Code Builder)	Формирование синтаксических навыков языка навыков программирования (лексических, грамматических, фонетических, орфографических). Позволяет автоматизировать постройки объектов, изменяет игровое поведения Minecraft
Коллекция созданных другими преподавателями иммерсивных миров и заданий	Использование заданий для самообразования, восполнения пробелов учащихся, использованием преподавателями коллекций для создания новых проектов и уроков, опираясь на наработки других специалистов

Вышеизложенное позволяет сделать вывод о том, что все сервисы обладают дидактическим потенциалом для использования в образовательном процессе, но среда Minecraft EDU обладает большими возможностями по сравнению с другими. В основу, которой положена одна из самых популярных компьютерных игр в мире – игра Minecraft, разработанная в 2011 году шведским программистом Маркусом Перссоном. В этой модификации компьютерной игры Minecraft учащиеся могут начать изучать основы программирования, управляя блоками – элементы, из которых состоит игровой мир, и «мобами» (в игре присутствует специальный исполнитель «Агент») с помощью сервиса MakeCode. Вместо обычного вывода работы программы в виде текстовой строки на экране, учащиеся видят результат «вживую» в игре. Так, например, можно построить лабиринт и запрограммировать Агента – искать из него выход или с помощью кода построить любое сооружение. С помощью элементов игры учащиеся могут познакомиться с переменными, условными командами и ветвлениями, различными видами алгоритмов, с циклами, функциями и понятиями, составляющими основы программирования.

Среди ключевых преимуществ Minecraft EDU по сравнению с остальными платформами можно выделить следующие:

- погружение учебного процесса в одну из самых популярных игр в мире;
- работа в трехмерном мире;
- возможность программировать исполнителя различными способами (блочное программирование, python и javascript);
- возможность создания игрового мира с заданиями для детей, которые автоматически проверяются платформой;
- возможность применять данную платформу в начальной, основной и средней школе (переход от блочного программирования к программированию на языке высокого уровня);
- возможность создавать задания, которые необходимо выполнять в команде;
- ученики могут объединять свой код в чат-команды, которые могут вызываться непосредственно из игры;
- предоставление обучающимся инструментов для автоматизации окружающего мира игры, упрощающих создание игровых элементов;
- командное окно работает в режиме реального времени и показывает, какие именно игровые команды исполняются непосредственно из вашего кода;
- возможность поделиться созданными модами с друзьями;
- возможность использования онлайн-обучения, т.е. присутствует на платформе списка ресурсов (Rutube – канал, видеоуроки и различные разработки), к которым дети могут получить доступ, чтобы продолжить свое компьютерное образование за пределами класса.

Вышеперечисленные особенности одновременно не встречаются ни в одной из рассмотренных игровых сред, что делает Minecraft EDU явным фаворитом среди них.

Вопросам использования Minecraft EDU на уроках информатики посвящено небольшое количество исследований [2; 6; 10; 21; 22; 33]. При этом

все эти работы описывают преимущества платформы и ее использование на уроках или в рамках внеурочной деятельности.

Можно сделать следующие выводы: технологии геймификации реализуют основные дидактические принципы, за счет возможности выбора индивидуальной траектории, моментальной корректирующей обратной связи и мультимедийно-динамической формы представления, что является ключевыми факторами, обеспечивающие эффективное обучение. Интерактивный характер геймификации способствует активному участию учащихся в процессе обучения и улучшению качества усвоения информации.

Таким образом, использование технологии геймификации в игровой среде Minecraft EDU при изучении основ программирования, имеет дидактический потенциал и может быть использовано в рамках образовательного процесса для эффективного обучения.

## **2 Реализация технологии геймификации в формировании игровой среды для обучения основам программирования в 9 классе**

### **2.1 Анализ содержания линии «Алгоритмизация и программирование» в школьном курсе информатики**

На современном этапе развития общества программирование имеет огромное значение и влияет на многие аспекты нашей жизни. Умение программировать является востребованным и необходимым навыком в современном мире, его изучение в школе способствует развитию алгоритмического, логического мышления и формированию цифровых навыков. В своих исследованиях А. П. Ершов [10] обращал особое внимание на программирование, которое он считал второй грамотностью. По его мнению, мы живем в мире программ, и часто сами не осознаем, как программируем нашу жизнь каждый день. Однако он не считал необходимым навязывать детям новые, не присущие им знания и навыки, а скорее, предлагал научить их способности планировать свои действия и предсказывать их последствия. Такой подход поможет раскрыть скрытые стороны мышления и поведения, которые обычно формируются неосознанно. «Программируя» свою жизнь, дети научатся более эффективно использовать свои возможности и быть успешными в будущем [10].

В течение многих лет развития отечественного курса информатики в школах, линия «Алгоритмизация и программирование» сохраняется как одна из ключевых разделов данного курса. В научно-исследовательских работах отечественных педагогов, таких как Л. Л. Босова [5], М. П. Лапчик [17], К. Ю. Поляков [29], И. Г. Семакин [35], Н. Д. Угринович [38] и других, были рассмотрены различные методы обучения данной теме. Эти ученые изучили методологические аспекты, связанные с формированием умения логически мыслить, учением понимать состояние задачи, представлять ее в виде алгоритма и решать данную задачу путем программирования.

Содержательная линия «Алгоритмизация и программирование» постоянно совершенствуется, находясь под влиянием мировых тенденций. Л. Л. Босова [6], рассматривая отечественные и зарубежные подходы обучения программированию, выделила ряд тенденций в этой области, к которым относятся: «раннее начало, обязательность и непрерывность обучения программированию; разнообразие современных сред для программирования» [6].

С учетом действующих в настоящее время ФГОС ООО, основная школа обязана предоставлять обучение информатике – не менее чем одного часа в неделю в VII–IX классах. В данном контексте, уроки по алгоритмизации и программированию имеют решающее значение для формирования компетенций современного гражданина, владеющего информационными технологиями. Всего на изучение предмета «Информатика и ИКТ» отводится 70 часов на базовом уровне и 280 часов на углубленном уровне [30].

В примерной основной образовательной программе основного общего образования учебного предмета «Информатика» при изучении раздела «Алгоритмы и элементы программирования» рассматриваются следующие темы, представленные на рисунке 4.

### **Алгоритмы и элементы программирования**

- 
- Исполнители и алгоритмы. Управление исполнителями
  - Алгоритмические конструкции
  - Разработка алгоритмов и программ
  - Анализ алгоритмов
  - Робототехника
  - Математическое моделирование

Рисунок 4 – Темы раздела «Алгоритмы и элементы программирования»

Данная программа определяет примерный учебный план, в котором выделяют базовый и углублённый уровень изучения содержательной линии «Алгоритмы и элементы программирования».

На базовом уровне изучают «алгоритмические конструкции, составление алгоритмов и их программная реализация, анализ алгоритмов, математическое моделирование». А углублённый уровень содержит темы «алгоритмы и структуры данных, языки программирования, разработка программ, элементы теории алгоритмов, математическое моделирование» [31].

Основной предметной целью раздела «Алгоритмы и элементы программирования» является формирование у учащихся умения использовать языки программирования для моделирования ситуаций и решения задач.

Рассмотрим подходы к изучению основ программирования в частных учебно-методических комплексах (далее УМК) по предмету «Информатика и ИКТ»: Л. Л. Босовой [5], К. Ю. Полякова [29], И. Г. Семакина [35] и Н. Д. Угриновича [38].

В учебнике Л. Л. Босовой [5] раздел алгоритмизация и программирование, включает в себя темы, представленные на рисунке 5.

### **2.1. Решение задач на компьютере**

- этапы решения задач на компьютере, задачи о пути торможения автомобиля

### **2.2. Одномерные массивы целых чисел**

- Описание, заполнение, вывод массива, вычисление суммы элементов массива, последовательный поиск в массиве, сортировка массив

### **2.3. Конструирование алгоритмов**

- последовательное построение алгоритма, разработка алгоритма методом последовательного уточнения для исполнителя Робот, вспомогательные алгоритмы

### **2.4. Запись вспомогательных алгоритмов на языке Паскаль**

- процедуры и функции

### **2.5. Алгоритмы управления**

- управление, обратная связь, системы с программным управлением, робототехника

Рисунок 5 – Содержание раздела «Алгоритмизация программирование»

в учебнике Л. Л. Босовой

В разделе связанном с программированием не рассматриваются различные языки программирования и их классификация, в данном разделе изучают школьный алгоритмический язык КуМир и язык программирования Pascal, в которых рассматриваются соответствующие части из примерной образовательной программы.

В учебнике Н. Д. Угриновича [38] раздел, относящийся к алгоритмизации и программированию, имеет содержание, отражённое в рисунке 6.

### 1. Основы алгоритмизации и объектно-ориентированного программирования

- алгоритм и его формальное исполнение; кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования; переменные: тип, имя, значение; арифметические, строковые и логические выражения; функции в языках объектно-ориентированного и процедурного программирования; графические возможности объектно-ориентированного языка программирования Visual Basic

Рисунок 6 – Содержание раздела «Алгоритмизация программирование»  
в учебнике Н. Д. Угриновича

Н. Д. Угринович [38] в учебнике «Информатика» для 9 класса совмещает алгоритмизацию и программирование в одну главу, но ее содержание сильно отличается от выше описанной примерной программы. Вначале кратко говорится об алгоритмах. По большей части в этой главе рассматривается объектно-ориентированное программирование на языке Visual Basic.

В учебнике К. Ю. Полякова [29] есть только одна глава, относящаяся к разделу алгоритмизации и программированию, представленная на рисунке 7.

### 4. Программирование

- символьные строки; обработка массивов, матрицы (двумерные массивы); сложность алгоритмов; как разрабатываются программы; процедуры и функции

Рисунок 7 – Содержание раздела «Алгоритмизация программирование»  
в учебнике К. Ю. Полякова

К. Ю. Поляков [29], рассматривая изучение вопросов линии алгоритмизации и программирования в курсе основной школы, знакомит с основами языка программирования Python и школьным алгоритмическим языком. При этом каждый параграф учебника сопровождается вопросами и интерактивными заданиями, которые нацелены на закрепление изложенного материала на понятийном уровне, а не на уровне механического запоминания.

В учебнике И. Г. Семакина [35] разделы, относящиеся к алгоритмизации и программированию, имеют содержание, представленное на рисунке 8.

## 1. Управление и алгоритмы

- управление и кибернетика; управление с обратной связью; определение и свойства алгоритма; графический учебный исполнитель; вспомогательные алгоритмы и подпрограммы; циклические алгоритмы; ветвление и последовательная детализация алгоритма

## 2. Введение в программирование

- что такое программирование; алгоритмы работы с величинами; линейные вычислительные алгоритмы; знакомство с языком Паскаль; алгоритмы с ветвящейся структурой; программирование ветвлений на Паскале; программирование диалога с компьютером; программирование циклов; алгоритм Евклида; таблицы и массивы; массивы в Паскале; одна задача обработки массива; поиск наибольшего и наименьшего элементов массива; сортировка массива

Рисунок 8 – Содержание раздела «Алгоритмизация программирование»  
в учебнике И. Г. Семакина

Большая часть учебника И. Г. Семакина [35] посвящена разделу алгоритмизации и программированию. Раздел алгоритмизации отличается от программы тем, что алгоритм в учебнике рассматривается с точки зрения кибернетики, а раздел программирования, так же как и у Л. Л. Босовой [5] полностью посвящен языку программирования Паскаль и работе в нем.

В УМК разных авторов на изучение раздела «Алгоритмы и элементы программирования» отводится от 15 до 27% учебного времени.

Из рассмотренных частных методик обучения информатики в базовом курсе, авторами в основном, предлагается проводить обучение основам

программирования либо на школьном алгоритмическом языке, либо на языке программирования Pascal, либо Visual Basic.

Согласно требованиям ФГОС ООО приведен перечень допустимых языков программирования: базовый уровень – «Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык», исполнители – «чертежник, черепаха»; углубленный уровень – «Python, C++ (JAVA, C#)» [30].

Для тестирования знаний учащихся в области программирования в заданиях основного государственного экзамена (далее ОГЭ) и единого государственного экзамена (далее ЕГЭ), предусмотрено использование пяти вариантов языков программирования: школьный алгоритмический язык (КуМир), Basic, Pascal, Python и C++. Данные языки подобраны для максимального охвата различных программных платформ и алгоритмов, что позволяет оценить уровень подготовки широкого круга учащихся.

В таблице 7 рассмотрим языки программирования, которые изучаются в вышеперечисленных УМК со стороны их синтаксиса и удобства для школьного обучения.

Таблица 7 – Анализ языков программирования

Язык программирования	Пример синтаксиса	Примечание	Среды
Школьный Алгоритмический Язык	<b>алг</b> Приветствие <b>нач</b> вывод «Привет, мир!» <b>кон</b>	Подходит для исполнителей: робот и черепаха. Поддерживает два типа числовых данных: целые и вещественные. Не подойдет для написания сложных программ	КуМир
Pascal	<b>program</b> Hello; <b>begin</b> writeln('Привет, мир!'); <b>end</b>	Современный язык программированию, который интегрирован в учебные планы школьной программы. Некоторые ученики сталкиваются с проблемой неудобства при использовании данного языка из-за его строгой типизации и структуры. Это может быть значительным помехой в понимании принципов и структуры работы программы	PascalABC.NET

Окончание таблицы 7

Язык программирования	Пример синтаксиса	Примечание	Среды
Python	<pre>print('Привет, мир!')</pre>	Современный язык программирования Python крайне популярен среди учащихся. Он не имеет строгих синтаксических правил, что упрощает его изучение и повышает интерес к этой теме. В Python реализован ряд новых функций и возможностей, что делает его более привлекательным для программистов. Одним из главных особенностей является использование объектно-ориентированного подхода, который позволяет упростить разработку сложных программных продуктов	Visual Studio, PyCharm, Python ide, Minecraft EDU
C++	<pre>int main(){     cout&lt;&lt; 'Привет, мир!';     return 0; }</pre>	Следует отметить, что данные языки программирования, рассчитанные на более опытных программистов, отличаются повышенной сложностью, обусловленной их специализированностью. Благодаря этому, их использование может ослабить интерес обучающихся к программированию	Visual Studio, Eclipse
C#	<pre>namespace HelloWorld {     class Hello     {         static void Main()         {             Console.WriteLine('Привет, мир!');         }     } }</pre>		Visual Studio, Eclipse
Java	<pre>class HelloWorld{     public static void     main(String[] args){         System.out.println('Привет, мир!');     } }</pre>	Язык программирования Java, более читаемым по сравнению с языками C++ и C#. Для успешной работы на языке Java необходимо подключать дополнительные библиотеки	Idea, Eclipse

Большинство авторов учебников обращаются к школьному алгоритмическому языку программирования, так как у него достаточно простой синтаксис, код записывается на русском языке и прекрасно подходит для

знакомства с основами алгоритмического мышления в текстовом формате. Pascal , в свою очередь, известен своей надежностью и по структуре похож на школьный алгоритмический язык, но при этом код записывается на английском со строгой типизацией. Python более приближен к «настоящему» программированию, применяется для создания инструментов машинного обучения, при этом его синтаксис прост. C++, C#, Java более сложные для обучения школьников в основной школе, больше подходит для изучения в 10–11 классах на углубленном уровне для написания программ высокой производительности [25].

В результате сравнительного анализа нескольких языков программирования, можно сделать вывод о том, что наиболее современным и практичным в настоящее время является язык программирования Python, интерес к которому растет не только в нашей стране, но и за рубежом. При использовании языка программирования Python, педагоги отмечают следующие его достоинства:

- отсутствие необходимости в инструкциях, не имеющих прямого отношения к алгоритму;
- программа на языке Python обычно записывается более компактно, чем на языках C++, Pascal и Basic, но при этом сохраняет свою понятность;
- обладает свободной и кроссплатформенной реализацией;
- интерпретируемая архитектура упрощает написание, запуск и отладку программ;
- предоставляет множество высокоуровневых структур данных, таких как списки, множества, ассоциативные массивы и длинная арифметика;
- богатая библиотека языка Python позволяет легко разрабатывать графические приложения, веб-приложения и многое другое.

Все эти качества делают использование языка Python возможным не только для освоения основ программирования в курсе информатики основной школы, но и для углубленного изучения программирования информатики в старшей школе [6].

Подводя итог обзору особенностей изложения содержательной линии «Алгоритмизация и программирование» в УМК и примерной образовательной программы, можно отметить, что у каждого автора свой подход к изучению алгоритмизации и программирования и не все из них придерживаются примерной образовательной программы. В учебниках отражены результаты большой исследовательской работы в построении методической системы обучения программированию.

Современный процесс обучения программированию требует использования языков программирования, соответствующих современным требованиям и стандартам, которые применяются в рамках государственных экзаменов ЕГЭ и ОГЭ. В этом контексте, авторские методики, хоть и соответствуют требованиям ФГОС, но не учитывают современные языки программирования, которые необходимы для успешной подготовки к экзаменам. При этом следует отметить, что авторские методики, хотя и могут обеспечить учащихся знаниями в рамках образовательной программы, но в перспективе не могут быть эффективными для подготовки к государственным экзаменам из-за использования устаревших языков программирования. Кроме того, в настоящее время мотивация учеников к изучению устаревших языков программирования ниже, чем к изучению современных языков программирования. Современные языки программирования не только облегчают процесс создания программ, но и предоставляют широкий спектр специальных возможностей, что делает их более привлекательными для учащихся.

На основе примерной образовательной программы и изученных УМК Л. Л. Босовой [5], К. Ю. Полякова [29], И. Г. Семакина [35] и Н. Д. Угриновича [38] разработано тематическое планирование по изучению основ языка программирования Python, представленное в таблице 8.

Таблица 8 – Тематическое планирование раздела «Основы языка программирования Python»

№	Тема	Часы
1	<b>Введение в основы языка программирования Python</b> (основные характеристики языка Python, переменные, основные типы данных, ввод и вывод данных, синтаксис языка, структура программы)	1
2	<b>Линейная алгоритмическая конструкция</b> (арифметические операции в Python, построение программ линейной структуры)	1
3	<b>Программирование разветвляющихся алгоритмов</b> (операции сравнения, логические операции, условные конструкции в Python, вложенные и каскадные условия, построение программ структуры ветвления)	2
4	<b>Программирование циклических алгоритмов</b> (операторы структуры цикла, счетчик (for), цикл с условием (while), разработка программ циклической структуры)	2
5	<b>Строки</b> (методы работы со строками)	2
6	<b>Списки</b> (методы работы со списками)	2
7	<b>Массивы</b> (ввод и вывод массивов, перестановка элементов массива, сортировка массива, алгоритмы обработки массивов)	2
8	<b>Функции</b> (инструкции и аргументы функции)	2
9	<b>Работа над итоговым проектом</b>	2
	<b>ВСЕГО</b>	<b>16</b>

Таким образом, данное тематическое планирование совпадает с примерной образовательной программой и рассчитано на 16 учебных часов. За основной язык программирования выбран Python.

## 2.2 Разработка учебно-методического сопровождения для изучения основ языка программирования Python с использованием игровой среды

На основе тематического планирования, представленного в параграфе 2.1, и учебника «Информатика» для 9 класса К. Ю. Полякова, Е. А. Еремина [29] разработано методическое сопровождение игровой среды Minecraft EDU для изучения основ языка программирования Python в 9 классе.

Потребность в создании методического сопровождения для обучения языку программирования Python, обусловлена несколькими факторами:

– прежде всего, существующая литература не адаптирована для использования учащимися 9 классов (самоучители по изучению Python);

– кроме того, методические материалы, разработанные авторами УМК, нацелены на проведение аудиторных занятий и не подходят для индивидуального самообучения. Например, презентации К. Ю. Полякова [29] по изучению алгоритмизации и языка программирования Python не содержат необходимые инструкции и пошаговые рекомендации для обучающихся;

– изучение языков программирования требует строгой структуризации знаний, для каждого отдельного языка программирования, в нашем случае Python.

Основной целью изучения раздела «Основы языка программирования Python» является формирование у учащихся необходимых навыков, которые позволят им использовать язык программирования Python для моделирования различных ситуаций и решения задач.

Уроки, на которых изучаются основы языка программирования Python, позволят решать следующие задачи:

– образовательные – изучить основные понятия языка программирования Python; ознакомить с основными типами данных, переменными и способами их создания; научить разрабатывать программы с использованием алгоритмических конструкций (линейные, ветвления, цикл и т.д.);

– развивающие – развитие мыслительной деятельности, речи и алгоритмического стиля мышления;

– воспитательные – воспитание эмоционально-положительной направленности на практическую деятельность, интереса к информатике, рост личной ответственности за результаты своей работы.

**Используемые на уроке средства ИКТ:** компьютеры для обучающихся с доступом в сеть Интернет, мультимедийный проектор с экраном, веб-сервис Microsoft MakeCode, приложение Classroom Mode for Minecraft (режим класса) и игровая среда Minecraft EDU.

Продолжительность урока – 45 минут, учитывая, что непрерывная работа за персональным компьютером или ноутбуком для девятого класса не

превышает 30 минут, согласно Санитарно-эпидемиологические требования к организациям воспитания и обучения [34].

На рисунке 9 отражены планируемые результаты учащихся после изучения раздела «Основы языка программирования Python» в 9 классе.



Рисунок 9 – Результаты изучения раздела «Основы языка программирования Python»

В процессе реализации урока по изучению основ языка программирования Python с использованием геймифицированной платформы Minecraft EDU деятельность учителя и обучающихся будет происходить по сценарию, представленному на рисунке 10.

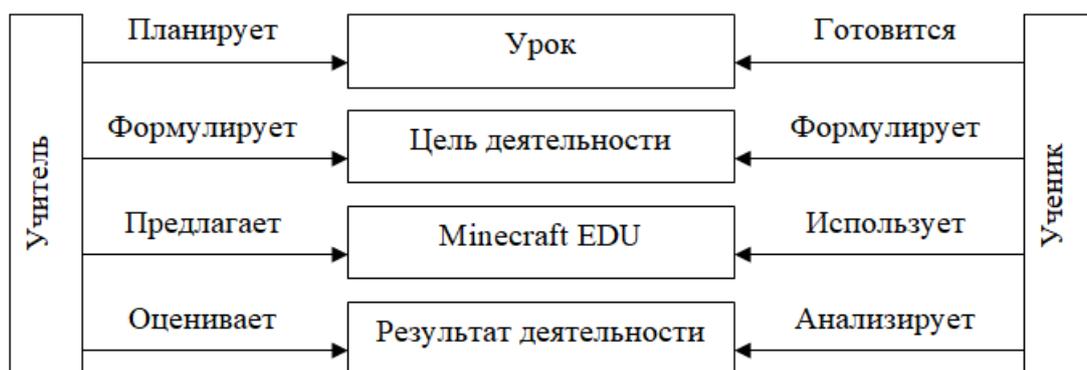


Рисунок 10 – Сценарий организации процесса обучения

Учитель:

- планирует урок: подбирает учебный материал и практические задания для учащихся – разрабатывает миры в платформе Minecraft EDU;
- планирует примерные результаты деятельности и средства, которые необходимо использовать для достижения планируемых результатов;
- формулирует цель деятельности и определяет задачи на урок;
- предлагает ученикам изучить теоретический материал и выполнить задания в игровой среде Minecraft EDU;
- оценивает деятельность учеников и сопоставляет полученные результаты с планируемыми.

Ученик:

- готовится к уроку (повторяет пройденный материал, выполняет домашнее задание);
- формулирует цель деятельности и вместе с преподавателем определяет задачи на урок;
- использует средства, которые предложил учитель и выполняет предложенные задания;
- анализирует результаты своей деятельности, а также оценивает результаты работы одноклассников.

На рисунке 11 представлены разработанные миры в игровой среде Minecraft EDU, согласно тематическому планированию, представленного в

параграфе 2.1. В каждом мире присутствует игровой сюжет, теоретический материал (доски) и интересные, нестандартные задания по каждой теме. Задания по изучению учебного материала разработаны в логике от простого к сложному.

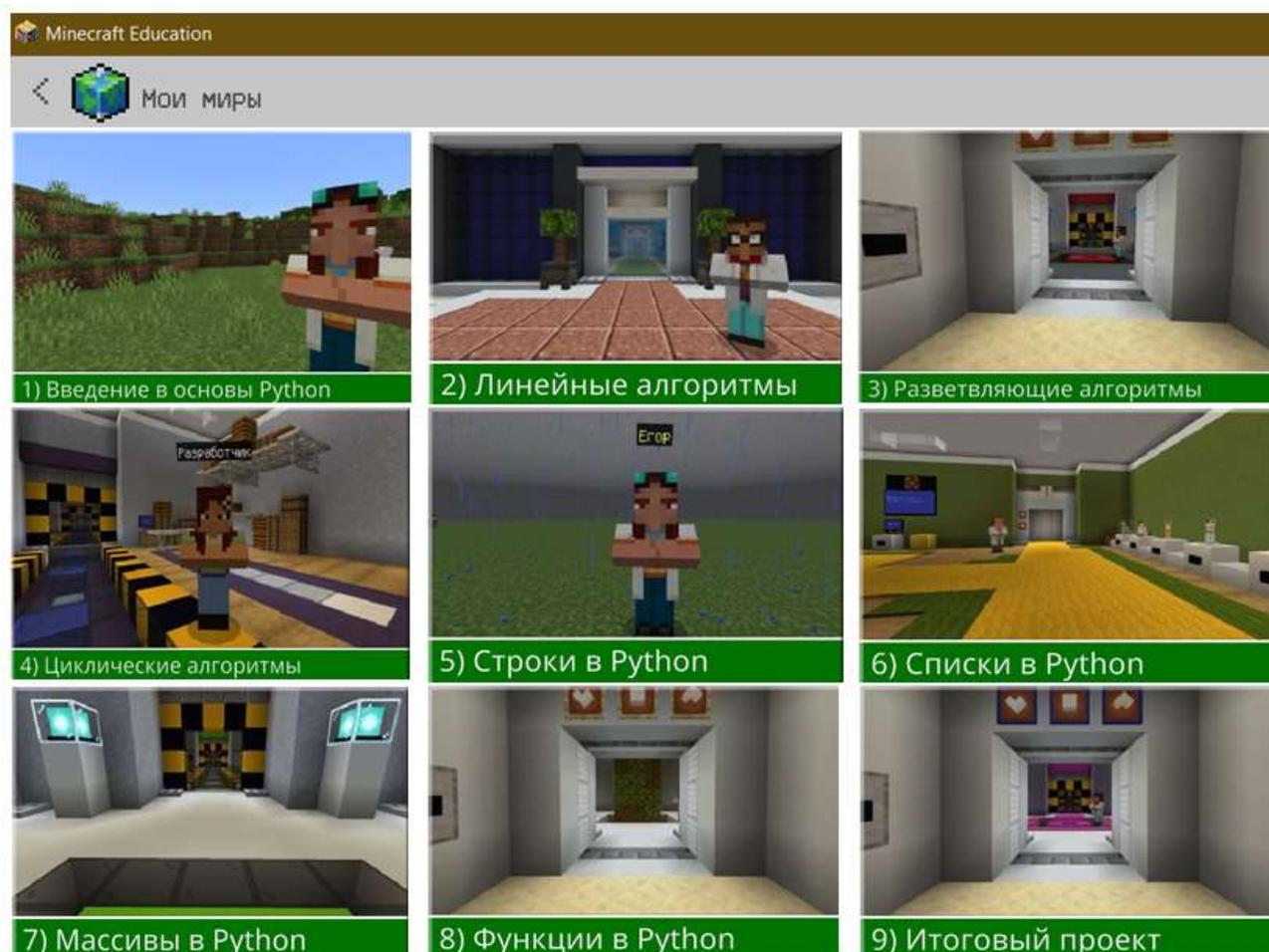


Рисунок 11 – Общая структура разработанного ресурса в Minecraft EDU

Рассмотрим фрагмент урока по теме «Программирование циклических алгоритмов» из раздела «Основы языка программирования Python» с использованием игровой среды Minecraft EDU, основанной на технологии геймификации.

**Тема урока:** Программирование циклических алгоритмов (операторы структуры цикла, счетчик (for), цикл с условием (while), разработка программ циклической структуры).

**Структура урока:**

1. Организационный момент (1 минута);
2. Постановка целей урока (3 минуты);
3. Изучение нового материала (10 минут);
4. Практическая работа (25 минут);
5. Рефлексия и подведение итогов (6 минут).

На организационном этапе происходит приветствие класса, проверка готовности к уроку, фиксация отсутствующих. Затем, преподаватель вместе с учеником ставит цель, и определяют задачи урока.

Рассмотрим этап изучение нового материала.

В программировании, циклы являются важнейшим инструментом, позволяющим выполнять одну и ту же операцию множество раз. Вместо того, чтобы многократно копировать один и тот же фрагмент кода, его можно описать однажды и запускать через цикл. Преимущество заключается не только в повышении эффективности разработки, но также и в возможности легко изменять количество повторений, что делает программу более гибкой и адаптивно. Циклы используются для повторения кода столько раз, сколько требуется.

**Цикл** – это многократное выполнение одинаковых действий либо непрерывно, либо определенное количество раз.

Мы не задумываемся над тем, что наши тела повторяют многие действия. Дыхание, сон, еда, питье, и ходьба – все это действия, которые повторяются ежедневно. Какие еще примеры циклов или повторения есть в вашей повседневной жизни? Какие задачи и действия состоят из повторяющихся шагов? Что это за шаги?

**Циклический алгоритм** – алгоритм, содержащий последовательность действий, многократно повторяющуюся в процессе его выполнения.

**Оператор цикла** – команда, реализующая на языке программирования алгоритмическую конструкцию «повторение».

В таблице 9 рассмотрены основные операторы цикла в языке программирования Python.

Таблица 9 – Основные операторы цикла в языке программирования Python

Операторы цикла	Цикл <b>for</b> (цикл со счетчиком или цикл с переменной)	Цикл <b>While</b> с условием	
		Цикл <b>while</b> с предусловием	Цикл <b>while</b> с постусловием
Общий вид	<p><b>for</b> &lt;параметр&gt; <b>in</b> <b>range (k, n, m):</b>                      &lt;оператор&gt;</p> <p>Где &lt;параметр&gt; – переменная целого типа;  <b>range()</b> – диапазон значений цикла <b>for</b>;  <b>k</b> – начальное значение переменной, <i>по умолчанию равно 0</i>;  <b>n</b> – конечное значение переменной, <i>не включая последнее</i>;  <b>m</b> – шаг изменения переменной, <i>по умолчанию равен 1</i></p>	<p><b>while</b> &lt;условие&gt; :                      &lt;тело цикла&gt;</p> <p>Где &lt;условие&gt; – логическое выражение; пока оно истинно, то выполняется тело цикла; &lt;тело цикла&gt; – последовательность действий (операторов). Каждое выполнение тела цикла называется итерацией</p>	<p><b>while True:</b>                      &lt;операторы&gt;  <b>if</b> &lt;условие&gt;: <b>break</b></p> <p>Команда <b>break</b> (англ. – прервать) позволяет выйти из цикл <b>while</b> в любой момент</p>
Описание	После каждого выполнения тела цикла происходит увеличение на единицу параметра цикла; условие выхода из цикла – достижение параметром конечного значения	Проверка выполняется на входе в цикл. При каждой итерации цикла <b>while</b> условие проверяется, и если оно возвращает <b>True</b> , то тело цикла выполняется снова. Если же условие возвращает <b>False</b> , то выполнение цикла прекращается и управление передается на следующую инструкцию после тела цикла	При входе в цикл условие не проверяется и цикл выполняется хотя бы один раз. Цикл будет выполняться бесконечно, потому что по умолчанию условие <b>True</b> всегда истинно. Выйти из такого цикла можно только с помощью оператора <b>break</b>
Блок-схема			
Пример	<pre>for i in range(1, 10, 2):     player.say(«:»)</pre>	<pre>k = 1 While k &lt;= 5:     player.say(«:»)</pre> <pre>k += 1 player.say(«Конец»)</pre>	<pre>while True:     k = 0     player.say(«:»)</pre> <pre>if k &gt; 0: break</pre>

*Примечание.* Отступы в Python используются, когда мы хотим объявить, что фрагмент кода принадлежит приведенной выше команде. Чтобы идентифицировать фрагмент кода, используйте кнопку tab.

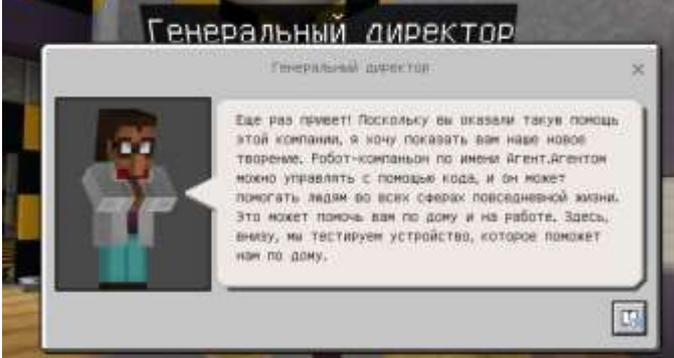
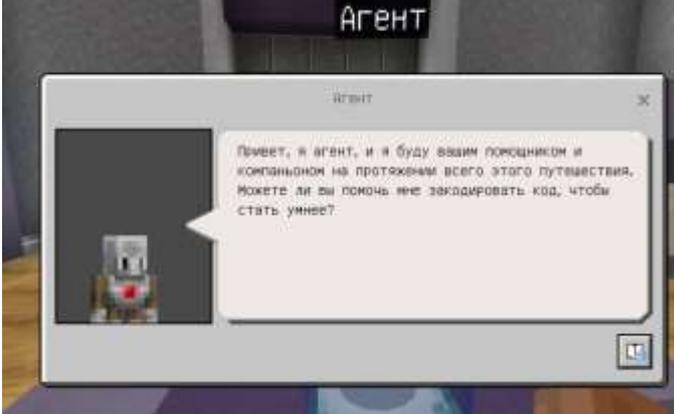
**Вложенные циклы.** При кодировании один цикл может быть помещен (вложен) внутри (в теле) другого цикла, который называется вложенным циклом. Количество вложенных циклов может быть больше двух. При кодировании вложенные циклы – отличный способ выполнять более сложные задачи.

После изучения теоретического материала учащиеся выполняют практическую работу в игровой среде Minecraft EDU, представленную в таблице 10. А учитель отслеживает процесс работы каждого ученика в «режиме класса» через приложение Classroom Mode for Minecraft.

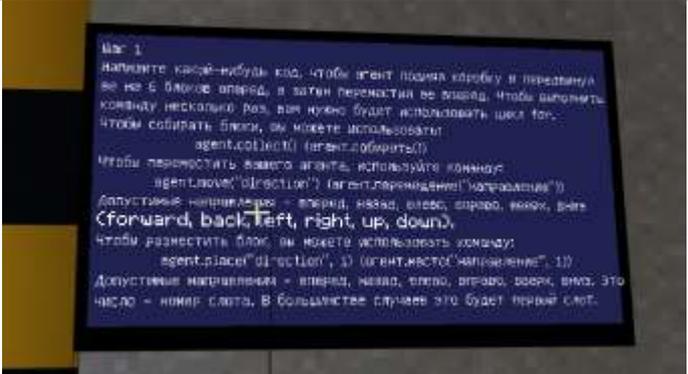
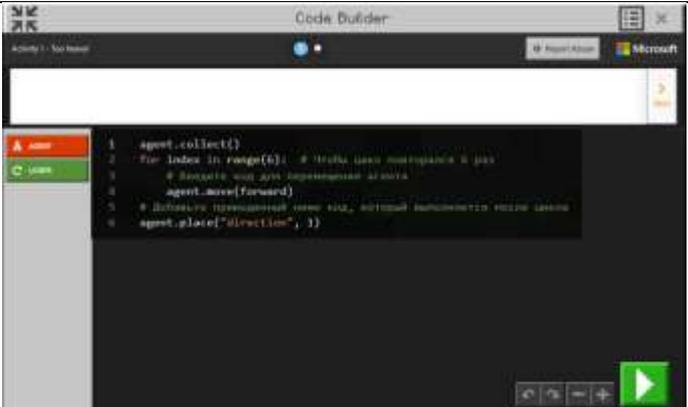
Таблица 10 – Практическая работа в игровой среде Minecraft EDU

Описание	Иллюстрации
<p>Добро пожаловать! Сегодня вы будете работать над сверхсекретным проектом под названием «Агент». Агент – это робот, которым можно управлять с помощью кода для выполнения широкого спектра действий. На этом уроке вы будете кодировать агента для выполнения различных работ по дому. Вы будете использовать различные команды для управления агентом: <code>agent move</code> (перемещение) агента, <code>agent place</code> (размещение агента), <code>agent collect all</code> (сбор всех агентов) и <code>agent drop all</code> (удаление всех агентов). Это ваша точка возрождения, место, где вы начинаете игру</p>	
<p>Пройдите в следующую комнату и поговорите с генеральным директором</p>	

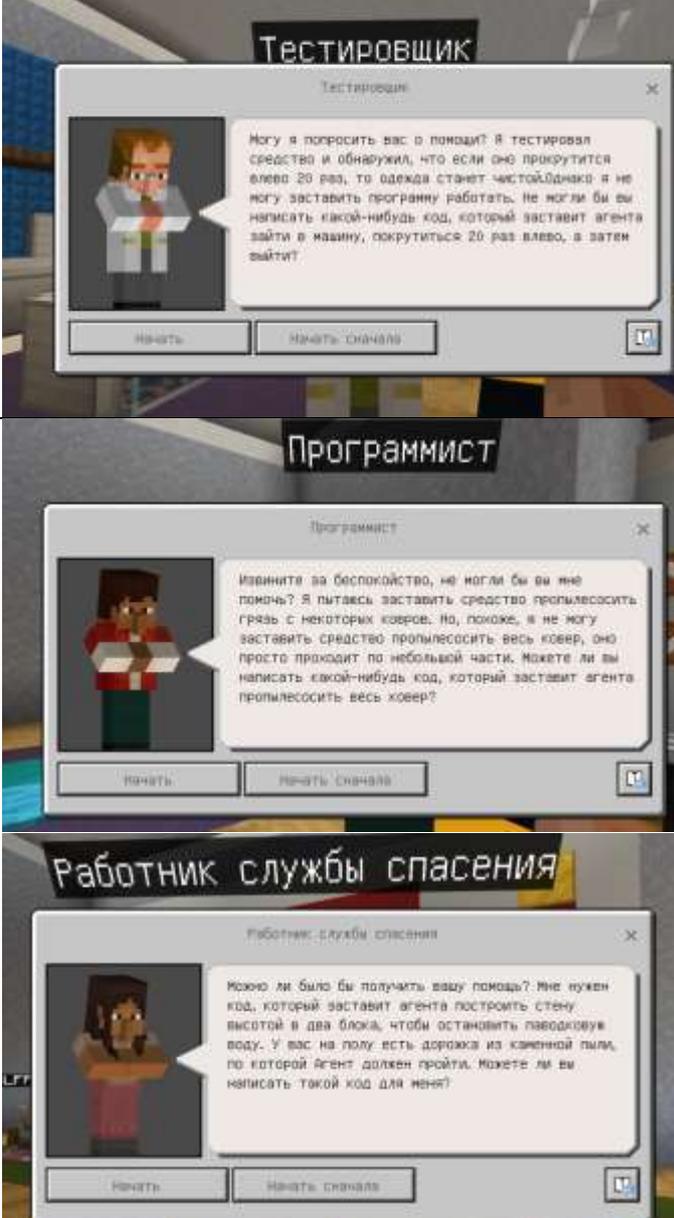
Продолжение таблицы 10

Описание	Иллюстрации
<p>Это всплывающее окно, которое вы увидите на нашем экране. Прочитайте сообщение от главного директора. После того, как вы прочтете сообщение, нажмите на «X» в правом верхнем углу, чтобы продолжить игру</p>	
<p>Потяните за рычаг, чтобы появился агент</p>	
<p>Прочитайте сообщение от Агента. После того, как вы прочтете сообщение, нажмите на «X» в правом верхнем углу, чтобы продолжить игру</p>	
<p>Переходите в следующую комнату</p>	

Продолжение таблицы 10

Описание	Иллюстрации
<p><b>Упражнение 1.</b> Ваше первое задание требует от вас закодировать агента для поднятия тяжелых предметов по дому</p>	
<p>Обратите внимание, что на стенах расположены доски с теоретическим материалом по теме урока. А также они помогут для выполнения задания</p>	
<p>Закодируйте агента, чтобы он забрал коробку, а затем поместил ее в отмеченную область. Когда вы запустите код, агент должен взять коробку, продвинуться вперед и поместить ее. <i>Примечание:</i> Агенту не обязательно сталкиваться лицом к лицу с товаром, чтобы забрать его</p>	
<p>Протестируйте свой код</p>	

## Окончание таблицы 10

Описание	Иллюстрации
<p>Затем проходите в следующие комнаты и выполняйте задания. После выполнения всех заданий, можно поиграть в свободном мире Minecraft, если осталось свободное время</p>	 <p>The illustrations show three dialogue boxes from the game. The first, titled 'Тестировщик', shows a character in a white lab coat asking for help with a program. The second, titled 'Программист', shows a character in a red and white striped shirt asking for help with a stain. The third, titled 'Работник службы спасения', shows a character in a pink shirt asking for help with a wall. Each box has 'Назад' (Back) and 'Начать сначала' (Start over) buttons.</p>

На заключительном этапе проводится подведение итогов, проверка усвоенного материала через рефлексивную деятельность и выдается задание на дом.

Разработанное методическое сопровождение для изучения основ языка программирования Python с использованием геймифицированной игровой среды Minecraft EDU, позволит ученикам пошагово изучать раздел «Основы языка программирования Python» вместе с преподавателем, но также могут и самостоятельно проверить степень уже усвоенных знаний.

Таким образом, можно сделать вывод, что применения геймифицированные платформы в обучении основам языка программирования Python дают возможность использования игровой мотивации детей для привлечения и удержания внимания учащихся в процессе изучения темы урока. Стоит отметить, что данный подход к обучению еще находится в стадии разработки и не может претендовать на полную всестороннюю оценку. Недостаточность перечня игровых обучающих механик также ограничивает его применение. Тем не менее, данный подход впервые сопоставляет интересы и мотивацию обучающихся с интересами и целями педагогов, что может значительно способствовать улучшению использования технологии геймификации в образовании.

### **2.3 Результаты опытно-экспериментальной работы по организации обучения основам языка программирования Python с использованием игровой среды, реализованной на основе технологии геймификации**

В рамках настоящего исследования проведена опытно-экспериментальная работа в МБОУ «СОШ № 1» г. Лесосибирска с обучающимися 9А (20 человек) и 9Б (20 человек) классов в течение марта и апреля 2023 учебного года.

#### **1 Этап: Констатирующий**

Цель: анализ УМК и составление тематического планирования по изучению основ языка программирования Python в 9 классе.

Разработано методического сопровождение игровой среды Minecraft EDU, представленное в п. 2.2 для изучения основ языка программирования Python. Также на первом этапе определены экспериментальная (9А класс) и контрольная (9Б класс) группы.

#### **2 Этап: Формирующий**

Цель: организация образовательного процесса с использованием игровой среды Minecraft EDU, основанной на технологии геймификации, для обучения

основам языка программирования Python и выявление степени обученности класса.

Для достижения цели данного этапа организована работа по изучение раздела «Основы языка программирования Python». В экспериментальной группе организация образовательного процесса строилась с использованием игровой среды Minecraft EDU, основанной на технологии геймификации, а в контрольной группе – «традиционным» для детей образом.

### 3 Этап: Контрольный

Цель: выявление степени обученности класса, после изучения раздела «Основы языка программирования Python».

На данном этапе для выявления уровня знаний и умений проводится контрольное тестирование по разделу «Основы языка программирования Python». Вариант теста представлен в таблице А.1 (приложения А).

Каждый правильный ответ предполагает 1 балл. В сумме наивысшая оценка – 10 баллов. Баллы и оценки, полученные учащимися, представлены в таблице 11.

Таблица 11 – Результаты тестирования

№	Имя, Ф.	№ вопроса с правильными ответами	Балл	Оценка
Контрольная группа				
1.	Анна А.	1,2,3,4,5,6,7,8,9,10	10	5
2.	Александр Б.	1,2,3,4,5,6,7,8,10	9	5
3.	Николай Б.	1,2,3,4,6,7	6	3
4.	Анастасия В.	1,2,3,4,5,6,7,8,9,10	10	5
5.	Николай В.	1,2,3,4,5,6,7	7	4
6.	Мария Г.	1,2,3,4,5,6,7,8,9,10	10	5
7.	Жанна З.	1,2,3,4,5,6,7,8,9	9	5
8.	Анатолий И.	1,2,3,6,7,8,9	7	4
9.	Наталия И.	1,2,3,4,5	5	3
10.	Михаил К.	1,2,3,4,5,6,7,8,9	9	5
11.	Богдан Л.	1,2,3,4,5,7,8	7	4
12.	Сергей М.	1,2,3,4,5,6,7,9	8	4
13.	Павел М.	1,2,3,5,6,7	6	3
14.	Алена Н.	1,2,3,4,5,6,8,10	8	4
15.	Марина С.	1,2,3,4,5,6,7	7	4
16.	Петр С.	1,2,3,4,5,6,7,10	8	4
17.	Елена Т.	1,2,3,4,6	5	3
18.	Владимир Т.	1,2,3,4,5,6,7,8,9	9	5
19.	Сергей Т.	1,2,3,5,7	5	3
20.	Вениамин Я.	1,2,3,5,7,8,9	7	4

Окончание таблицы 11

№	Имя, Ф.	№ вопроса с правильными ответами	Балл	Оценка
Экспериментальная группа				
1.	Анастасия А.	1,2,3,4,5,6,7,8,9,10	10	5
2.	Анна А.	1,2,3,4,5,6,9,10	8	4
3.	Юрий Б.	1,2,3,4,5,6,7	7	4
4.	Иван В.	1,2,3,4,5,6,7,8	8	4
5.	Ирина Г.	1,2,3,4,5,6,7,8,9	9	5
6.	Василий И.	1,2,3,4,5,6,7,8,9,10	10	5
7.	Оксана И.	1,2,3,4,5	5	3
8.	Павел К.	1,2,3,4,5,6,7,8,9,10	10	5
9.	Алексей М.	1,2,3,4,5,6,7,10	8	4
10.	Андрей Н.	1,2,3,4,5,6,7	7	4
11.	Дмитрий П.	1,2,3,4,5,6,7,9,10	9	5
12.	Елена П.	1,2,3,4,5,6,7,9	8	4
13.	Инна П.	1,2,3,4,5,6,7,8,9,10	10	5
14.	Роман Р.	1,2,3,4,5,6,7,8,9,10	10	5
15.	Сергей С.	1,2,3,5,6,7,8	7	4
16.	Дмитрий С.	1,2,3,4,6,7	6	3
17.	Анжелика Т.	1,2,3,4,5,6,7,8	8	4
18.	Лилия Т.	1,2,3,4,5,6,9,10	8	4
19.	Петр Ч.	1,2,3,4,5,6,7,8,9,10	10	5
20.	Сергей Ф.	1,2,3,4,5,7,8,9	8	4

Количественный результат контрольной диагностики представлен на рисунке 12.

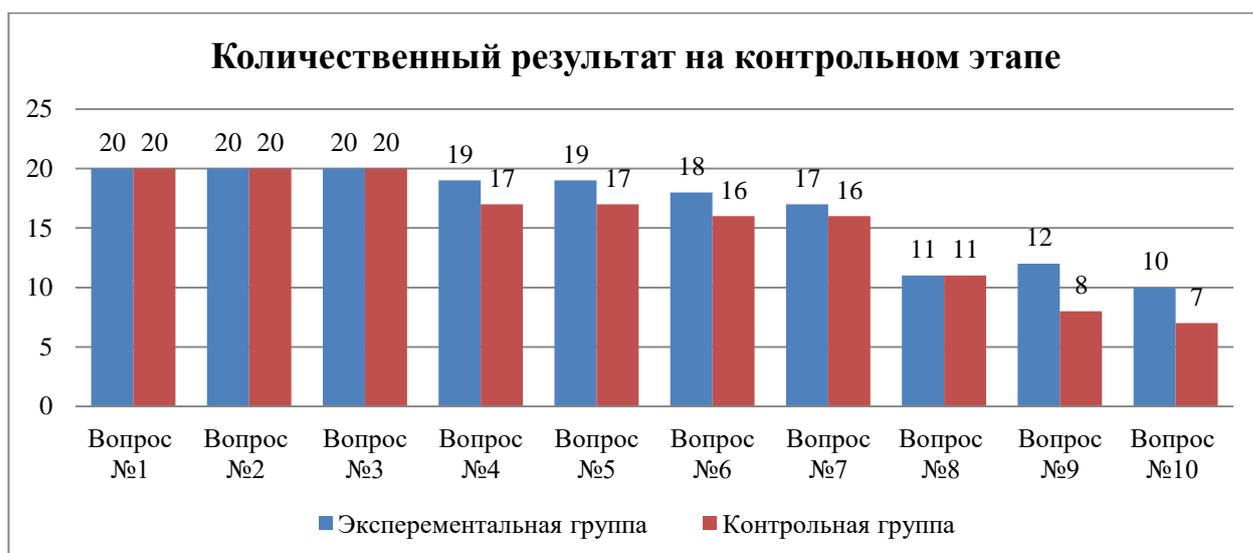


Рисунок 12 – Результаты контрольной диагностики

Для более точного описания результатов опытно-экспериментальной работы, определили три уровня овладения знаниями по разделу «Основы языка программирования Python». Каждый из этих уровней оценивается по 10-ти балльной шкале.

– высокий уровень овладения соответствует от 9 до 10 баллов, что свидетельствует о понимании и полном усвоении изученного материала. Учащиеся на этом уровне успешно прошли от 90% до 100% тестовых заданий предложенного теста.

– средний уровень овладения знаниями соответствует оценке от 6 до 8 баллов. На этом уровне учащиеся успешно прошли от 60% до 80% тестовых заданий, что свидетельствует о неполноценном, но достаточном для дальнейшего обучения, усвоении материала.

– низкий уровень овладения знаниями соответствует оценке от 1 до 5 баллов. Учащиеся, получившие такую оценку, успешно прошли от 10% до 50% тестовых заданий. Это свидетельствует о том, что учащиеся не понимают тему достаточно хорошо, не достигли желаемых результатов и, возможно, нуждаются в дополнительных занятиях по теме.

Количественные результаты показателей каждого уровня представлены на рисунке 13.

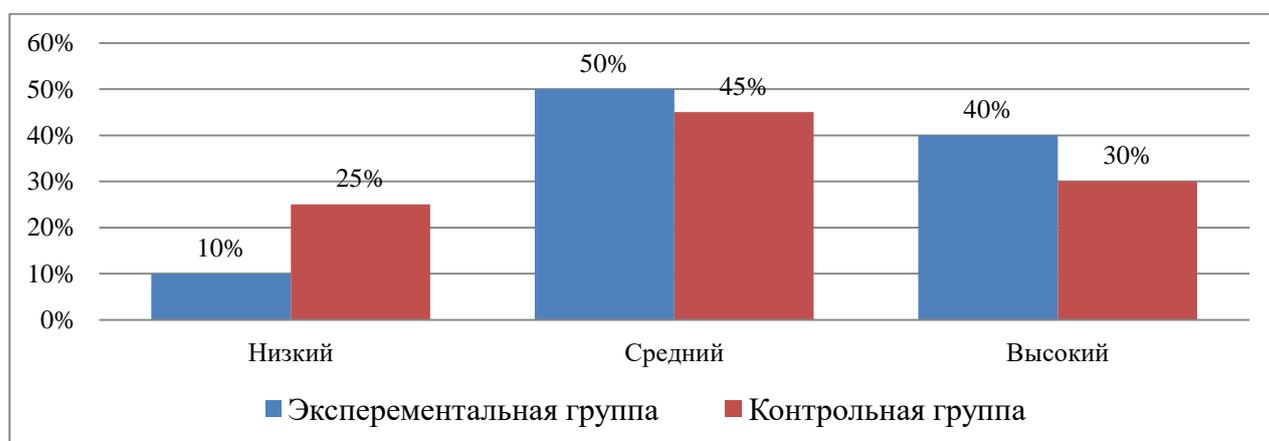


Рисунок 13 – Количественные результаты контрольной диагностики по уровню овладения знаниями

Из рисунка 13 следует, что у обучающихся экспериментальной группы высокий уровень показали 40% учащихся, средний уровень у 50% детей и на низком уровне усвоения знаний находится 10% обучающихся. У обучающихся контрольной группы высокий уровень составил 30% учащихся, средний уровень 45% обучающихся и низкий уровень показали 25% учащихся.

На основе результатов тестирования, в таблице 12 рассчитано значение степени обученности класса по формуле (1) для экспериментальной и контрольной групп.

$$K = \frac{(N_5 + 0,64 * N_4 + 0,36 * N_3 + 0,16 * N_2)}{N} \quad (1)$$

Где  $N$  – общее количество учащихся класса,  $N_k$  – количество учащихся, получивших оценку  $k$ .

Таблица 12 – Степень обученности класса

Контрольная группа		Экспериментальная группа	
Оценка	Количество	Оценка	Количество
«5»	6	«5»	8
«4»	9	«4»	10
«3»	5	«3»	2
«2»	0	«2»	0
Степень обученности класса			
<b>67,8%</b>		<b>75,6%</b>	

Анализ степени обученности класса показал, что использование игровой среды Minecraft EDU, основанной на технологии геймификации, положительно влияет на уровень усвоения материала учащимися.

Полученные результаты показывают, что организация образовательного процесса по изучению раздела «Основы языка программирования Python» с использованием игровой среды Minecraft EDU, основанной на технологии геймификации, эффективно, так как в экспериментальной группе степень обученности класса выше на 8% чем в контрольной группе, где образовательный процесс был построен «стандартным» образом.

Таким образом, использование технологий геймификации в основной школе при обучении основам языка программирования Python, способствует раскрытию индивидуальных особенностей, самореализации обучающихся, повышению их самооценки, уверенности в себе, формированию и развитию у обучающихся исследовательских умений и повышению качества знаний, мотивации и интереса к предмету «Информатика и ИКТ».

## ЗАКЛЮЧЕНИЕ

В заключение работы отметим следующее. Теоретические основы технологии геймификации в формировании игровой среды для обучения основам языка программирования Python в основной школе раскрыты, а также разработано и апробировано методическое сопровождение теоретической и практической составляющих игровой среды, реализованной на основе технологии геймификации, для обучения основам языка программирования Python в 9 классе.

В ходе работы решены следующие задачи.

1. Теоретический обзор научной и учебно-методической литературы по теме исследования позволил определить, что геймификация является новым способом организации обучения, имеющим огромный педагогический потенциал. Геймификация характеризуется как использование игровых элементов в неигровой ситуации, т. е. процесс, когда элементы игры используют для достижения реальных целей. Важно, что геймификация – это не создание полноценной игры, а только использование определенных элементов. Целью этой технологии является привлечение и повышение внимания обучающихся, интереса к изучаемому предмету и улучшения мотивации при решении практических задач.

2. На основе анализа игровых сред и их возможностей в обучении основам программирования; анализа УМК по разделу «Алгоритмизация и программирование» предмета информатики основной школы, разработали методическое сопровождение теоретическое и практической составляющих игровой среды Microsoft EDU, реализованной на основе технологии геймификации, имеющей дидактический потенциал для изучения основ языка программирования Python. Данное методическое сопровождение позволит ученикам пошагово изучать раздел «Основы языка программирования Python» не только вместе с преподавателем, но также дети могут самостоятельно проверить степень уже усвоенных знаний.

3. Опытнo-экспериментальная работа организована и проведена МБОУ «СОШ № 1» г. Лесосибирска. Основной целью работы являлось определение эффективности обучения основам языка программирования Python с использованием игровой среды, реализованной на основе технологии геймификации. Для достижения данной цели сформированы экспериментальная и контрольная группы. Организация обучения с использованием игровой среды, реализованной на основе технологии геймификации на уроках информатики в 9 классе дала положительные результаты при изучении раздела «Основы языка программирования Python», об этом свидетельствует степень обученности класса.

Следовательно, использование технологий геймификации в основной школе при обучении основам языка программирования Python, способствует раскрытию индивидуальных особенностей, самореализации обучающихся, повышению их самооценки, уверенности в себе, формированию и развитию у обучающихся исследовательских умений и повышению качества знаний, мотивации и интереса к предмету «Информатика и ИКТ».

Таким образом, задачи, поставленные перед исследованием, успешно выполнены, цель достигнута.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Белоконова, С. С. Геймификация как способ организации обучения на уроках информатики / С. С. Белоконова, В. В. Назарова // Актуальные проблемы методики обучения информатике в современной школе : материалы Международной научно-практической интернет-конференции / Московский педагогический государственный университет. – Москва, 2020. – С. 45–54.
2. Белоконова, С. С. Использование веб-сервисов геймификации для организации обучения на уроках информатики / С. С. Белоконова, В. В. Назарова // Информатика в школе. – 2020. – № 7. – С. 30–37.
3. Белоконова, С. С. Современные веб-технологии в образовательном процессе и их роль в обучении / С. С. Белоконова, В. В. Назарова // Вестник Таганрогского института имени А. П. Чехова. – 2018. – № 1. – С. 3–7.
4. Борисов, Н. Г. Использование геймификации при обучении школьников основам программирования / Н. Г. Борисов, А. С. Харюнин // Образовательные технологии и общество. – 2018. – Т. 21, № 1. – С. 469–476.
5. Босова, Л. Л. Информатика. 9 класс : учебник / Л. Л. Босова, А. Ю. Босова. – Москва : БИНОМ. Лаборатория знаний, 2017. – 208 с. – ISBN 978-5-9963-3045-4.
6. Босова, Л. Л. Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе / Л. Л. Босова // Информатика в школе. – 2018. – № 6. – С. 3–11.
7. Вербах, К. Вовлекай и властвуй. Игровое мышление на службе бизнеса / К. Вербах, Д. Хантер ; пер. с англ. А. Кардаш. – Москва : Манн, Иванов и Фербер, 2015. – 224 с. – ISBN 978-5-00057-344-0.
8. Герасимова, Е. К. Исследование условий включения элементов геймификации в изучение основ алгоритмизации и программирования для повышения качества образовательных результатов обучающихся / Е. К. Герасимова, Н. В. Гавриловская, Е. В. Соболева [и др.] // Перспективы науки и образования. – 2021. – № 4. – С. 461–477.

9. Гудин, Ю. В. Урок + игра : Современные игровые технологии для школьников : методическое пособие для учителей / Ю. В. Гудин. – Москва : Речь, 2010. – 158 с. – ISBN 978-5-9268-0862-6.
10. Ершов, А. П. Программирование – вторая грамотность / А. П. Ершов // Журнал Института вычислительной математики и математической геофизики Сибирского отделения РАН. Проблемы информатики. – 2015. – № 4. – С. 71–85.
11. Заславская, О. Ю. Применение принципов игрового дизайна и игровых механик к неигровому контенту / О. Ю. Заславская, А. Г. Сиденко // Вестник Московского городского педагогического университета. Серия: «Информатика и информатизация образования». – 2020. – № 1. – С. 30–33.
12. Идрисова, А. Геймификация в образовании / А. Идрисова // Сколки – мысли о неформальном образовании. – 2020. – URL: <https://skolki-project.com/blog/gejmifikatsija-v-obrazovanii>. (дата обращения: 29.04.2023).
13. Идрисова, А. Что такое геймификация? / А. Идрисова // Сколки – мысли о неформальном образовании. – 2020. – URL: <https://skolki-project.com/blog/chto-takoe-gejmifikatsija> (дата обращения: 29.04.2023).
14. Кавтарадзе, Д. Н. Обучение и игра : введение в интерактивные методы обучения / Д. Н. Кавтарадзе. – Москва : Просвещение, 2018. – 200 с. – ISBN 978-5-09-019851-6.
15. Капп, К. М. Геймификация обучения и методических пособий: идеи на практике / К. М. Капп. – Нью-Йорк : John Wiley & Sons, 2012. – 336 с. – ISBN 978-1-118-09634-5.
16. Краснова, Г. А. Электронное образование в эпоху цифровой трансформации : [монография] / Г. А. Краснова, Г. В. Можяева ; Томский государственный университет. – Томск : Издательский Дом Томского государственного университета, 2019. – 200 с. – ISBN 978-5-94621-813-9.
17. Лапчик, М. П. Методика обучения информатике : учебное пособие / М. П. Лапчик, М. И. Рагулина, И. Г. Семакин [и др.]. – Санкт-Петербург : Лань, 2016. – 392 с. – ISBN 978-5-8114-1934-0.

18. Ляпина, Г. А. Теория и практика игр : учебно-методическое пособие / Г. А. Ляпина. – Уфа : Академия ВЭГУ, 2009. – 159 с. – ISBN 978-5-87865-450-0.
19. Мазелис, А. Л. Геймификация в электронном обучении / А. Л. Мазелис // Вестник Владивостокского государственного университета. Территория новых возможностей. – 2013. – № 3. – С. 139–142.
20. Мацуца, К. И. Некоторые аспекты применения компьютерных игр на уроках информатики / К. И. Мацуца // Образовательные технологии XXI века : материалы VII городской научно-практической конференции / Московский педагогический государственный университет. – Москва, 2008. – С. 102–106.
21. Мишин, В. А. Геймификация на уроке информатики на основе платформы Minecraft EDU / В. А. Мишин // Актуальные проблемы обучения математике и информатике в школе и вузе : материалы VI Международной интернет-конференции / Московский педагогический государственный университет. – Москва, 2021. – С. 107–113.
22. Мишин, В. А. Использование платформы Minecraft: Education Edition на уроках информатики / В. А. Мишин // Информатика в школе. – 2021. – № 3. – С. 50–59.
23. Никитин, С. И. Геймификация, игрофикация, играизация в образовательном процессе / С. И. Никитин // Молодой ученый. – 2016. – № 9. – С. 1159–1162.
24. Олейник, Ю. П. Игрофикация в образовании : к вопросу об определении понятия / Ю. П. Олейник // Современные проблемы науки и образования. – 2015. – № 3. – С. 476–488. – URL: <https://science-education.ru/ru/article/view?id=20103> (дата обращения: 20.04.2023).
25. Омарова, Г. Р. Современные языки программирования при обучении программированию школьников / Г. Р. Омарова, И. В. Шимов // Актуальные вопросы преподавания математики, информатики и информационных технологий. – 2018. – № 3. – С. 270–275.

26. Орлова, О. В. Геймификация как способ организации обучения / О. В. Орлова, В. Н. Титова // Вестник Томского государственного педагогического университета. – 2015. – № 9 – С. 60–64.

27. Павлов, Д. И. О перспективах использования технологий геймификации при раннем обучении объектно-ориентированному программированию / Д. И. Павлов, К. В. Бутарев, Е. В. Балашова // Современные информационные технологии и ИТ-образование. – 2018. – Т. 14, № 4. – С. 977–985.

28. Певзнер, В. В. Некоторые особенности применения геймификации в процессе обучения / В. В. Певзнер, В. И. Погорелов, Д. А. Шуклин // Проблемы современного образования. – 2016. – № 2. – С. 98–101.

29. Поляков, К. Ю. Информатика. 9 класс : учебник / К. Ю. Поляков, Е. А. Еремин. – Москва : БИНОМ. Лаборатория знаний, 2019. – 288 с. – ISBN 978-5-9963-4587-8.

30. Приказ Министерства образования и науки Российской Федерации от 17 декабря 2010 г. №1897 (с изменениями на 8 ноября 2022 года) «Об утверждении федерального государственного образовательного стандарта основного общего образования» // Электронный фонд правовых и нормативно-технических документов. – URL: <https://docs.cntd.ru/document/902254916> (дата обращения: 15.02.2023).

31. Примерная основная образовательная программа среднего общего образования (одобрена решением федерального учебно-методического объединения по общему образованию (протокол от 28 июня 2016 г. N 2/16-з)) // Гарант: справочная правовая система. – URL: <https://www.garant.ru> (дата обращения: 28.04.2023).

32. Распоряжение Министерства просвещения РФ от 18 мая 2020 г. № Р-44 «Об утверждении методических рекомендаций для внедрения в основные общеобразовательные программы современных цифровых технологий» // Электронный фонд правовых и нормативно-технических

документов. – URL: <https://docs.cntd.ru/document/565227683> (дата обращения: 15.02.2023).

33. Ричардсон, К. Програмуем с Minecraft. Создай свой мир с помощью Python / К. Ричардсон ; научный редактор Г. А. Гаджиев. – Москва : Манн, Иванов и Фербер, 2017. – 368 с.

34. СП 2.4.3648-20. Санитарные правила. Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи : утвержден Главным государственным санитарным врачом Российской Федерации от 28.09.2020 г. №28 : дата введения 2021-01-01 // Электронный фонд правовых и нормативно-технических документов. – URL: <https://docs.cntd.ru/document/566085656> (дата обращения: 15.02.2023).

35. Семакин, И. Г. Информатика : учебник для 9 класс. / И. Г. Семакин, Л. А. Залогова, С. В. Русаков [и др.]. – Москва : БИНОМ. Лаборатория знаний, 2015. – 200 с. – ISBN 978-5-9963-1938-1.

36. Сиденко, А. Г. Использование стратегий геймификации для мотивации школьников обучению информатике / А. Г. Сиденко // Вестник Московского городского педагогического университета. Серия «Информатика и информатизация образования». – 2020. – № 1. – С. 92–94.

37. Соболева, Е. В. Применение обучающих программ на игровых платформах для повышения эффективности образования / Е. В. Соболева, Н. И. Исупова, Т. Н. Суворова [и др.] // Вестник Новосибирского государственного педагогического университета. – 2017. – Т. 7, № 4. – С. 7–25.

38. Угринович, Н. Д. Информатика : учебник для 9 класса / Н. Д. Угринович. – Москва : БИНОМ. Лаборатория знаний, 2016. – 152 с. – ISBN 978-5-906812-88-9.

39. Храшкин, П. В. Геймифицируй это : как урок превратить в игру / П. В. Храшкин // Онлайн-обучение iSpring. – URL: <https://www.ispring.ru/elearning-insights/gameschool> (дата обращения: 12.02.2023).

40. Эйхорн, М. В. Геймификация образовательного процесса : методическое пособие / М. В. Эйхорн. – Томск, 2015. – 39 с. – URL:

[https://studylib.ru/doc/6263921/gejmifikaciya-obrazovatel\\_nogo-processa](https://studylib.ru/doc/6263921/gejmifikaciya-obrazovatel_nogo-processa) (дата обращения: 14.03.2023).

41. Apostol, S. Gamification of learning and educational games / S. Apostol, L. Zaharescu, I Alexe // Quality and efficiency in e-learning. – 2013. – URL: <http://proceedings.elseconference.eu/index.php?paper=42e250b284110445d15f593e1084a44d> (дата обращения: 10.02.2023).

42. Deterding, S. Gamification : using game-design elements in non-gaming contexts / S. Deterding, M. Sicart, L. Nacke – New York : ACM Press. – 2011. – P. 2425–2428. – URL: <http://doi.org/10.1145/1979742.1979575> (дата обращения: 12.02.2023).

43. Oblinger, D. G. The Next Generation of Educational Engagement / D. G. Oblinger // Journal of Interactive Media in Education. – 2004. – Iss. 8. – P. 1–18. – URL: <http://jime.open.ac.uk/articles/10.5334/2004-8-oblinger> (дата обращения: 12.04.2023).

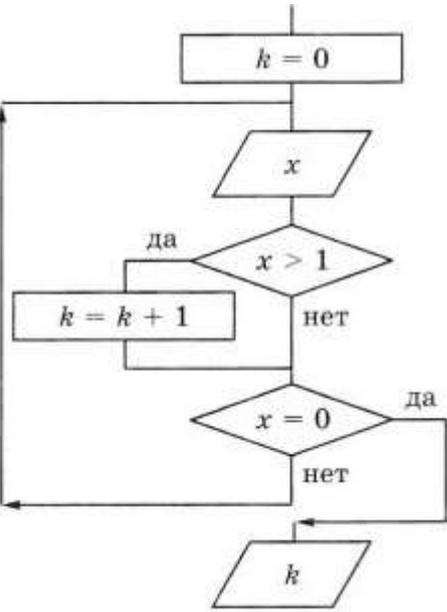
## ПРИЛОЖЕНИЕ А

### Тестовые задания для проведения контрольного этапа опытно-экспериментальной работы

Таблица А.1 – Контрольные вопросы для тестирования и ответы

№ вопроса	Вопросы	Ответы												
1.	<b>Выберите, кто является разработчиком языка Python:</b> а) Норберт Винер б) Джон фон Нейман в) Исаак Ньютон г) Гвидо ванн Россум	г)												
2.	<b>Выберите, что из ниже перечисленного входит в алфавит языка Python:</b> а) латинские строчные и прописные буквы б) служебные слова в) русские строчные и прописные буквы г) знак подчёркивания	а, г)												
3.	<b>Дополните определение:</b> Языковые конструкции, с помощью которых в программах записываются действия, выполняемые в процессе решения задачи, называются ...	операторами												
4.	<b>Для генерации случайного целого числа из отрезка [10, 20] необходимо использовать выражение:</b> а) randint (2*10) б) randint (1020) в) randint ((10, 20)) г) randint (10) *2	в)												
5.	<b>Установите соответствие между названиями типов данных и их обозначениями.</b> <table style="width: 100%; border: none;"> <tr> <td style="text-align: center;"><b>Тип данных</b></td> <td style="text-align: center;"><b>Обозначения</b></td> </tr> <tr> <td>1. Целочисленный</td> <td>а) str</td> </tr> <tr> <td>2. Вещественный</td> <td>б) bool</td> </tr> <tr> <td>3. Строковый</td> <td>в) int</td> </tr> <tr> <td>4. Логический</td> <td>г) float</td> </tr> <tr> <td></td> <td>д) set</td> </tr> </table>	<b>Тип данных</b>	<b>Обозначения</b>	1. Целочисленный	а) str	2. Вещественный	б) bool	3. Строковый	в) int	4. Логический	г) float		д) set	а) – 3) б) – 4) в) – 1) г) – 2)
<b>Тип данных</b>	<b>Обозначения</b>													
1. Целочисленный	а) str													
2. Вещественный	б) bool													
3. Строковый	в) int													
4. Логический	г) float													
	д) set													
6.	<b>Отметьте условия, правильно записанные на языке Python.</b> 1) $a > 0$ 2) $x \leq 3$ 3) $x > 0$ or $y < 0$ 4) $c \neq 0$ 5) $-5 < a < 10$ 6) $x < 0$	2, 3, 5, 6												
7.	<b>Определите значение переменной С после выполнения следующего фрагмента программы:</b> a,b=100,30 a=a-b*3 if a>b: c=a-b else:	20												

Окончание таблицы А.1

№ вопроса	Вопросы	Ответы						
	c=b-a							
8.	<p>Установите правильную последовательность строк программы, чтобы получилась программа, рассчитывающая по двум введенным с клавиатуры вещественным значениям катетов квадрат гипотенузы прямоугольного треугольника.</p> <p>1) C = A*A + B*B                  2) print('Квадрат гипотенузы', C)                  3) A=float (input())                  4) print ('Введите длины катетов')                  5) B=float (input())</p>	4, 3, 5, 1,2						
9.	<p>Решите задачу.                  Дан одномерный массив А из числовых значений, насчитывающий 6 элементов.</p> <table border="1" data-bbox="531 808 1102 853" style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding: 2px 10px;">-125</td> <td style="padding: 2px 10px;">200</td> <td style="padding: 2px 10px;">10</td> <td style="padding: 2px 10px;">5</td> <td style="padding: 2px 10px;">43</td> <td style="padding: 2px 10px;">11</td> </tr> </table> <p>Напишите программу, которая поменяет местами элементы, стоящие на чётных и нечётных местах: A[1] ↔ A[2]; A[3] ↔ A[4]                  ...                  Запишите листинг программы</p>	-125	200	10	5	43	11	
-125	200	10	5	43	11			
10.	<p>Запишите на языке Python фрагмент программы, соответствующий блок схеме</p>  <pre> graph TD     Start(( )) --&gt; K0[k = 0]     K0 --&gt; X[/x/]     X --&gt; Xgt1{x &gt; 1}     Xgt1 -- да --&gt; Kplus[k = k + 1]     Kplus --&gt; X     Xgt1 -- нет --&gt; Xeq0{x = 0}     Xeq0 -- да --&gt; X     Xeq0 -- нет --&gt; Kout[/k/]     Kout --&gt; X     </pre>							