

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ  
УНИВЕРСИТЕТ им.В.П.АСТАФЬЕВА  
(КГПУ им.В.П.Астафьева)

Институт/факультет

Математики, физики и информатики  
(полное наименование института/факультета/факультета)

Выпускающая(ие) кафедра(ы)

Базовая кафедра Информатики и  
информационных технологий в образовании  
(полное наименование кафедры)

Артемьева Ангелина Александровна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема Система мега-уроков по программированию для 10 класса с использованием нелинейных технологий обучения (на примере раздела «Основы алгоритмических конструкций»)

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)

(код направления подготовки)

Профиль физика и информатика  
(наименование профиля для бакалавриата)



зав.кафедрой

(ученая степень, ученое звание, фамилия, инициалы)

ДОПУСКАЮ К ЗАЩИТЕ

Н.И. Пак, д.п.н.,

профессор

(дата, подпись)

Руководитель

к.п.н., Сокольская М.А.

(ученая степень, ученое звание, фамилия, инициалы)

Дата защиты

Обучающийся

Артемьева А.А.

(фамилия, инициалы)

(дата, подпись)

Оценка

(прописью)

Красноярск 2018

Введение.....	3
Глава 1. Нелинейные технологии обучения в образовательном кластере.....	6
1.1. Применение нелинейных технологий обучения в образовательном процессе школы.....	6
1.2 Сущность образовательного кластера (на примере платформы мегакласс) .....	10
1.3. Особенности обучения программированию старшеклассников инженерно-технического профиля в условиях образовательного кластера ...	16
Выводы по главе 1.....	27
Глава 2. Система мега-уроков для 10 класса (на примере раздела «Основы алгоритмических конструкций»)	28
2.1. Цели и задачи обучения школьников программированию в образовательном кластере.....	28
2.2. Содержание и средства обучения программированию на мега-уроках в нелинейных технологиях.....	31
2.3. Формы организации работы обучающихся в сетевом кластере и контроль результатов их деятельности.....	44
Выводы по главе 2.....	58
Заключение .....	60
Список использованных источников .....	61
Приложение А .....	65
Приложение Б.....	79
Приложение В.....	89
Приложение Г .....	97
Приложение Д.....	106
Приложение Е.....	112
Приложение Ж.....	119

## **Введение**

В условиях развития массовой коммуникации и информатизации государства и общества, а также стремительного нарастания нехватки высококвалифицированных программных инженеров в условиях глобализации информационных технологий, существенную роль приобретает подготовка обучающихся в обучении программированию в школьном курсе информатики и ИКТ.

Изучение программирования в школе на сегодняшний день является одной из главных проблем обучения в курсе информатики. Объем часов, предоставленный для изучения темы «Алгоритмизация и программирование», не позволяет в полной мере изучить и углубить знания по данной теме как в базовом, так и в профильном курсе.

Для решения данной проблемы существуют различные методики обучения. Методики подразделяются, в свою очередь, на разнообразные линейные и нелинейные технологии обучения и образовательные платформы, являющиеся неким «трендом» современного образования. Линейные технологии обучения программированию не дают должного результата, поэтому идет активное привлечение в учебный процесс нелинейных методов, в которых создаются условия, где обучающийся становится субъектом процесса приобретения знаний, а, следовательно, и субъектом процесса формирования универсальных учебных действий. Одним из вариантов объединения учебных учреждений в единый сетевой образовательный процесс является технология мега-класс как средство повышения мотивации к познавательной деятельности и формирования основ успешности школьников в условиях ЭО и ДОТ. Применение в образовательном процессе данных технологий и платформ может существенно повысить качество обучения программированию, а также послужит хорошей подготовкой обучающихся к дальнейшему образованию.

Программирование в курсе старшей школы представляет собой весьма сложный раздел информатики, и каждый учитель задается вопросом о

выборе тех или иных методов и средств обучения для обеспечения максимально эффективного усвоения материала. Организация современных методов обучения (исследовательской, поисковой деятельности) при решении проблем позволяет ученикам развивать умения формулировать и проверять предположения, также происходит углубление представлений о возможностях компьютера в автоматизированной обработке и хранении информации. Проведение системы уроков по программированию в образовательном кластере поможет не только повысить мотивацию, развить творческие потенциалы учеников, но и обеспечить более качественное усвоение материала.

**Объект исследования:** процесс обучения информатике в старшей школе.

**Предмет исследования:** система мега-уроков по программированию с использованием нелинейных технологий для 10 класса инженерно-технологического профиля (на примере раздела «Основы алгоритмических конструкций»).

**Цель исследования:** разработать систему мега-уроков по разделу «Основы алгоритмических конструкций» для 10 класса инженерно-технологического образовательного кластера на основе нелинейного обучения.

**Задачи исследования:**

1) проанализировать литературу по теме исследования с целью выделения особенностей нелинейной технологии обучения в применении к программированию и выделения сущности и особенностей организации учебного процесса в образовательном кластере (на примере образовательной платформы Мегакласс);

2) выявить особенности обучения программированию старшеклассников инженерно-технологического профиля;

3) выбрать содержание системы мега-уроков на базе практико-ориентированных задач, отобрать методы, приемы обучения, описать формы

организации обучения по сетевой модели с использованием нелинейных технологий обучения;

4) разработать систему мега-уроков по программированию с использованием нелинейной технологии обучения.

Основа для наблюдения и разработки методических материалов – инженерно-технологический образовательный кластер, созданный базовой кафедрой информатики и информационных технологий КГПУ и м. В.П. Астафьева, включающий образовательные учреждения: КГПУ им. В.П. Астафьева, лицей №1 г. Ачинска, школа №3 г.Ачинска, гимназия №9 г.Красноярска; школа-лицей Туран г.Алматы, Казахстан.

Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка литературы и приложений.

## **Глава 1. Нелинейные технологии обучения в образовательном кластере**

### **1.1. Применение нелинейных технологий обучения в образовательном процессе школы**

Система образования претерпевает процессы совершенствования всех ее частей на основе использования новых информационных технологий. В частности, перемены касаются средств и технологий обучения [11].

Нелинейные технологии обучения - это совокупность системных методов моделирования и реализации деятельности преподавателя и деятельности ученика в личностно-ориентированной системе образования, базирующихся на принципах активного способа обучения в условиях информатизации и глобальной коммуникации.

Традиционно содержание предметной области формируют последовательно, по темам и разделам. Модели знаний по структуре представляют линейные отношения понятий и объектов (например, учебные программы курсов, оглавления учебников). Альтернативными нелинейными структурами являются семантические сети знаний.

Новые информационные технологии, в силу своей специфики, не могут в полной мере реализовать свой образовательный потенциал в традиционной образовательной системе, в которой доминируют дидактические линейные технологии передачи готовых знаний. В сложившейся школьной системе образования обучающая среда (учитель, средства и технологии обучения) нацелены на линейную модель знаний (последовательное, систематическое представление знаний предметной области) и линейную технологию обучения.

Эффективность обучения зависит от того, насколько педагогам удастся обеспечить единство своих действий с действиями обучаемых. При совпадении активности субъектов и объектов педагогического процесса его эффективность резко возрастает, наступает явление «педагогического резонанса».

Современная система образования должна обеспечить условия для развития у обучаемого умений и навыков ставить задачи, моделировать, оптимизировать, принимать решения в условиях неопределенности, учить умению добывать знания. В этой связи во многих случаях, особенно в прикладных предметных областях, целесообразно использовать нелинейные модели. Элементы обучения по подобному пути развиты в аналитических школах развивающего, проблемно-ориентированного направлений. Здесь обучение осуществляется по схеме: цель - задача - исследование - знание – контроль [17].

По всей видимости, удачное сочетание линейных и нелинейных технологий в обучении может существенно облегчить достижение целевых задач учебного процесса, в особенности при работе с одаренными детьми.

Учитель в первом случае выступает в роли проводника знаний. От него требуется владеть знаниями предметной области в соответствии с линейной структурой, реализовывать дидактические технологии передачи знаний, уметь разумно проводить декомпозицию заданного объема знаний на упорядоченную последовательность разделов, передать знания этих разделов, а затем систематизировать их.

При использовании нелинейных моделей роль учителя меняется. Теперь учитель выступает как организатор деятельности обучаемых, постановщик задач, системный интегратор [17].

Одна из задач педагогической науки - обобщение и систематизация знаний в определенных областях деятельности человека. Не менее важной представляется задача обучения ученика способам и технологиям добывания знаний, методам познания. Для решения данных задач представляются нелинейные формы организации обучения. Рассмотрим основные из них.

**Исследовательская деятельность.** Метод компьютерного моделирования в исследовательской деятельности является важным подходом в формировании системного мировоззрения обучающегося и выступает в роли технологии производства условий для деятельности,

овладения способами самоорганизации, и организации своей деятельности. Также обучающимся представлены различные методы и способы добывания и приобретения ЗУН [17].

При использовании таких технологий в целях образования значительно ускоряется развитие школ, ориентированных на исследовательские, развивающие формы и методы обучения школьника, а также на его познавательные способности, в частности.

**Учебно-проектная деятельность**, построенная на основе метода проектов. Метод учебных проектов приобретает актуальность в связи с его высокой эффективностью при дистанционном обучении на основе средств телекоммуникаций и информационных систем.

Метод проектов создается в рамках педагогической теории, которая сориентирована на связь школы с жизнью, с целью обучения, с одной стороны, а с другой - с целью изменения последней. Самостоятельная, умеющая мыслить и действовать независимо от воли других личность, воспитанная в школе, способна на активное участие в общественных, экономических преобразованиях. С этим и связано активное использование метода проектов в личностно-ориентированных системах образования.

**Рейтинговая деятельность**, на основе олимпиад, конкурсов, конференций. Рассматриваемые виды рейтинговых мероприятий, как правило, относят к внеучебной деятельности. Однако для личностно-ориентированной системы, для открытого и дистанционного обучения при использовании нелинейной технологии, они могут оказаться важными методическими средствами и методиками обучения в "урочной" классной системе.

Важно, чтобы все рейтинговые мероприятия в классе, школе носили массовый характер и были связаны с официальными мероприятиями, проводимыми органами образования района, города, региона, страны.

**Дистанционная деятельность**, на основе сетевых технологий, телеконференций.

*Параллельное обучение*, обозначающее технологию увеличения скорости обработки информации за счет распараллеливания процессов. Существует два подхода к организации параллельного обучения:

а) в первом подходе имеет значение специальное структурирование модели знаний предметной области так, чтобы отдельные темы делились на определенное количество разделов, которые являются одним классом эквивалентности. Задания по различным разделам одного класса эквивалентности распределяются между обучающимися для последующей публичной (в рамках группы) защиты их совместно с управляющими комментариями преподавателя. На семинаре при публичной защите своих проектов обучающиеся в целом получают обзорное представление по всем разделам темы;

б) во втором подходе предусмотрено создание рабочих мини-групп обучающихся в рамках организованной проектной деятельности при решении одной комплексной задачи. Деятельность подразделяется на независимые этапы, распределяемые мини-группами между собой. Группы работают параллельно со своей частью задачи, и в итоге все обучающиеся успешно справляются за довольно непродолжительное время с трудоемким и сложным методом. Сам алгоритм метода решения задачи усваивается всеми группами во время обмена промежуточными результатами, подведения итога и обсуждения найденного решения [11].

С помощью параллельного способа обучения значительно сокращается аудиторное время обучения (без урезания объема курса). Также имеется возможность использования данного способа в качестве оболочки индивидуальных схем обучения и включения его в качестве составного элемента в другие обучающие системы [21].

В настоящее время активно разрабатываются компьютерные инструментальные средства для организации нелинейного учебного процесса:

1) электронные учебники и самоучители;

- 2) демонстрационные модели и примеры;
- 3) компьютерные тесты (для контроля знаний обучающихся).

Главной особенностью нелинейных технологий выступает проблема постановки учебных целей и задач по предметной области. Они позволят педагогу формировать экспертный путь их достижения и решения, предложить необходимый инструментарий, методический материал, инструкции, опыт.

Для нелинейных технологий обучения важную роль играет контроль обучения, который чаще проходит в форме семинаров, конференций, публичных защит выполненных проектов.

Весьма полезны компьютерные тесты, которые отражают «гипертекстовую» контрольную экспертизу с использованием мультимедиа систем [17].

Обобщив вышеизложенное, можно подвести итог: применение нелинейных технологий обучения дает возможность осуществления принципа опережающего обучения, эффективного использования компьютеров в обучении, а также развития умений решать задачи с неполными исходными данными и плохо поставленные задачи, навыков принятия решений в условиях неопределенности, что является очень важным как для учебного процесса, так и для самих обучающихся.

## **1.2 Сущность образовательного кластера (на примере платформы мегакласс)**

Развитие технологий обработки информации и вычислительной техники определяет приход кластерных систем и многоядерных процессоров, распространение параллельных процессов обработки информации, параллельных вычислений, распределенных принципов сетевого взаимодействия. Но содержание курсов информатики не отражает эти изменения [3, 5].

Особый интерес вызывает моделирование таких интеграционных методических систем и структур предметного обучения молодежи, с помощью которых мог бы быть организован единый учебный процесс в связке «педвуз-школа».

Данные цели можно реализовать с помощью образовательного кластера.

Образовательный кластер – это совокупность учреждений образования, связанных между собой прежде всего партнерскими отношениями с предприятиями по отраслевому признаку. Также представляется возможным интеграция образования, науки и жизни в образовательном кластере, реализация непрерывной практико-ориентированной подготовки обучающихся, организация рабочей среды между школами, вузом и производством в условиях ДОТ и ЭО. Для организации планируемой работы необходимо выделить грани образовательной платформы, в которой возможно взаимодействие существующих сфер образования, бизнеса и науки без их «перестройки». Подобная платформа может быть построена на базе идей и принципов проекта «Мега-класс» [4].

Проведение исследований, а также приобретение необходимого опыта стало возможным при создании нескольких образовательных кластеров (Красноярский край). Структура образовательной платформы на текущий момент включает:

- 1) IT-кластер;
- 2) инженерно-технический кластер;
- 3) северо-арктический кластер.

Изучим основное содержание и задачи каждого кластера в составе образовательной технологической платформы.

Инженерно-технический кластер представляет собой открытую платформу с базовыми организациями: СибГАУ им. академика М.Ф. Решетнева, СФУ, КазНПУ имени Абая (Алматы, Казахстан), КГПУ им В.П. Астафьева, школы с классами инженерно-технологического профиля (МАОУ

гимназия 9 г. Красноярска, МОУ лицей №1 г. Ачинска, школа г. Алматы), МБОУ лицей 28 г. Красноярска. Штаб кластера – МАОУ гимназия 9 г. Красноярска. Работа кластера основана на формировании у каждого обучающегося УУД при коллективной, распределенной между собой работе, развитии инженерного мышления школьников. Предметная основа работы кластера: математика и физика, информатика, робототехника.

IT-кластер представляет собой также открытую платформу, в которую входят базовые учреждения: СибГАУ им. академика М.Ф. Решетнева, СФУ, КГПУ им В.П. Астафьева, МБОУ гимназия 14 г. Красноярска, Открытая платформа с базовыми организациями: IT-фирмы – ООО «Системы Промышленной Автоматизации», «Открытые технологии», МАОУ школа 17 г. Ачинска, МАОУ школа 3 г. Ачинска. Штаб кластера – КГПУ им В.П. Астафьева. Деятельность кластера нацелена на глобализацию и массовизацию курса информатики и ИКТ в практико-ориентированную направленность: решение практико-ориентированных задач, интеграция с образовательным процессом педагогического вуза.

Северо-арктический кластер. Открытая платформа с базовыми организациями: СФУ, КБ «Искра», КГПУ им В.П. Астафьева, 2 школы п. Северо-Енисейска, IT-фирмы - СПА, школы северных и арктических территорий, включая МБОУ СОШ №8 г. Лесосибирска, школы пос. Носок, Нерюнгри. Штаб кластера – КГПУ им В.П. Астафьева. Основной целью данного кластера является устранение дефицита кадров в школах северо-арктических территорий.

Мега-класс представляет собой методическую систему учебно-воспитательной деятельности разных школ в информационно-образовательной среде облачных сервисов, основанной на интеграции учебно-воспитательного и научного процессов педвуза, образовательных организаций, муниципальных управлений образования с применением ЭО и ДОТ [20].

Суть данной платформы состоит в том, что обучение проводится одновременно в формате мегаурока по конкретной дисциплине в онлайн режиме, при формировании образовательного кластера «школа-педвуз», объединяющем повышение квалификации педагогов в рабочей среде с помощью облачных технологий, обучение студентов и школьников в совместный процесс. Для создания и проведения мега-урока создаются методические пособия (сценарий взаимодействия всех участников урока, облачные сервисы), инструктаж для каждого преподавателя, учителя, студента [6].

Преимуществом технологии мега-класс является кооперация школьного и педагогического образования, слияние вузовской науки и бизнеса без дополнительных материально-финансовых затрат, лишь за счет ресурсов и регламентов участников кластера [18].

Данная образовательная платформа базируется на научно-исследовательской деятельности сообщества учителей, преподавателей и студентов по систематизации и представлению знаний и информации окружающего мира и предоставляет новые результаты обучения, а именно:

- формирование компетенций студента как будущего учителя;
- формирование алгоритмического мышления школьника;
- повышение квалификации учителей и преподавателей вуза [18].

Реализация модели обучения по технологии Мега-класс ведет к достижению следующих результатов:

1. У преподавателей, ученых и профессуры вуза: взаимосвязь педагогической науки со школьной практикой, жизнью.
2. У работников IT-фирм: привлечение интеллектуальных и трудовых резервов для развития своей индустрии.
3. У учителя: стабильное повышение профессиональной квалификации при совместной деятельности в кластере, находясь на рабочем месте, со специалистами IT-фирм, учеными и профессурой педвуза.

4. У студента педвуза: возрастание познавательных интересов к деятельности за счет участия в проведении мега-уроков, что является хорошей мотивацией обучения предметам и педагогической практики, а также получение новых навыков профессиональной работы.

5. У школьника: значительное повышение познавательной активности в обучении за счет смены образовательного процесса, а также возможность обеспечения качественного образования, развития когнитивных навыков и УУД.

Также модель мегаурока решает ряд проблем методики преподавания информатики, а именно:

1. Недостаточно разработанные методы контроля знаний школьников.
2. Отсутствующее мультимедийное сопровождение в курсе информатики.
3. У большинства авторов различное содержание программ по курсу информатики.
4. Ограниченное количество часов в учебном плане.

Как уже было описано выше, мегаурок проводится одновременно во всех школах кластера с участием преподавателей и студентов, помогающих в проведении урока согласно предложенному плану. Облако представляет порталы обучающих средств (ментальные учебники, видеолекции и пр.), УУД учащихся, диагностик качества обучения, разработок учеников, студентов и преподавателей по актуальным проблемам науки и общества.

#### ***Обобщенная модель урока.***

1. Вводное занятие - погружение в проект в форме видеоконференции.
2. Теоретическая часть в форме: описания проблемы, видеолекции, презентации на платформе Web-сервисов и др.
3. Совместная деятельность педагога и учащихся. Разделение учащихся на группы из разных школ, студенты университета являются тьюторами, организовывающими активную работу с вопросами модератора и одновременную оценку на предмет качества выполнения задания (информационно - учебный кейс, включающий карту ответа).

4. Представление результата работы в виде вывески на «стену ответов».
5. Комментарии ответов. Вариантов несколько:
  - а) коллективное обсуждение учащихся;
  - б) комментирует выборочно модератор;
  - в) комментируют студенты;
6. Подведение итогов в виде обобщения, идут обсуждения и комментарии на слайде.
7. Обозначение опережающей проблемы на следующее занятие.
8. Заполнение мониторинговых или рейтинговых форм для отслеживания достижений учащихся в виде оценки учителем, самооценки, взаимооценки (на выходе или на каждом этапе урока). Возможность занесения результата в тестовой оболочке [4].

***Модель: Решение практико-ориентированных задач.***

1. Создание модератором ситуации для решения практико-ориентированных задач (наличие банка практико-ориентированных задач по блокам).
2. Создание учителем ситуации выбора:
  - а) одна задача на класс;
  - б) одна задача на группу по принципу: «Выбрал, неси ответственность»;
3. Выбор дает для учителя информацию о личностных результатах.
4. Открытость обсуждения решения задачи путем выставления на форуме портала. В обсуждении принимают участие студенты, возможны комментарии к оценкам, свободное обсуждение.

Данная модель проведения урока предусматривает одни и те же требования к составлению задач, разработку шаблона решений и систему критериев оценки. За каждой из групп закрепляется студент-тьютор и организуется взаимодействие групп в форме сетевого общения.

***Модель: Решение типовых задач***

1. Организация видеотрансляции занятия модератора с группой студентов по определенной теме.

2. Просмотр данной видеотрансляции учениками с возможностью включения в процесс.

3. Предварительно учитель мотивирует учащихся на включение в процесс решения.

Основной задачей такого метода является обучение нахождению (прежде всего в памяти) соответствующего конкретной задаче шаблона и наполнять выбранный шаблон контекстом, в нашем случае – способом решения конкретной задачи.

В области информатики большое внимание уделяется программированию, как одному из ключевых разделов, образующих предметное содержание профильного курса школьной информатики. В дальнейшей работе все внимание сконцентрировано на процессе обучения программированию в связи со сложностью и объемностью изучения данного раздела и с его большим развивающимся потенциалом [31].

### **1.3. Особенности обучения программированию старшеклассников инженерно-технического профиля в условиях образовательного кластера**

#### **1.3.1. Анализ требований ФГОС**

Новое время предъявляет к школе требования, которые выражены в Федеральном государственном образовательном стандарте (ФГОС) среднего (полного) общего образования, утвержденные Министерством образования и науки Российской Федерации от 17 мая 2012 г, № 413 (ред. От 29.06.2017) [2].

Федеральные государственные образовательные стандарты (ФГОС) - совокупность требований, обязательных при реализации основных образовательных программ начального общего, основного общего, среднего (полного) общего, начального профессионального, среднего профессионального и высшего профессионального образования образовательными учреждениями, имеющими государственную аккредитацию, которые должны осуществлять единообразие

образовательного пространства РФ, а также связь основных образовательных программ начального общего, основного общего, среднего (полного) общего, начального профессионального, среднего профессионального и высшего профессионального образования.

Изучение предметной области «Математика и информатика» связанные с областью «Информатика» по ФГОС должно обеспечить осознание значения математики и информатики в повседневной жизни человека, понимание роли информационных процессов в современном мире.

В результате изучения предметной области «Математика и информатика» обучающиеся развивают логическое и математическое мышление, получают представление о математических моделях; овладевают умениями решения учебных задач; получают представление об основных информационных процессах в реальных ситуациях.

Предметные результаты изучения предметной области «Математика и информатика», связанные с изучением области «Информатика» (углубленный уровень) должны включать требования к результатам освоения базового курса и дополнительно отражать:

1) овладение универсальным языком программирования высокого уровня (по выбору), понятиями о базовых структурах и типах данных; умением использовать основные управляющие конструкции;

2) владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;

3) сформированность представлений о компьютерных сетях и их роли в современном мире, знаний базовых принципов организации и функционирования компьютерных сетей, об устройстве современных компьютеров, о тенденциях развития компьютерных технологий, об общих принципах разработки и функционирования интернет-приложений, владение

основными сведениями о базах данных, их структуре, средствах создания и работы с ним;

4) сформированность представлений о важнейших видах дискретных объектов и об их простейших свойствах, о кодировании и декодировании данных и причинах искажения данных при передаче; систематизацию знаний, относящихся к математическим объектам информатики; умение строить математические объекты информатики, в том числе логические формулы;

5) владение опытом построения и использования компьютерно-математических моделей, проведения экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов, пользоваться базами данных и справочными системами.

Владение универсальным языком программирования, а также развитие алгоритмического мышления - неотъемлемая часть целевых предметных установок нового стандарта. Но формирование алгоритмической культуры и освоение языков программирования также присутствует в других пунктах данного стандарта [1].

В современном российском образовании школьнику необходимо формировать УУД, помимо знаний, умений и навыков. Наличие в совокупности УУД и ЗУН способствует применению обучающимися данных знаний и навыков при решении практико-ориентированных задач.

Особенно, данные УУД должны формироваться у ученика по отношению к продуктам информационной деятельности (как созданным им самим, так и другими людьми), к способам обмена этими продуктами, их хранения, а также по отношению к техническим и программным средствам информационной деятельности. Разработка алгоритмов и написание программ являются полезными средствами при достижении новых результатов обучения [13].

Методологической основой нового стандарта является системно-

деятельностный подход к обучению, который обеспечивает:

- 1) формирование готовности к саморазвитию и непрерывному образованию, активной учебно-познавательной деятельности школьников;
- 2) построение процесса обучения с учётом индивидуальных возрастных, психологических особенностей школьников;
- 3) проектирование и создание социальной среды развития школьников в системе образования.

Данный подход основан на теоретических положениях концепции Л.С.Выготского, А.Н.Леонтьева, Д.Б.Эльконина, П.Я.Гальперина, раскрывающих основные психологические закономерности процесса обучения и воспитания, структуру образовательной деятельности учащихся с учетом общих закономерностей развития детей, и представляет собой единство системного и деятельностного подходов. Системный подход в педагогике является общенаучным и основан на идее рассмотрения объекта познания как системы.

Деятельностный подход исходит из положения о том, что психологические способности человека есть результат преобразования внешней предметной деятельности во внутреннюю психическую деятельность путем последовательных преобразований, т.е. развитие учащихся определяется характером организации их деятельности, в первую очередь, учебной [16].

Так как роль программирования значительно возрастает, как и требования обеспечения «...активной учебно-познавательной деятельности обучающихся; формирования готовности обучающихся к саморазвитию и непрерывному образованию», при углубленном изучении информатики в старших классах становится актуальной проблема разработки методики обучения программированию в школе.

Программирование, вне всяких сомнений, является таким видом деятельности, с помощью которого можно проектировать действия,

совершенствовать и развивать мышление, определять последовательности и закономерности.

Проанализировав вышеизложенное, в обучении раздела программирования «Основы алгоритмических конструкций» в школьной информатике можно выделить три целевых аспекта:

Общеобразовательный аспект, который связан с формированием алгоритмической культуры учащихся, способности планировать пути достижения цели, в том числе альтернативные, а также умения соотносить свои действия с планируемыми результатами, осуществление контроля своей деятельности в процессе достижения результата. Данный аспект связан с подготовкой школьников к практической деятельности в условиях широкого использования информационных технологий.

Программистский аспект. Связывается с фундаментальными идеями информатики: формализованное описание алгоритма в условиях жестких ограничений используемого языка программирования и среды программного исполнителя, как условий автоматизации процесса исполнения алгоритма (фундаментальные идеи, практическая деятельность по решению задач с использованием современных технологий программирования).

Развивающий аспект. Предполагает развитие алгоритмического (операционного) мышления учащихся: умение выстраивать план действий, умение предвосхищать результат, умение формально исполнять последовательность действий (принимать условия обстановки, в которой происходит деятельность), понимать последовательные, параллельные, недетерминированные действия и умение анализировать результат. Данные целевые компоненты обязаны отразиться в формировании целей изучения раздела программирования «Основы алгоритмических конструкций» в курсе информатики старшей школы [2,16,26].

**1.3.2. Особенности изучения раздела «Основы алгоритмических конструкций» в старшей школе.**

Обучение информатике школьников имеет огромное значение для реализации потенциала общего среднего образования и переживает существенные изменения при введении ФГОС общего образования второго поколения.

На данный момент в курсе информатики нужно осваивать не только отдельные компоненты прикладной направленности, но также и развивать представления об информатике как дисциплине естественнонаучной. Изучение информатики в настоящее время направлено не только на информационные технологии и процессы, но и базисные методы и закономерности реализации этих процессов в обществе и окружающей среде. И в связи с этим можно сказать, что информатика является самостоятельной научной отраслью. Также информатика имеет и междисциплинарный характер, который является первоосновой для последующего формирования и развития естествознания, социальных и гуманитарных наук.

Основной проблемой в настоящее время является неполная разработка методик преподавания информатики, особенно это касается программирования вообще и раздела «Основы алгоритмических конструкций» в частности, который является неотъемлемой частью школьного курса информатики и основой автоматизации информационных процессов, а также играет важную роль в формировании алгоритмического, наглядно-образного, логического мышления учеников. Развитие данных типов мышления необходимо для дальнейшей профессиональной деятельности обучающихся в обществе [15].

Обучение программированию с методической точки зрения является одной из самых трудных задач. Составление программ – это сложный процесс, в который входит огромное количество различных этапов. Особую сложность вызывает постановка задачи и ее алгоритмизация. Как правило, в учебниках и учебных пособиях при объяснении данных разделов приводятся конкретные задачи различных типов и соответствующие им алгоритмы с некоторыми частными пояснениями. В основном рассматриваются примеры

трех типов: линейного, разветвляющегося, циклического, а также алгоритмы с «вложенными» циклами. Если дается описание сложных задач, то приводятся готовые алгоритмы решения без описания процесса построения алгоритма.

Далее проведем анализ основных школьных учебников по информатике для 10-11 классов.

1) УМК «Информатика и ИКТ» 10-11 класс М. Е. Фиошин. Базовый и профильный уровень. Базовый уровень: изучение языка программирования Pascal. Вначале дается изучение темы «Алгоритмизация», а затем идет переход к программированию. Автор предлагает типовую последовательность разработки программы, включающую постановку задачи, разработку алгоритма, запись программы на языке программирования, ее запуск и отладку. Далее происходит закрепление знаний об основных алгоритмических конструкциях. Практические занятия осуществляются с помощью составления алгоритмов при внедрении блок-схем.

Основным целями в данном учебнике являются: формирование и развитие системно-деятельностного подхода к изучению информатики, а также алгоритмического мышления, способности к построению алгоритмических моделей с последующей реализацией с помощью программных средств. К учебнику прилагается компакт-диск, который содержит тексты, упражнения, видеоуроки и справочный материал. В практикуме нет задач на разработку программы, содержащей подпрограмму и алгоритма (программы), требующего для решения поставленной задачи использования логических операций [29];

2) УМК «Информатика» 10-11 класс, Л.З. Шауцукова. Раздел «Алгоритмизация» направлен на изучение алгоритмов, базовых алгоритмических конструкций, алгоритмических языков и исполнителей.

В практикуме, в отличие от теоретической части учебника, встречаются профессиональные языки. Также в практикуме приводится большое количество примеров задач с решениями на школьном

алгоритмическом языке, а также на языке TurboPascal и QBasic. Ученикам предоставлено изучения языка программирования с помощью примеров в виде кодов программ с соответствующими подробными комментариями.

Практические занятия основаны на составлении простых программ, т.к. базовый курс не предполагает более углубленного изучения языка. Практикум по алгоритмизации и программированию ориентирован на учащихся, имеющих начальное представление об орфографии языков TurboPascal, предназначен для обучения основам программирования и развития навыков алгоритмического мышления.

Учебник частично подходит для профильного уровня изучения программирования, т.к. в нем нет понятийного аппарата об объектах, событиях и объектно-ориентированном программировании. Также в нем не изучаются следующие темы: «Вычислимые функции» и «Доказательство правильности алгоритма» [33];

3) УМК «Информатика и информационные технологии», 10 – 11 классы. Н.Д.Угринович. Учебник ориентирован на преподавание в общеобразовательных учреждениях профильного курса на базовом уровне в 10 классе. Основные изучаемые языки программирования: VisualBasic и VBA. Особое внимание уделяется объектно-ориентированному программированию. Происходит изучение следующих тем: «Классы объектов», «Экземпляры класса», «Семейства объектов», «Иерархия объектов в VBA», «Интегрированная среда разработки языка VBA», «Кодирование алгоритмов в форме макросов». Интерфейс среды MicrosoftVisualBasic также не остается без внимания: в ней подробно изучается каждое окно интерфейса. В теоретической части нет понятий об этапах разработки и отладки программ, нет учебных языков программирования, изучение начинается с VisualBasic. Может использоваться для углубленного изучения программирования с учетом начальных знаний учеников в данной теме [8];

4) УМК «Информатика» 10-11 классы, К.Ю.Поляков.

Преимуществом данного учебника является возможность выбора учителем профессионального языка программирования из нескольких предложенных. К примеру: в основной версии учебника приведен язык программирования Pascal, но на официальном сайте К. Ю. Полякова в свободном доступе выложены варианты глав «Алгоритмизация и программирование» и «Объектно-ориентированное программирование» с использованием языков Pascal, C++ и Python.

В 10 классе изучение программирования начинается с параграфа «Алгоритм и его свойства», где рассматриваются свойства и способы записи алгоритмов, а также представлены определения понятий алгоритма и исполнителя. С параграфа «Простейшие программы» начинается изучение языка программирования в котором представлена структура программы, операторы ввода вывода, понятие переменной. Далее идет изучение основных алгоритмических конструкций, массивов и работа с файлами.

Базовый уровень не предусматривает изучение этапов разработки и приемов отладки программ. В профильном уровне отсутствуют вычисляемые функции и доказательства правильности. Учебник подходит для изучения программирования, как на базовом, так и на профильном уровнях [23].

Таким образом, проведя анализ каждого из учебников, а также документа ФГОС среднего (полного) общего образования, можно сделать вывод о том, что содержание учебного материала по программированию в старшей школе соответствует требованиям ФГОС, а значит подходит для дальнейшей исследовательской деятельности.

Но в рабочих программах учебников по информатике в 10-11 классах существенно не хватает таких критериев, как направленность на взаимодействие обучающихся друг с другом, групповая форма работы, формирование более активной учебно-познавательной деятельности обучающихся; готовность обучающихся к саморазвитию и непрерывному образованию. Скомпенсировать недостаток данных требований возможно

при включении в образовательный процесс платформы «Мега-класс» в совокупности с нелинейными технологиями, в условиях которой учащимся представится возможность групповой работы как в своих классах, так и работы в межсетевых группах, что будет способствовать активной учебно-познавательной деятельности и готовности к непрерывному образованию [2,14,19].

Таким образом, обучение программированию старшеклассников инженерно-технического профиля в условиях образовательного кластера включает дополнительные цели:

1) развитие взаимодействия обучающихся друг с другом (на основе межсетевого общения), что положительно повлияет на повышение навыков коммуникации каждого обучающегося;

2) активное использование группового способа обучения, что влечет за собой и развитие творческой самореализации обучающихся, их познавательных способностей;

3) формирование более активной учебно-познавательной деятельности обучающихся и готовности к саморазвитию и непрерывному образованию, что также положительно повлияет на повышение познавательных интересов, универсальных учебных действий учащихся.

При включении образовательного кластера с использованием нелинейных технологий в процесс обучения программированию, меняется построение содержания учебных занятий:

а) применение нелинейных технологий. Если традиционная форма проведения уроков полностью соответствует содержанию того или иного учебника, идет последовательное, линейное изложение материала, то при нелинейном подходе содержание параграфов, разделов или глав может быть использовано в качестве базовой основы планирования уроков, но сама конструкция проведения уроков меняется. Имеется огромное количество вариантов построения учебного процесса: с объединением нескольких уроков (при этом существенно сокращается объем времени изучения материала) или

же разбиения одного урока на несколько модулей для более качественного усвоения материала, изучение более обширных тем с последующим включением тем, содержащих конкретное, детальное изучение того или иного параграфа;

б) включение образовательного кластера – технологии мегакласс – также меняет построение содержания учебных занятий: урок проводится не в традиционной форме, а с использованием дополнительных электронных средств обучения, различных межсетевых форм работы обучающихся.

Введение образовательной платформы мегакласс с использованием нелинейных подходов влечет за собой изменения форм организации деятельности обучающихся. В отличие от традиционных форм обучения, технология мегакласс основана прежде всего на межсетевом взаимодействии обучающихся, преподавателей, тьюторов. Особое внимание уделяется групповым формам работы, фронтальная работа и форма лекционного изложения материала учителем сводится к минимуму или исключается совсем. Изучение информации ведется преимущественно в форме исследовательской деятельности.

Также в организацию работы учащихся привлекаются дополнительные средства обучения: коммуникация межсетевых групп осуществляется в облачных сервисах (например, в интерактивной доске Linoit, виртуальной сети Edmodo, облачном сервисе google-документы), совместное написание программ осуществляется в онлайн-компиляторах или объектно-ориентированных средах (например, Alice), и контролируется через системы контроля версий (к примеру, веб-сервис GitHub).

Контроль за деятельностью обучающихся также кардинально меняется. Для контроля каждой группы подключают различных тьюторов-экспертов. Взаимодействуя со своими группами через облачные сервисы, тьюторы контролируют действия и результаты деятельности учащихся, помогают при возникновении различных вопросов. Контроль всего образовательного процесса в мегаклассе осуществляется мега-учителем.

## **Выводы по главе 1**

Проанализировав нормативные документы, можно сделать вывод, что для достижения образовательных результатов, которые отвечают новым требованиям к результатам обучения программированию в старшей школе, нужно разрабатывать специальные (отличные от традиционных) уроки и задания, которые позволяют обучающемуся подготовиться к дальнейшему образованию в высшей школе как в качественном освоении содержания предмета, так и в формировании и развитии умственных способностей.

При нарастающих противоречиях динамичного научно-технологического прогресса и классно-урочной системы обучения, обучение информатике, в частности, программированию, рационально осуществлять в исследовательской, практико-ориентированной среде – образовательных кластерах. Мегакласс является одним из самых востребованных образовательных кластеров, включение в образовательный процесс которого приведет к значительному повышению познавательной активности обучающихся за счет смены образовательного процесса, а также к обеспечению качественного образования, развитию когнитивных навыков и УУД.

Использование нелинейных технологий в обучении программированию дает возможность направить учебный процесс в практико-ориентированную, научно-исследовательскую деятельность обучающихся, что благоприятно влияет на создание и развитие познавательных потребностей. Также методы и средства нелинейных технологий обучения способствуют значительному повышению мотивации обучающихся к обучению, развитию творческих способностей, самостоятельности и самоорганизованности.

## **Глава 2. Система мега-уроков для 10 класса (на примере раздела «Основы алгоритмических конструкций»)**

### **2.1. Цели и задачи обучения школьников программированию в образовательном кластере**

Разработка методики обучения программированию, направленной на развитие самостоятельности, критического мышления, творческой активности является сложным процессом сопоставления синтаксиса и смыслового значения языков программирования, основ разработки программ, специализированных алгоритмов с методами их усвоения.

Уроки программирования направлены в первую очередь на работу с информацией, ее структуризации и управления ею. Овладение данными навыками обучающимися очень важно в современных нарастающих инновационных технологиях [20, 25].

Для успешного обучения программированию, нужны дополнительные методы обучения, подходы упрощения дополнительной мотивации учащихся. Одним из средств упрощения подачи материала является его распараллеливание (метод «параллельного обучения»). Данный метод позволит существенно сократить аудиторное время обучения, не урезая при этом объем курса. Также его можно использовать в качестве оболочки для индивидуальных схем обучения. Его также можно включать в другие обучающие системы в качестве составного элемента.

Обучение программированию в старших классах начинается с раздела «Основы алгоритмических конструкций».

В условиях образовательного кластера и усовершенствования учебного процесса целью обучения программированию школьников (на примере раздела «Основы алгоритмических конструкций») является изучение базовых алгоритмических конструкций в курсе информатики на профильном уровне с использованием нелинейных технологий.

Следовательно, для достижения поставленной цели необходимо решить

соответствующие задачи обучения программированию, а именно:

1. Научиться выдвигать и обосновывать идею решения задачи (освоить методы программирования), а также структурировать эту идею (модель данных, технология проектирования программы, структурный подход, пошаговая детализация и др.).

2. Освоить основные конструкции языка программирования Python:

- Знать основные типы данных, логических переменных, основные арифметические операции и выражения, уметь распознавать выражения и арифметические операции при анализе условия задачи, а также применять их при написании простейших программ;

- знать способы организации ветвлений, распознавать ветвления в процессе анализа условия задачи и построения предварительного алгоритма решения, применять условные конструкции в процессе написания программы, уметь писать составные конструкции, в которых есть вложенные ветвления;

- иметь представления о способах организации циклов с условием и циклов с переменной, распознавать циклы при анализе условий задач и построения блок-схемы (алгоритма) решения, научиться составлять конструкции, имеющие вложенные циклы;

- иметь представления об основных процедурах и функциях в Python, уметь распознавать функции при анализе условий задач и построения предварительного алгоритма решения, уметь писать процедуры с параметрами, а также писать конструкции с логическими функциями;

- знать способы организации рекурсивных функций, приемы построения рекурсий, уметь распознавать составные части рекурсии при решении конкретной задачи, а также составлять алгоритм решения такой задачи, уметь применять рекурсию при написании программ;

3. Научиться формализовать элементы полученной структуры средствами выбранного языка (правила представления данных, основные операторы и синтаксис языка, правила записи программы), анализировать

результат решения задачи при разных значениях исходных данных, сравнивать различные методы решения задачи (по эффективности, простоте, сложности алгоритма и т.п.

4. овладеть навыком составления тестов для проверки работоспособности программ, научиться читать чужой программный код и видеть ошибки в чужом коде;

5. Использовать приобретенные знания и умения в практической деятельности и повседневной жизни для поиска и отбора информации, в частности связанной с личными познавательными интересами, самообразованием [26].

При решении данных задач обучающимися будут достигнуты следующие результаты обучения программированию по разделу «Основы алгоритмических конструкций»:

1) овладение умениями строить программы на формальном языке с использованием базовых алгоритмических конструкций, удовлетворяющие заданному описанию, а также создавать программы на языке программирования по данному описанию:

- знания основных типов данных, логических переменных, основных арифметических операций и выражений, способов организации ветвлений, циклов с условием и циклов с переменной, основных процедур и функций в Python, приемы построения рекурсий;

- умения распознавать выражения и арифметические операции, ветвления, циклы, функции и рекурсию при анализе условия задачи и построения предварительного алгоритма решения, а также применять данные алгоритмические конструкции при написании программ;

2) умения использовать общепользовательские инструменты и настраивать их для нужд пользователя, а также развитие алгоритмического мышления, способностей к формализации, элементов системного мышления;

3) умение составлять тесты для проверки работоспособности программ, умение читать чужой программный код, умение видеть ошибки в чужом коде;

4) воспитание умения планировать, работать в коллективе, чувства ответственности за результаты своего труда, используемые другими людьми, установки на позитивную социальную деятельность в информационном обществе, недопустимости действий, нарушающих правовые и этические нормы работы с информацией.

## **2.2. Содержание и средства обучения программированию на мега-уроках в нелинейных технологиях**

Разработка системы мега-уроков по программированию на языке Python была основана на учебнике К.Ю.Полякова и Е.А.Еремина «Информатика. Углублённый уровень» для 10-11 классов. Углубленное изучение программирования на Python в данном учебнике начинается с главы 8 «Алгоритмизация и программирование». Был выбран первый раздел этой главы «Основы алгоритмических конструкций», который включает в себя 8 параграфов. На основе этого раздела было решено разработать систему мега-уроков с использованием нелинейного подхода в соответствии с представленными параграфами [23].

Составление системы мега-уроков по разделу «Основы алгоритмических конструкций» производилось по следующему плану (см. Таблицу 1):

*Таблица 1. Тематическое (поурочное) планирование*

<b>№ урока</b>	<b>Тема</b>	<b>Система контроля</b>	<b>Кол-во часов</b>
1	Введение в программирование. Алгоритм и его свойства	Разноуровневые задания по изучению программного обеспечения компьютера, тест по теме «Алгоритм и его свойства» (Приложение А)	1
2	Простейшие программы.	Задания на изучение темы	1

	Вычисления	«Простейшие программы. Вычисления» (Приложение Б)	
3	Ветвления	Задания на изучение темы «Ветвления», задача на закрепление (Приложение В)	1
4	Циклы	Задания на изучение темы «Циклы», задача на закрепление (Приложение Г)	1
5	Вложенные циклы	Задания на изучение темы «Вложенные циклы», задача на закрепление (Приложение Д)	1
6	Подпрограммы в Python	Задания на изучение темы «Подпрограммы в Python» (Приложение Е)	1
7	Рекурсия	Задания на изучение материала по теме «Рекурсия» (Приложение Ж)	1
Итого:			7

Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента, который контролирует выполнение заданий обучающихся и помогает по мере поступления вопросов. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. При самостоятельном изучении материала каждая группа может использовать имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. Организатором всего учебного процесса является мега-учитель, а также он подводит итоги каждого занятия. Следующей особенностью является выступление педагога из вуза (модератора) в конце каждого мега-урока, в котором в необычной занимательной форме анализируются информационные процессы, происходящие в компьютере при решении конкретных информационных задач.

Для поддержки соревновательного элемента к каждому уроку подготавливается электронная таблица, которая предоставляется ученикам в режиме просмотра в начале урока. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп. Данный

рейтинг позволяет учащимся на протяжении всего урока отслеживать успехи своей группы (см. Приложения). Ниже представлен краткий обзор содержания каждого мега-урока:

1) первый урок является вводным. Для того, чтобы перейти к разделу «Основы алгоритмических конструкций», необходимо воспроизвести и систематизировать знания по ранее изученному материалу раздела «Программное обеспечение компьютера». Для этого в начале урока каждой группе обучающихся в печатном варианте представлены задания на повторение знаний о программном обеспечении компьютера. По окончании выполнения заданий подводятся итоги проведенной работы, обсуждение заданий, вызвавших затруднения. Далее мега-учителем озвучивается новая тема «Алгоритм и его свойства», где обучающиеся сами изучают материал по предложенной теме и выполняют тест на гугл-диске (см. Приложение А). При подведении итогов модератор анализирует взаимодействие каждой группы и объясняет вопросы теста, вызвавшие затруднение у большинства групп.

Данный урок проходит в традиционной (линейной) форме. Нелинейным средством обучения, внедренным в учебный процесс мега-урока, является канонический тест, состоящий из 4 вариантов ответа, один из которых верный.

Учебные материалы, используемые на уроке, представлены на гугл-диске и состоят из шести документов в форматах docx и pptx, а именно:

1. «Прикладное программное обеспечение», в котором описаны составляющие прикладного программного обеспечения, а также предоставлены примеры программ в табличной форме.
2. «Программное обеспечение», в котором содержится определение понятия программного обеспечения и приведена общая схема его состава.
3. «Системное программное обеспечение», содержание которого представлено в подробном описании составляющих и примеров программ системного программного обеспечения.

4. «Системы программирования», в содержание которого входит описание состава инструментальных программных обеспечений, а также примеры их программ.

5. «Алгоритм и его свойства» - презентация, в которой описаны основные свойства алгоритма, способы его записи, приведены примеры записи алгоритма.

6. «Ссылки (дополнительная информация по алгоритму)», в котором предоставлены ссылки на дополнительную, более обширную информацию об алгоритме.

Организация взаимодействия межсетевых групп осуществляется на интерактивной доске Linoit, а также в окошке для беседы в гугл-документе.

Данное занятие в формате мега-урока с использованием нелинейных средств обучения позволит обучающимся за короткий промежуток времени (1 урок) изучить и закрепить знания по теме «Алгоритм и его свойства», а также систематизировать полученные ранее знания о программном обеспечении компьютера;

2) второе занятие предусматривает изучение сразу двух тем «Простейшие программы» и «Вычисления». Работа групп осуществляется на основе нелинейной технологии «параллельное изучение». Деятельность подразделяется на независимые этапы, распределяемые мини-группами между собой. Группы работают параллельно со своей частью задачи. Далее представлен фрагмент мега-урока (подробнее см. Приложение Б).

#### ***Фрагмент мега-урока «Простейшие программы. Вычисления»:***

Мега-учитель с помощью презентации на гугл-диске объясняет учащимся новый материал по теме «Вычисления». Затем учащимся предстоит самостоятельное, более детальное изучение материала: группе №1 предлагается подробное изучение пункта «Арифметические операции и выражения», группе №2 – «Стандартные функции», группе №3 – «Случайные числа». Использовать для этого ученики могут имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и

ресурсы Интернет. После изучения материала каждая группа приступает к выполнению второй части заданий – составлению программ, для закрепления и систематизации полученных знаний. Тьютор следит за временем выполнения задания, проверяет готовую программу и выставляет баллы в рейтинговую таблицу.

По завершении второй части заданий группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного пункта по теме «Простейшие программы и вычисления» (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала).

Ниже представлена схема взаимодействия межсетевых групп (см. рис.1).

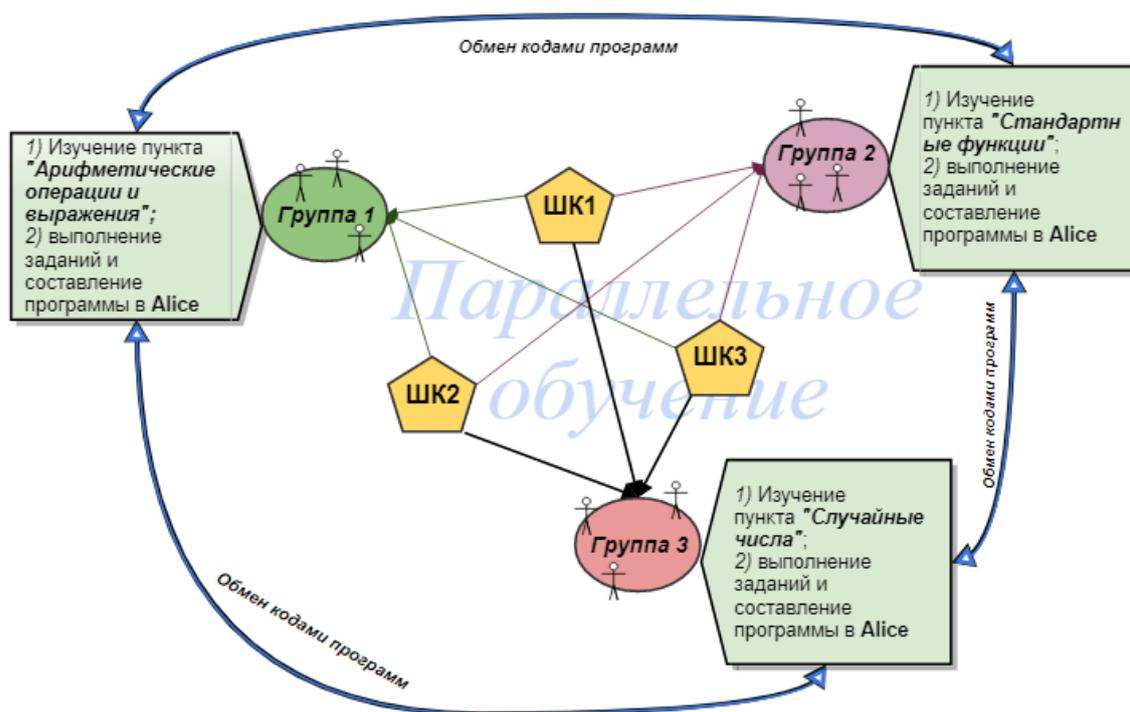


Рис.1. Схема деятельности межсетевых групп

Учебные материалы, используемые на уроке, состоят из семи документов в форматах docx и pptx, а именно:

1) 2 документа в формате pptx: «Вычисления», в котором имеется краткая теория об основных вычислениях в Питоне, а также «Простейшие

программы», где представлено описание и примеры простейших программ на Питон;

2) «ввод и вывод» - даны дополнительные ссылки на углубленное изучение материала;

3) «арифметические выражения и операции» - представлено описание основных арифметических операций в Питон, а также дополнительные ссылки на подробное изучение подтемы;

4) «случайные числа», где дано подробное описание модуля random и примеры программ, а также имеются дополнительные ссылки по данной подтеме;

5) «стандартные функции» - имеется описание основных стандартных функций в Питон и дополнительные ссылки;

6) «рейтинг» - рейтинговая таблица с баллами для всех групп.

Организация взаимодействия межсетевых групп и тьютора осуществляется в окошке для беседы в гугл-документе.

Данная форма проведения урока с использованием метода «параллельного обучения» позволит обучающимся за непродолжительное время полноценно усвоить сразу две темы раздела «Основные алгоритмические конструкции» во время обмена промежуточными результатами при использовании гугл-документов в межсетевом взаимодействии, подведения итога и обсуждения найденного решения;

3) третий мега-урок посвящен изучению следующей темы раздела: «Ветвления». Учащиеся также работают в межсетевых группах в форме исследовательской деятельности. Фрагмент занятия с использованием данного метода представлен ниже (подробнее см. Приложение В).

#### ***Фрагмент мега-урока «Ветвления»:***

Для закрепления полученных знаний по всей пройденной теме урока учащиеся возвращаются к заданию, представленному в начале урока. Задание находится в отдельном документе, ссылка на который имеется в рабочих тетрадях:

Задача выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 группа занимается составлением блок-схемы решения задачи, 3 группа работает над составлением множественных ветвлений. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа [4].

Ниже представлена схема взаимодействия межсетевых групп (см. рис.2)

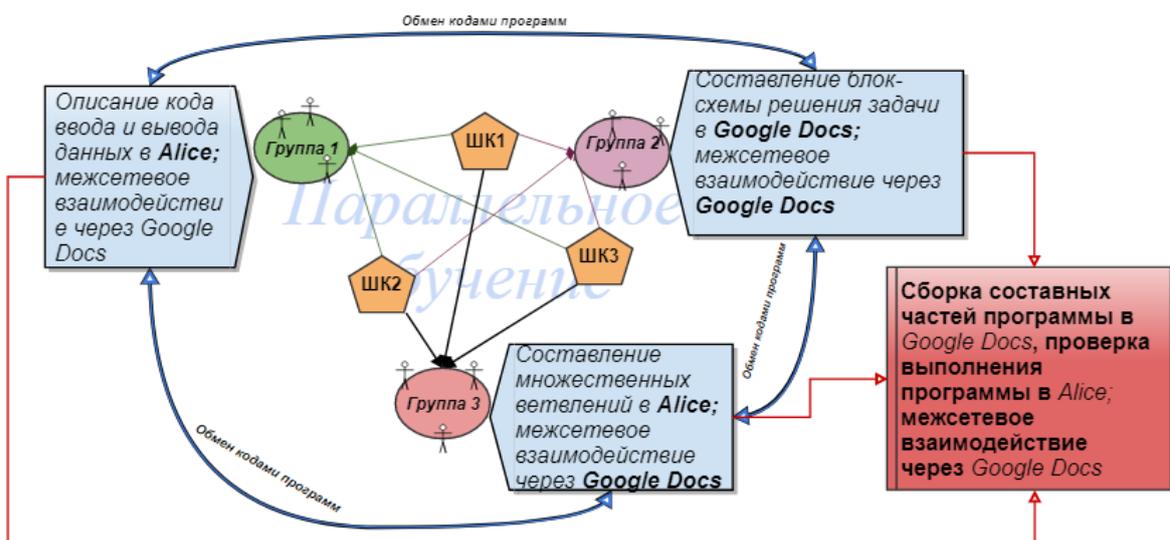


Рис.2. Схема деятельности межсетевых групп

Проведение занятия в формате мега-урока с использованием метода параллельного обучения позволит обучающимся в полном объеме изучить сразу две темы за 1 урок, а именно сформировать:

- знания основных понятий: ветвления, каскадные ветвления;
  - умения применять полученные знания о ветвлениях языка программирования Питон в практической деятельности и при решении творческих задач;
  - развитие своих коммуникативных УУД, когнитивных способностей;
- 4) четвертый мега-урок посвящен изучению следующей темы раздела: «Циклы». Учащиеся также работают в межсетевых группах на основе

«параллельного обучения». Фрагмент занятия с использованием данного метода представлен ниже (подробнее см. Приложение Г).

**Фрагмент мега-урока «Циклы»:**

Учащимся предстоит самостоятельное, более детальное изучение материала: группе №1 предлагается подробное изучение пункта «Цикл for», группе №2 – «Цикл while». Использовать для этого ученики могут имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в рабочей тетради для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет ответы и выставляет баллы в рейтинг.

Учащиеся должны убедиться в правильности разработки программ по своим для того, чтобы наглядно продемонстрировать готовую программу и объяснить изученный материал другим группам. Далее группы обмениваются готовыми программами.

Ниже представлена схема взаимодействия межсетевых групп (см. рис.3)

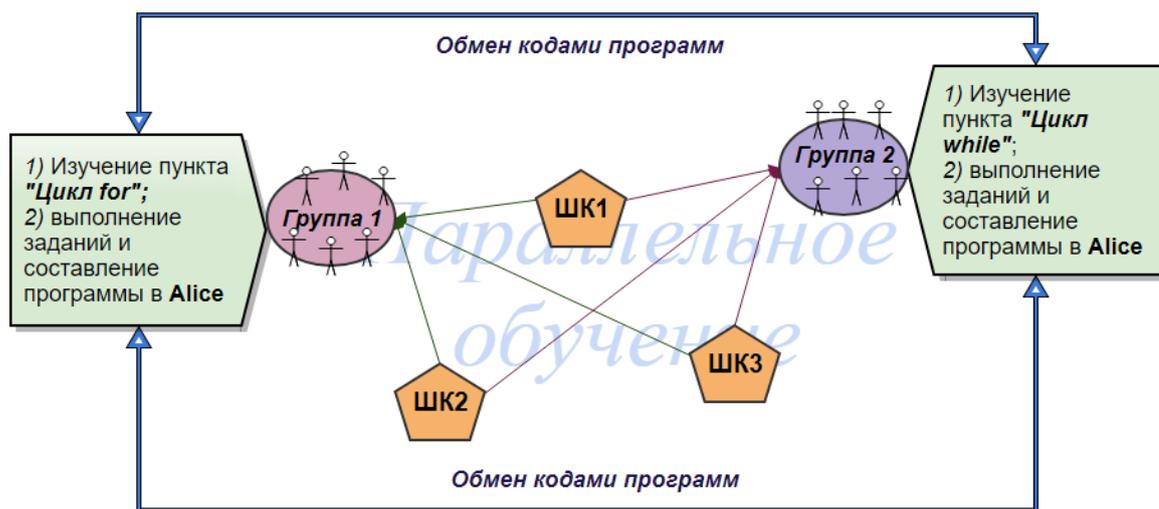


Рис.3. Схема деятельности межсетевых групп

Организация работы групп осуществляется в гугл-пространстве в рабочей тетради. Учебные материалы, используемые на уроке, также предоставлены каждой группе в виде гугл-документов в начале урока (см. Приложение Г).

Проведение занятия в формате мега-урока с использованием метода параллельного обучения позволит обучающимся в полном объеме изучить сразу две темы за 1 урок, а именно сформировать:

- знания основных понятий: циклы, циклы с условием и постусловием, циклы с переменной;
- умения применять полученные знания о циклах языка программирования Питон в практической деятельности и при решении творческих задач;

- развитие своих коммуникативных УУД, когнитивных способностей;

5) пятый мега-урок посвящен изучению следующей темы раздела: «Вложенные циклы». Учащиеся работают в межсетевых группах в форме исследовательской деятельности. Фрагмент занятия с использованием данного метода представлен ниже (подробнее см. Приложение Д).

***Фрагмент мега-урока «Вложенные циклы»:***

Для закрепления полученных знаний по всей пройденной теме урока учащиеся возвращаются к заданию, представленному в начале урока. Задание находится в отдельном документе, ссылка на который имеется в рабочих тетрадях.

Учащиеся разбиваются на 4 межсетевых группы: 1 группа работает над вводом и выводом данных, 2 группа занимается составлением блок-схемы решения задачи, 3 группа работает над составлением множественных ветвлений, 4 группа работает над разработкой циклов. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа [4].

Ниже представлена схема взаимодействия межсетевых групп (см. рис.4)

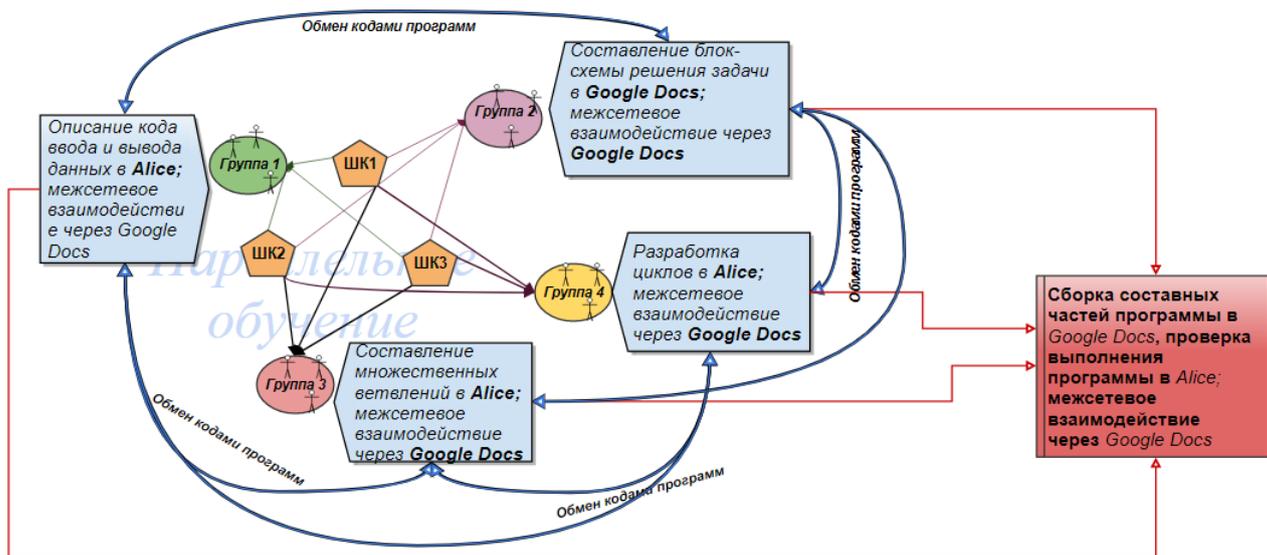


Рис.4. Схема деятельности межсетевых групп

Организация работы групп осуществляется в гугл-пространстве в рабочей тетради. Учебные материалы, используемые на уроке, также предоставлены каждой группе в виде гугл-документов в начале урока (см. Приложение Д).

Проведение занятия в формате мега-урока с использованием метода параллельного обучения позволит обучающимся в полном объеме изучить данную тему, а именно сформировать:

- знания основных вложенных конструкций: вложенные циклы, сложные условия;
- умения применять полученные знания о вложенных циклах языка программирования Питон в практической деятельности и при решении творческих задач;

- развитие своих коммуникативных УУД, когнитивных способностей;

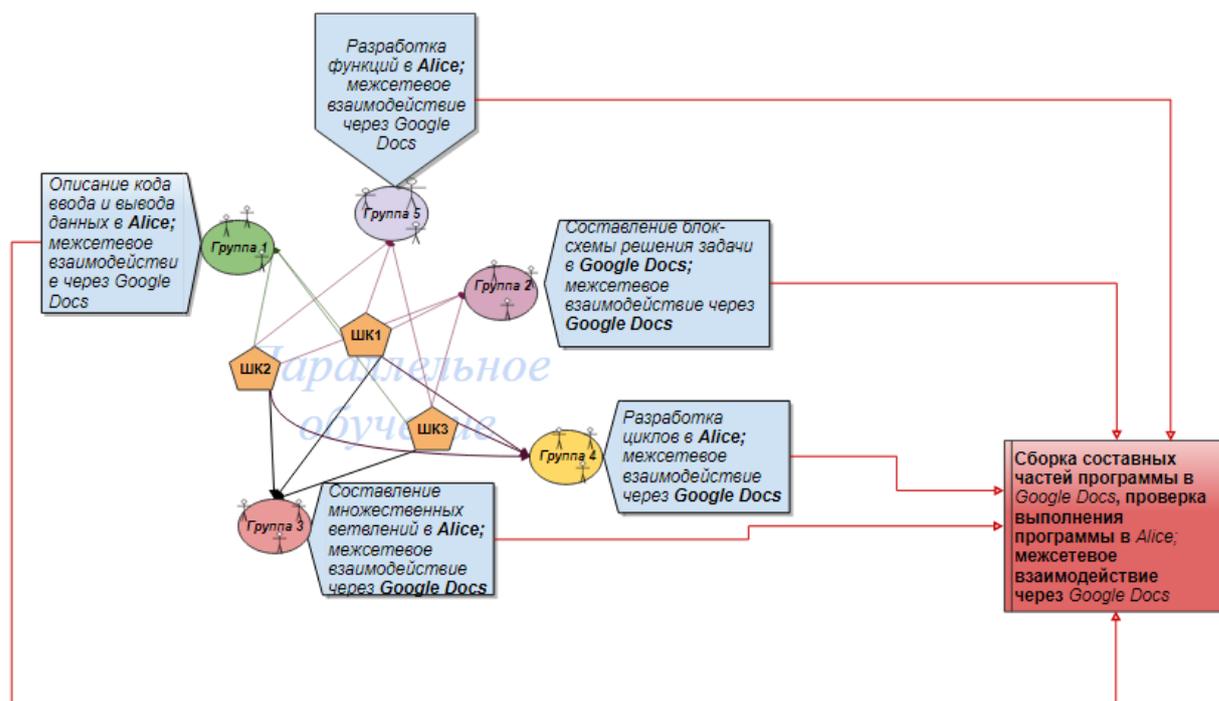
б) шестой урок направлен на изучение следующих тем раздела «Основы алгоритмических конструкций»: «Процедуры» и «Функции». Мега-урок построен на основе нелинейной технологии «параллельного обучения», где обучающиеся в онлайн режиме межсетевых групп работают со своей

частью задачи. Фрагмент урока представлен ниже (подробнее см. Приложение Е).

### **Фрагмент мега-урока «Подпрограммы в Python»:**

Для закрепления полученных знаний по всей пройденной теме урока учащимся предоставляется более сложное задание в отдельном документе, ссылку на который группы получают по завершении теоретико-практической части урока. Учащиеся делятся на 4 межсетевые группы. Задача выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 занимается составлением блок-схемы решения задачи, 3 группа занимается составлением ветвлений, 4 группа занимается составлением циклов, 5 группа занимается составлением функций. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа.

Ниже представлена схема взаимодействия межсетевых групп (см. рис.5)



*Рис.5. Схема деятельности межсетевых групп*

Организация взаимодействия обучающихся и тьютора осуществляется

в онлайн-пространстве гугл-документа с заданиями. Учебные материалы, используемые на уроке, также предоставлены каждой группе в виде гугл-документов в начале урока (см. Приложение Е).

Проведение занятия в формате мега-урока с использованием «параллельного обучения» позволит обучающимся в полном объеме изучить сразу две темы за 1 урок, а именно сформировать:

- знание основных понятий: процедуры, процедуры с параметром, локальные и глобальные переменные, функции, логические функции;
- умения и навыки применения полученных знаний о процедурах и функциях на практике и при решении интересных, познавательных задач;
- развитие коммуникативных УУД и познавательных способностей;

7) последний урок по изучению базовых алгоритмических конструкций посвящен рекурсии. Ученикам в группах предоставляется учебный материал на гугл-диске для самостоятельного изучения материала, а также для первичного закрепления предлагается решить несколько разноуровневых задач (подробнее см. Приложение Д). Далее урок организован на основе метода «параллельного обучения», где учащимся предлагается решить сложную задачу, где должны быть применены все изученные ранее темы. Фрагмент урока представлен ниже (подробнее см. Приложение Ж).

#### ***Фрагмент мега-урока «Рекурсия»:***

После обсуждения выполнения заданий учащимся на закрепление предлагается написать программу к задаче более высокого уровня сложности. Задача предоставляется всем группам в виде ссылки на отдельный гугл-документ и выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 занимается составлением блок-схемы решения задачи, 3 группа занимается составлением ветвлений, 4 группа занимается составлением циклов, 5 группа занимается составлением функций, 6 группа составляет рекурсии к задаче. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут

обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа.

В конце урока готовая программа выводится на экран, где педагог вуза (модератор) комментирует код программы и подсказывает другие варианты решения.

Ниже представлена схема взаимодействия межсетевых групп (см. рис.6)

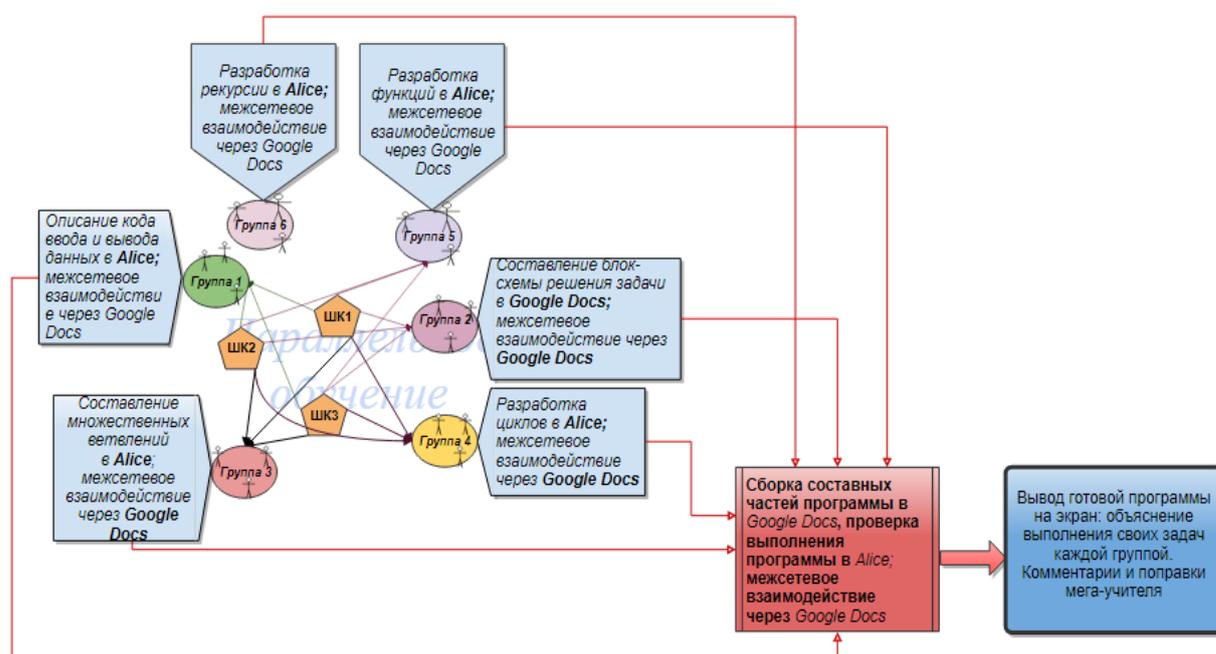


Рис.6. Схема деятельности межсетевых групп

Для организации взаимодействия групп между собой и тьютором в гугл-документе с заданиями имеется диалоговое окно.

При данной организации урока с использованием нелинейных технологий у обучающихся происходит формирование:

- знаний основных понятий: рекурсия, рекурсивный алгоритм, итерационный алгоритм, глубина рекурсии;
- навыков и умений использовать полученные знания на предыдущих занятиях, а также нового материала на практике;
- развитие коммуникационных УУД и логического мышления.

Разработанная система мега-уроков по разделу «Основы алгоритмических конструкций» позволяет достичь следующих результатов:

1) формирование алгоритмической культуры учащихся, способности планировать пути достижения цели, в том числе альтернативные, а также умения соотносить свои действия с планируемыми результатами, осуществление контроля своей деятельности в процессе достижения результата;

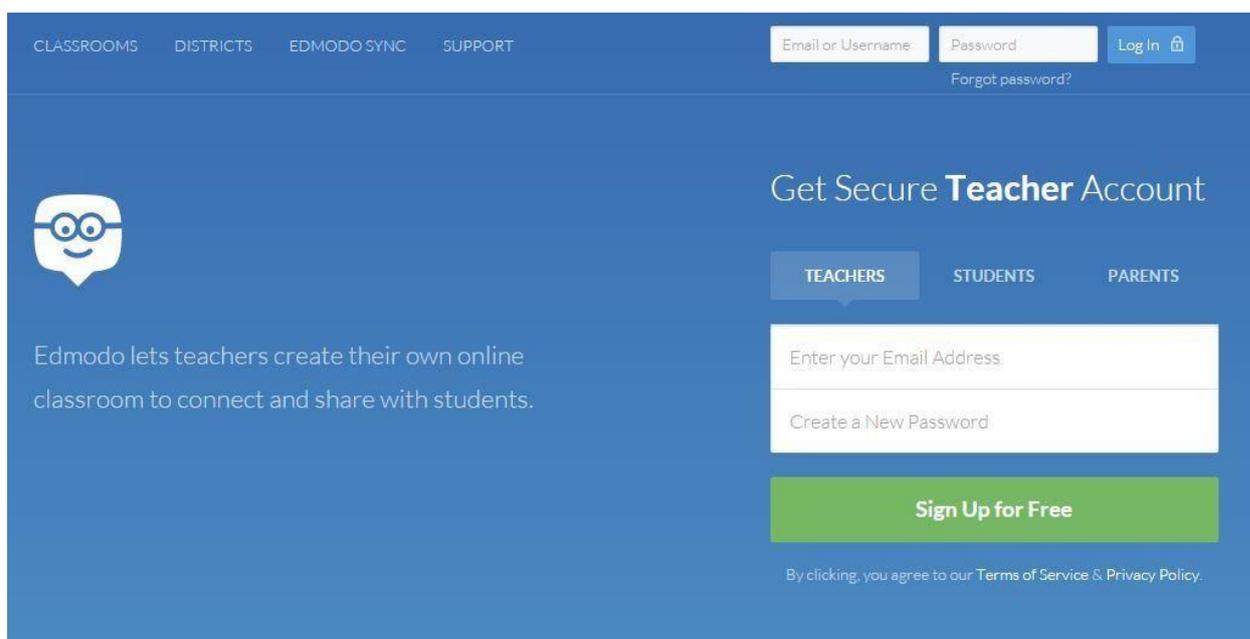
2) формирование навыков формализованного описания алгоритма в условиях жестких ограничений языка программирования Python и среды программного исполнителя, как условий автоматизации процесса исполнения алгоритма;

3) развитие алгоритмического мышления учащихся: умение выстраивать план действий, предвидеть результат, формально исполнять последовательность действий, понимать последовательные, параллельные действия и умение анализировать результат.

### **2.3. Формы организации работы обучающихся в сетевом кластере и контроль результатов их деятельности**

Организация работы обучающихся включает в себя множество различных средств взаимодействия, учебников, методик, виртуальных сред и т.п.. Наша задача отобрать эффективные, в первую очередь нацеленные на программирование, классифицировать, и объединить в систему все вышеперечисленное для достижения поставленных ранее целей обучения.

**Edmodo** — виртуальная среда для обучения и создания сообществ, своеобразная социальная сеть (см. рис.7).



*Рис.7. Начальная страница сайта edmodo.com*

С помощью этого сервиса есть возможность организовать дистанционное обучение. Каждый класс является закрытой площадкой для общения, создания библиотек документов и ссылок, ведения оценивания, отслеживания статистики успеваемости, выполнения различных заданий. Можно создавать внутри класса мини-группы для обсуждения вопросов и взаимодействия.

Учитель имеет возможность создать группу, получить код на неё, и все обладатели кода (ученики) могут вступать в эту группу. Есть возможность импортировать ленты с сайтов, вкладывать рисунки или же видео. Есть календарь для фиксации расписания зачетов. Так же имеется возможность ввести задание с датой выполнения (например - записать домашнее задание), и раздел для выставления оценок.

Отличия Edmodo от классических социальных сетей:

- нет внешней рекламы, отвлекающего контента;
- регистрация упрощена, в анкетах не просится указывать дни рождения, точные фамилию, имя отчество и прочую идентифицирующую информацию;
- пользователи делятся на три группы: учителя, учащиеся, родители;

- никакой персональной информации по учащимся от школы не требуется. В соглашении о персональной информации точно указывается, какая загруженная на сайт информация может обрабатываться и как.

Для пользования этим сервисом больше всего подходят ученики старших классов. И совсем не обязательно, что это будет дистанционное обучение. Можно просто создать среду для активного взаимодействия учителя с учениками помимо школы, например, для того же контроля знаний.

**GitHub** — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности, а для частных проектов предлагаются различные платные тарифные планы. Слоган сервиса — «SocialCoding» — на русский можно перевести как «Пишем код вместе» (см. рис.8).



*Рис.8. Главная страница веб-сервиса GitHub*

Создатели сайта называют GitHub «социальной сетью для разработчиков». Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых.

С помощью широких возможностей GitHub программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева.

Фактически git - это система контроля версий, встраиваемая в практически все современные программные среды.

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок. Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

А также:

- на платных тарифных планах можно создавать приватные репозитории, доступные ограниченному кругу пользователей;
- есть возможность прямого добавления новых файлов в свой репозиторий через веб-интерфейс сервиса;
- код проектов можно не только скопировать через Git, но и скачать в виде обычных архивов с сайта (для этого достаточно добавить /zipball/master/ в конец адресной строки);
- кроме Git, сервис поддерживает получение и редактирование кода через SVN и Mercurial;
- на сайте есть pastebin-сервис [gist.github.com](https://gist.github.com) для быстрой публикации фрагментов кода.

Стоит только зарегистрироваться и огромный набор бесплатных возможностей вам обеспечен. Единственный сайт не русифицирован.

Облачный сервис Linoit – онлайн доска, с помощью которой создаются «полотна», на которые крепятся листы - стикеры (рис.). Это происходит мгновенно и прекрасно заменяет пересылку через e-mail, Skype, mail – агента и др. То есть при работе в облаке «Linoit» процесс организован в одном web-пространстве, обеспечен всем необходимым и не требует лишних переключений (см. рис.9).



Рис.9. Интерактивная доска Linoit

В образовании может служить не только отличным наглядным средством, но и платформой для совместной продуктивной деятельности учителя и учеников. Доска предоставляет неограниченное место для ваших идей и проектов, а отличные инструменты редактирования позволяют наполнить это пространство нужным вам смыслом.

**HotPotatoes** - универсальная программа-оболочка, позволяющая любому человеку самостоятельно, не прибегая к помощи программистов, создавать интерактивные тренировочно-контролирующие упражнения в формате HTML.

Программа широко используется во всем мире для создания заданий на различных языках по различным дисциплинам.



[Home](#) [News](#) [Support](#) [Downloads](#) [HotPot Sites](#) [Tutorials](#) [Bugs/Updates](#) [FAQ](#) [Extras](#)

## Hot Potatoes Home Page

### News - 12/06/2013

- Dr. Stan Bogdanov has published [Hacking Hot Potatoes: The Cookbook](#), available in paperback, PDF and ePub format. Check it out!
- A change to the user agent string in Firefox 17 results in Hot Potatoes and Quandary exercises showing an error message when loading in the browser. Version [6.3.0.5 \(Windows\)](#) of Hot Potatoes, [6.1.0.6 \(Java for Mac\)](#) of Hot Potatoes and version 2.4.2.1 of [Quandary](#) (Windows) have updated source files which fix the problem, but you will need to rebuild your exercise HTML pages to solve it. See the [Bugs and Updates](#) page for more information.
- Hot Potatoes is now freeware. The complete version of the programs is now available for free, from the [Downloads](#) section below. The free version of Hot Potatoes for Windows is version 6.3, and the Java version is 6.1.
- [Quandary](#), our other authoring tool for creating Web-based action mazes, is also now free.
- We no longer provide any technical support for Hot Potatoes or Quandary.

*Рис.10. Начальная страница сайта hotpot.uvic.ca*

Среди многих интерактивных тестовых программ, эту программу отличает то, что интерактивные задания возможно сохранять в архиве и использовать их без подключения к интернету. Созданные с помощью нее различного рода упражнения настолько разнообразны и увлекательны, что могут служить не только для оценки знаний учащихся, но и в качестве их индивидуальной творческой работы. В программе можно составлять всевозможные кроссворды, викторины и тесты. Есть возможность включения различных аудио и видеоматериалов, возможность создавать задания на установление соответствия и по заполнению пропусков в тексте.

HotPotatoes совершенно несложная в использовании программа, при этом она эффективно выполняет свои главные задачи. Скачивайте, устанавливайте и создавайте свои собственные тренировочные задания.

**LearningApps** является очень интересным приложением Web 2.0 для поддержки процесса преподавания с помощью интерактивных модулей.

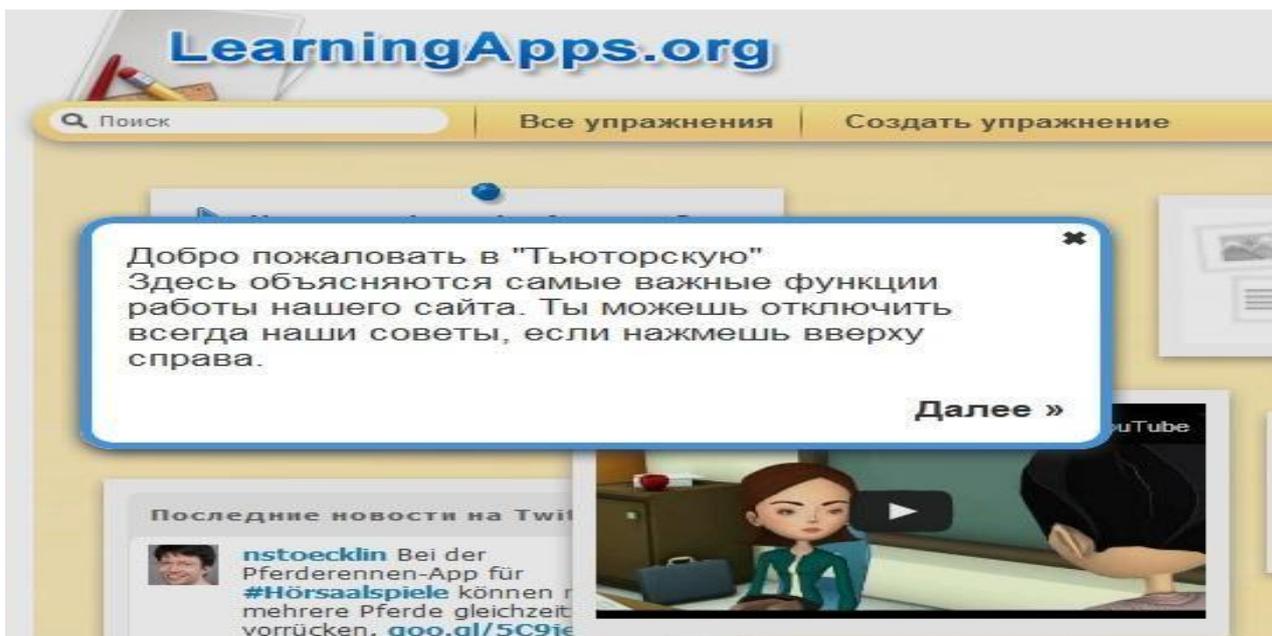


Рис.11. «Тьюторская» приложения LearningApps

На сайте даны все указания по работе, подсказки, в общем, весь материал для подготовки приложений. Можно создать свое упражнение, а можно сделать подобное из вкладки «все упражнения». Упражнения не однотипные, они подразделяются на множество видов. Для обучения информатике данное приложение предлагает огромные возможности, особенно для тренировки навыков или знаний теоретического материала. Например, если предложить ребятам вместо теста по закреплению знаний основных алгоритмических конструкций поиграть в игру «Кто хочет стать миллионером», вероятность того, что они с интересом включатся в образовательный процесс, будет намного выше.

Объектно-ориентированная среда Alice позволяет управлять трехмерными объектами, создавать программы (компьютерные видеоигры и анимационные приложения-фильмы), генерирующие анимацию в виртуальных мирах на основе использования условий, циклов, функций/методов, обработки массивов данных, списков, объектов, наследования, инкапсуляции, полиморфизма и т.д.

В отличие от распространенного в обучении программированию подхода, когда в первую очередь знакомятся с понятием данных и алгоритмов, Alice использует объектно-ориентированное программирование

изначально. Проведенные создателями проекта исследования показали, что это способствует более глубокому пониманию концепции объектно-ориентированного программирования — ведь если обучаемые сразу начинают оперировать объектными понятиями, то им нет необходимости переучиваться, что неизбежно происходит при переходе от одной парадигмы программирования к другой.

Среда программирования Alice выпускается для Windows, Mac OS и Linux в двух редакциях: основной (для вузов) и упрощенной (для школ). В среде имеется большая библиотека объемных объектов из реального мира (природа, животные, инструменты быта и т.д.). Их можно двигать, вращать, менять цвет и размер при помощи мыши, а на основе полученного виртуального мира программно описывать анимацию и создавать игровые модели (см. рис. 12).

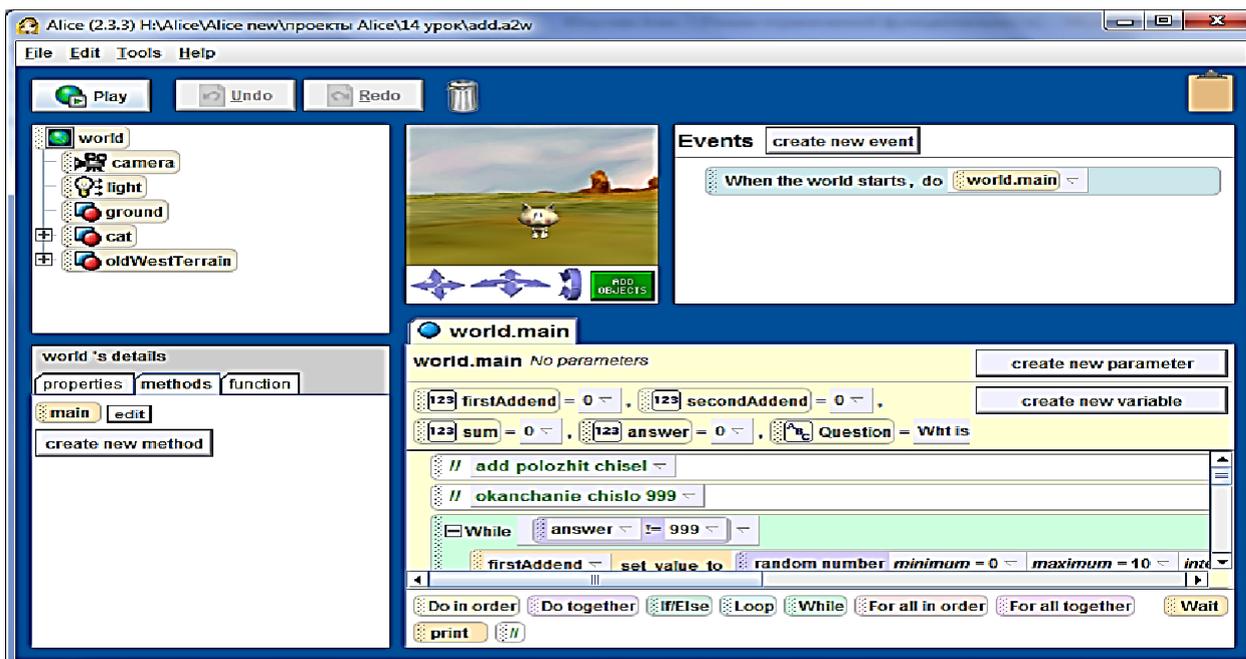


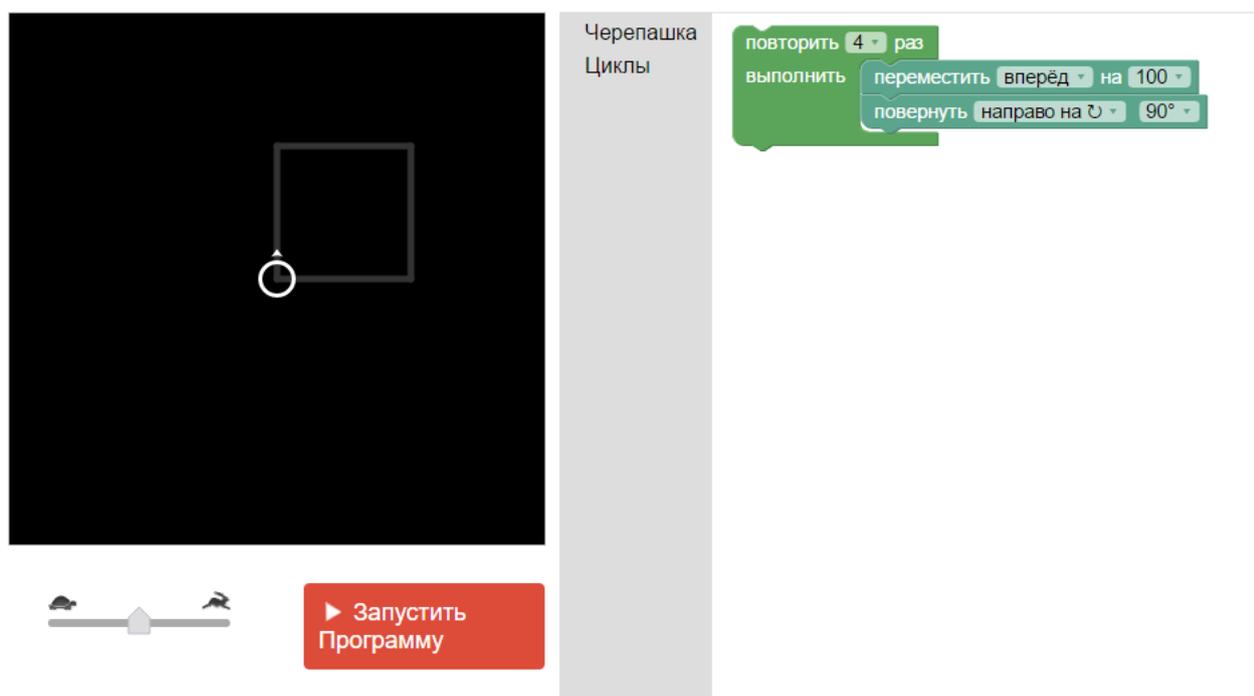
Рис.12. Окно программы

В среде Alice используется собственный встроенный язык программирования, приближенный к синтаксису языков современных объектно-ориентированных языков программирования таких, как Java, C++ или VisualBasic. Поскольку программное обеспечение Alice позволяет создавать только синтаксически правильные команды, то программирование в данной среде сводится лишь к разработке и реализации соответствующих

алгоритмов. Учащемуся не нужно запоминать синтаксис какой-либо конструкции, он всегда может воспользоваться набором имеющихся процедур и функций рассматриваемого объекта или всплывающими подсказками. Именно это позволяет школьникам в дальнейшем сконцентрировать свое внимание на сценарии игры, сцены, используемых объектах, их свойствах и методах, а не беспокоиться о синтаксических ошибках своих приложений.

Для учеников, только начинающих изучать программирование, можно использовать среду **Blockly**. Эта среда опирается на использование наглядно-деятельностного уровня мышления, которое в принципе выражено у всех учащихся. Если у человека имеются проблемы в переводе конструкций алгоритма на язык программирования ему может помочь Blockly. Важная особенность графической среды: освобождение от заботы о синтаксисе, что позволяет начинающему сосредоточиться на логике программы. Можно интегрировать редактор в собственные веб-приложения, чтобы дать возможность начинающим пользователям создать свой интерфейс. Созданную с помощью этого редактора программу перевести на код JavaScript, Python, PHP, Dart, или какой-либо другой язык: Приложение может выполнить код по желанию. Blockly поставляется с большим количеством predefined блоков. Все, начиная от математических функций цикла структуры.

Пример: построение квадрата с помощью блоков в среде Blockly (см. рис.13).



*Рис.13. Вид программы построения квадрата в среде Blockly*

В сети можно найти достаточное количество примеров и других языков и их редакторов. Плюсы понятны: при программировании с использованием визуальных объектов человек описывается на образы, обучение активизирует различные уровни мышления. Но минус - может произойти торможение в развитии программирования, в человеческой истории все языки, опирающиеся на иероглифы, приводили к дополнительным сложностям в обучении.

Для разработки системы уроков и задач были выбраны следующие формы организации работы и контроля:

- 1) среда программирования Alice, в которой обучающиеся составляют программы для решения практико-ориентированных задач;
- 2) облачный сервис Linoit для организации общения групп между собой и тьютором, а также облачный сервис google-docs;
- 3) язык программирования Python. Данный язык удобен в изучении школьниками только в том случае, если учащиеся уже знакомы с одним из строго-типизированных языков. Преимуществом Python является то, что синтаксис написанных на нем программ всегда читабельный. Также данный

язык программирования компактен и легок в обучении, а динамическая типизация позволяет сократить количество переменных.

Изучение программирования в старших классах будет интересно и понятно для обучающихся, т.к. Python - интерпретируемый язык программирования, который обладает рядом преимуществ:

1. Динамическая типизация. В Python не надо заранее объявлять тип переменной, что очень удобно при разработке.

2. Хорошая поддержка модульности. Вы можете легко написать свой модуль в других программах.

3. Встроенная поддержка Unicode в строках. В Python необязательно писать всё на английском языке, в программах вполне может использоваться ваш родной язык.

4. Поддержка объектно-ориентированного программирования. При этом его реализация в Python является одной из самых понятных.

5. Автоматическая сборка мусора, отсутствие утечек памяти.

6. Понятный и лаконичный синтаксис, способствующий ясному отображению кода. Удобная система функций позволяет при грамотном подходе создавать код, в котором будет легко разобраться другому человеку в случае необходимости.

7. Огромное количество модулей, как входящих в стандартную поставку Python 3, так и сторонних. В некоторых случаях для написания программы достаточно лишь найти подходящие модули и правильно их скомбинировать.

Кроссплатформенность. Программа, написанная на Python, будет функционировать совершенно одинаково вне зависимости от того, в какой операционной системе она запущена [7,30,35].

Также следует отметить, что при проведении каждого урока ведется контроль результатов деятельности обучающихся. В обучении программированию, а именно раздела «Основы алгоритмических конструкций», были применены следующие формы контроля:

предварительный, текущий и итоговый контроль.

**Предварительный контроль.** Данная форма контроля имеет диагностические задачи и проводится с целью выявления имеющихся знаний, умений и навыков учащихся к началу обучения. Предварительный контроль позволяет учителю находить оптимальные формы и методы работы.

При разработке первого урока была использована форма предварительного контроля методом выполнения практических заданий для проверки остаточных знаний о программном обеспечении компьютера.

**Текущий контроль.** Данная форма контроля осуществляется по ходу обучения и позволяет определить степень сформированности знаний, умений, навыков, а также их прочность и глубину. С помощью этого контроля есть возможность своевременного выявления пробелов в знаниях обучающихся и оказания помощи в усвоении материала.

Текущий контроль на каждом мега-уроке осуществляется методом постоянного наблюдения мега-учителя и тьюторов за работой учащихся, проверок домашнего задания, фронтального опроса и выполнения практических заданий в гугл-пространстве:

- фронтальный опрос. Для получения экспресс информации о степени готовности обучающихся к усвоению нового материала и актуализации имеющихся знаний мега-учитель проводит фронтальный опрос (вопросы обращены ко всему мегаклассу). Фронтальный опрос должен проходить в быстром темпе и не занимать много времени на уроке;
- проверка домашнего задания – метод самоконтроля - осуществляется непосредственно перед началом занятия с целью выявления пробелов в знаниях и актуализации знаний (каждый ученик может зайти и посмотреть проверенное домашнее задание с целью развития умений самостоятельно находить допущенные ошибки, неточности, намечать способы устранения обнаруживаемых пробелов)
- практические задания в межсетевом пространстве при изучении нового материала ставят своей целью выявление сформированности знаний,

умений и навыков практической деятельности обучающихся:

а) умения применять полученные знания об основных алгоритмических конструкциях при анализе условия задачи, составлении предварительного алгоритма решения, написании программ;

б) развитие алгоритмического мышления обучающихся, а также навыков межсетевого общения;

- педагогические тесты. Данный метод контроля позволяет оценивать результаты деятельности обучающихся с более высокой степенью объективности. Объективность обеспечивается за счет стандартизации вопросов и ответов, особой процедуры проведения тестирования и способов обработки результатов.

Для проверки текущего контроля был разработан тест закрытого типа (с выбором ответа) в мега-уроке по теме «Введение в программирование. Алгоритм и его свойства». Тесты открытого типа (со свободным ответом) были разработаны в качестве проверки усвоения знаний обучающимися в следующих мега-уроках: «Ветвления», «Циклы» и «Подпрограммы в Python».

**Итоговый контроль.** Данная форма контроля призвана определить конечные результаты обучения. Он охватывает всю систему знаний, умений и навыков по разделу.

Итоговый контроль осуществляется на крайнем занятии при изучении темы «Рекурсия». Обучающимся для определения качества усвоения знаний, умений и навыков практической деятельности по всему пройденному разделу «Основы алгоритмических конструкций» предлагается практическое задание высокого уровня сложности, требующее применения знаний всего изученного раздела.

Обобщив вышеизложенное, можно составить схему деятельности обучающихся на мега-уроках (рис.14):

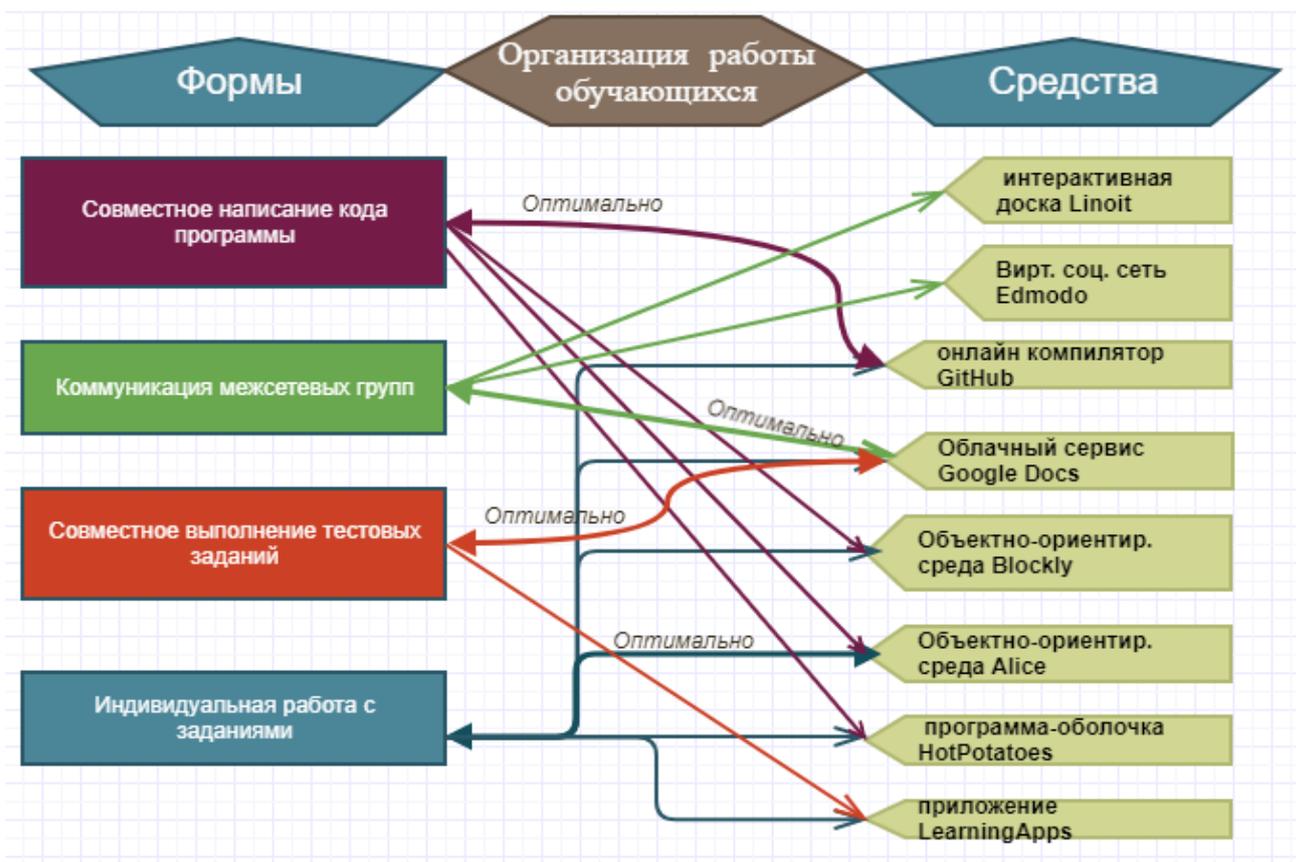


Рис.14. Схема организации работы учащихся на мега-уроках

Использование вышеизложенных средств для организации работы обучающихся в совокупности с формами контроля результатов их деятельности, используемых в системе разработанных мега-уроков, дает более точное определение результатов обучения с учетом их достижения, а также выявляет тенденции, динамику образовательного процесса.

## **Выводы по главе 2**

Проанализировав цели и задачи обучения школьников программированию в образовательном кластере, была разработана система мега-уроков по программированию с использованием нелинейного подхода в соответствии с разделом «Основы алгоритмических конструкций». В разработку системы мега-уроков входит ряд задач, которые были выполнены, а именно:

1) выделены основные формы организации работы в сетевом кластере, такие как межсетевое групповое взаимодействие обучающихся, совместное написание кода программы и др.;

2) выявлены особые виды деятельности учащихся на мегауроках, отличающие проведение данных занятий от традиционных (например, исследовательская деятельность);

3) проанализированы и применены в разработке системы мегауроков наиболее оптимальные средства для организации работы межсетевых групп: облачное пространство Google Docs и интерактивная доска Linoit для организации межсетевого взаимодействия, объектно-ориентированная среда Alice для написания и проверки кодов программ и др.;

4) проанализированы и применены в разработке мега-уроков различные формы контроля обучающихся в образовательном кластере, такие как наблюдение и помощь тьюторов в облачных сервисах при выполнении заданий межсетевыми группами, контроль мега-учителем организации всего учебного процесса и др.

Организационные формы обучения в условиях открытого образовательного кластера, применяемые при создании мега-уроков, ориентируют учащихся разных школ на активное межсетевое взаимодействие, организованное посредством объединения в межшкольные группы сменного состава и соответствующее целям и особенностям проведения конкретного урока. Использование нелинейных подходов, основанных на самостоятельном изучении материала при поддержке и

помощи мега-учителя и тьюторов и применяемых в разработке системы мега-уроков, а также использование средств организации работы обучающихся и различных форм контроля результатов их деятельности, благоприятно влияет на развитие познавательных потребностей обучающихся в программировании, алгоритмического мышления, мотивации к обучению программированию [4].

## **Заключение**

Цель настоящего исследования состоит в разработке системы мега-уроков по разделу «Основы алгоритмических конструкций» для 10 класса инженерно-технологического образовательного кластера на основе нелинейного обучения.

Для достижения поставленной цели были выполнены следующие задачи:

1. Проведен анализ методической литературы по теме исследования. Выделены особенности нелинейной технологии обучения в применении к программированию и сущность и особенности организации учебного процесса в образовательном кластере (на примере образовательной платформы мегакласс).

2. Выявлены особенности обучения программированию старшеклассников инженерно-технологического профиля.

3. Выбрано содержание системы мега-уроков на базе практико-ориентированных задач, отобраны методы, приемы обучения, описаны формы организации обучения по сетевой модели с использованием нелинейных технологий обучения. Разработаны уроки в соответствии с применяемыми технологиями обучения и образовательным кластером.

Таким образом, задачи решены в полном объеме, цель данной работы достигнута - разработана система мега-уроков по программированию с использованием нелинейной технологии обучения.

### Список использованных источников

1. Андреева Е.В. Алгоритмизация и программирование в школьном курсе информатики // Информатика. Выпуск 1,2. – 2008. – №14. – С.48.
2. Бешенков, Ракитина, Миндзаева: Примерные программы среднего (полного) общего образования: информатика. 10-11 классы. ФГОС // Изд. «Вентана-граф». 2012, 56 с.
3. Елисеева Л.В. Инновационные технологии как средство совершенствования методики преподавания информатики// Информационные технологии в образовании [Электронный ресурс]. URL: <http://tmo.ito.edu.ru/2013/section/222/96093/>(Дата обращения: 13.05.2018).
4. Ивкина Л.М., Кулакова И.А., Пак Н.И., Романов Д.В., Симонова А.Л., Сокольская М.А., Хегай Л.Б., Яковлева Т.А. Мегакласс как инновационная модель обучения информатике с использованием ДОТ и СПО: коллективная монография //Краснояр. гос. пед. ун-т им. В.П. Астафьева. — Красноярск, 2014 г.
5. Ивкина Л.М., Методическое сопровождение в условиях глобализации учебного процесса / Л.М. Ивкина, Л.Б. Хегай //Информатика и образование – № 10 (269), 2015 г.
6. Ивкина Л.М., Формирование методической готовности будущих учителей информатики в условиях образовательной платформы «Мега-класс»: дисс. ... к-та пед. наук. Красноярск, СФУ, 2017 г.
7. Изучаем питон [Электронный ресурс]. URL: <http://slusar.su/uroki/izuchaem-python-vse-uroki/> (дата обращения: 02.05.18).
8. Информатика и ИКТ. 11 класс. Базовый уровень. Угринович Н.Д. М.: Бином. Лаборатория знаний, 2008. — 188 с. 8.

9. Лаборатория линуксоида/ Python. Обучение программированию [Электронный ресурс]. URL: <https://younglinux.info/python> (дата обращения: 02.05.18).
10. Лапчик М.П. и др. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов / М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер; Под общей ред. М.П. Лапчика. – М: Издательский центр «Академия», 2001. – 624с
11. М. А. Сокольская О методах обучения бакалавров направления «педагогическое образование» по профилю «информатика» основам параллельного программирования. [Электронный ресурс]. URL: <http://fan5.ru/fan5-docx/doc-126160.php> (дата обращения: 20.04.18).
12. Методические вопросы построения графических задач на примере языка Python / М А.В.Багаев, Н.А.Александрова // Информационные технологии в образовании: Материалы VII Всерос. Научно-практ. Конф. – Саратов. ООО Изд. «Наука», 2015. С. 14-19.
13. Методологические и теоретические подходы к решению проблем практики образования [Текст]: сборник статей. – Красноярск, 2004. –112 с.
14. Мирончик Е.А. Развитие логического и алгоритмического мышления учащихся на уроках информатики // Информатика и образования. –2008. – №4. – С.17-19.
15. Нельзина О.В. Проблемы обучения программированию по курсу информатики в системе «школа-вуз» // Вопросы Интернет - образования. – 2006. – №13.
16. Нилова Ю.Н. Методика обучения программированию учащихся старшей школы на основе системно-деятельностного подхода: дис.канд.пед. наук: 13.00.02 / Ю.Н. Нилова – СПб, 2015. – 244 с.
17. Пак Н. И. Нелинейные технологии обучения в условиях информатизации образования. – Красноярск: РИО КГПУ, 2004 г.
18. Пак Н.И. Инновационная технология «Мега-класс» как синергетическое средство обучения в образовательных кластерах /Сб. трудов

Международной научно-практической конференции «Информатизация образования-2015», Казань, 2015, с. 288-294.

19. Пак Н.И. Стратегии информационного подхода в проектировании кластерной системы образования школа-педвуз //В сборнике: Фундаментальные науки и образование Материалы II международной научно-практической конференции (02-05 марта 2014 г.). Бийск. – 2014, – с.66-76.

20. Пак Н.И., Сокольская М.А. Региональная модель образовательного кластера на технологической платформе «Мега-класс». Красноярский государственный педагогический университет.

21. Пак, Н.И. Использование параллельных технологий обучения в курсах информатики / Н.И. Пак, Т.А. Степанова // Новые информационные технологии в университетском образовании: тез. конф. – Новосибирск: СГУПС, ИДМИ, 2007. – С.120.

22. Питон для школы. Олимпиадные задачи. [Электронный ресурс]. URL: <http://python.inspweb.ru/index.php/olimpiadnye-zadachi> (дата обращения:)

23. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013 — Ч.1 - 344с., Ч.2 - 304с.

24. Психологические особенности изучения программирования в старших классах: novainfo./автор - А.В.Багаев, В.А.Векслер. / [Электронный ресурс]. URL: <http://novainfo.ru/archive/34/psikhologicheskie-osobennosti-izuchenie-programirovaniya> (дата обращения:10.03.18).

25. Розалинд – платформа для изучения биоинформатики/ Rosalind. [Электронный ресурс]. URL: <http://rosalind.info/problems/locations/> (дата обращения: 03.05.18).

26. Степанова Т.А. Методические условия развития алгоритмического мышления школьников на уроках информатики. / Информатика в школе: прошлое, настоящее и будущее: материалы Всеросс. науч.-метод. конф. по

- вопросам применения ИКТ в образовании, 6 – 7 февраля 2014 г. / отв. За вып. Ю,А, Аляев, И.Г. Семакин; Перм. Гос. нац. исслед. ун-т. – Пермь, 2014. – 282 с. [Электронный ресурс]. URL: <http://confer.cschool.perm.ru> (дата обращения 20.04.2018).
27. Сукин И.А. Python, проглатывающий слона // Информатика, № 2, 2012, с. 22–42.
28. Теория и методика обучения информатике: учебник / под ред. М. П. Лапчика. М.: Академия, 2008. 592 с.
29. УМК «Информатика и ИКТ» 10-11 класс Профильный уровень. М. Е. Фиошин. Изд.: Дрофа, 2013.
30. Учебник Python/ Python Software Foundation, 2018 [Электронный ресурс] URL: <https://docs.python.org/3/tutorial/index.html> (дата обращения 18.03.2018).
31. Хуторский, А. Деятельность как содержание образования / А. Хуторский // Народное образование. – 2003. – №8. – С.107-114.
32. Ч.Северенс, Введение в программирование на Python -М.: Национальный Открытый Университет «Интуит», 2-е изд., исправленное, 2016 г.
33. Шауцукова Л.З. Информатика 10 - 11. — М.: Просвещение, 2000 г [Электронный ресурс]. Режим доступа: <http://www.tomsk.ru/Books/informatica/theory/index.html>.13.06.2014(дата обращения:05.05.18).
34. Язык программирования Python: Учеб. пособие / Р. А. Сузи . – 2. изд., испр . – Москва: Интернет-Университет информационных технологий : Бином. Лаборатория знаний, 2007 . – 326 с.
35. Язык программирования Python. Copyright © 2001-2018 [Электронный ресурс] URL: <https://www.python.org/> (дата обращения:05.05.18).

*Сценарий мега-урока «Введение в программирование. Алгоритм и его свойства»*

Что необходимо, чтобы создать рисунок на компьютере? Без чего не обойтись при оформлении доклада? Что потребуется, если появиться необходимость выйти в интернет и посетить какой-нибудь сайт? Программа! Все, что делается на компьютере, делается с помощью той или иной программы. Даже сама операционная система – это программа, которая управляет работой компьютера. Без программного обеспечения любой компьютера – просто «железо».

Чтобы обеспечить систематизацию и углубление полученных знаний по разделу «Программное обеспечение компьютера», а также перейти к разделу «Алгоритмизация и программирование» на базе изучения языка программирования Python, было принято решение разработать Мега-урок, на котором учащиеся в режиме межшкольных групп взаимодействуют между собой и выполняют разноуровневые задания в online пространстве.

Для подготовки к уроку решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (Рис.15).

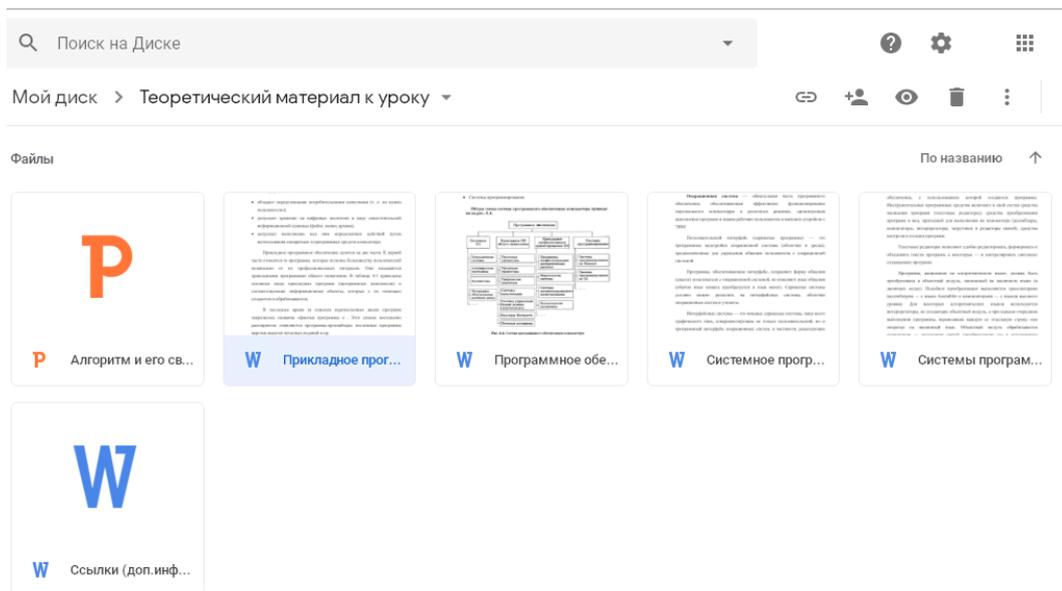


Рис.15. Теоретический материал к уроку

Подробнее можно посмотреть по ссылке: [\[https://drive.google.com/open?id=0B\\_SeLy8qkdb7eXVGMS1lb1MyNEU\]](https://drive.google.com/open?id=0B_SeLy8qkdb7eXVGMS1lb1MyNEU). Он состоит из шести документов в форматах docx и pptx, а именно:

1. «Прикладное программное обеспечение», в котором описаны составляющие прикладного ПО, а также предоставлены примеры программ в табличной форме.
2. «Программное обеспечение», в котором содержится определение понятия ПО и приведена общая схема состава ПО.
3. «Системное программное обеспечение», содержание которого представлено в подробном описании составляющих и примеров программ системного ПО.
4. «Системы программирования», в содержание которого входит описание состава инструментальных ПО, а также примеры их программ.
5. «Алгоритм и его свойства» - презентация, в которой описаны основные свойства алгоритма, способы его записи, приведены примеры записи алгоритма.
6. «Ссылки (доп. информация по алгоритму)», в котором предоставлены ссылки на дополнительную, более обширную информацию об алгоритме.

Также для каждого составляющего ПО описано, как выбрать необходимую программу для конкретной задачи. Доступ к материалу предоставляется за две недели до мега-урока, чтобы дети могли самостоятельно изучать его дома. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Содержание задач при их решении имеет важное значение для повышения мотивации познавательной деятельности учащихся. Задание должно заинтересовать учащегося. Данные требования и стали определяющими содержания заданий в системе разноуровневых задач для мега-урока. Система задач включает в себя 2 части. Первая часть посвящена систематизации знаний по теме «Программное обеспечение компьютера», задания которой включают 3 уровня сложности, внутри уровня для каждой задачи задается свой уровень сложности в форме балла, описаны условия рейтинга:

- критерии оценки «3»: сумма баллов за все задачи 1-го уровня;
- критерии оценки «4»: нижняя граница для получения хорошей оценки будет находиться в интервале между верхним уровнем удовлетворительной оценки и нижним уровнем оценки «4»;
- критерии оценки «5»: нижняя граница баллов для получения отличной оценки – сумма баллов за все задачи 1-го и 2-го уровней + 1 задачу третьего.

В итоге получаем таблицу (Таблица 3):

*Таблица 3. Критерии оценивания*

Оценка	«2»	«3»	«4»	«5»
Сумма баллов	Менее 35	От 35 до 50	От 50 до 62	62 и выше

Самый обширный первый уровень состоит из 20 задач, сложность которых задана в диапазоне 1-3 балла. Задания первого уровня состоят в форме тестов, предполагающих воспроизведение известной информации об

операционных системах, системном обеспечении компьютера. Более сложные задания предполагают умение рассуждать. Включены задания, где нужно объяснить, для каких целей необходимо системное ПО, привести комплекс программ, обеспечивающих совместное функционирование всех устройств компьютера и предоставляющих пользователю доступ к его ресурсам.

Сложность второго уровня заданий варьируется от 4 до 6 баллов. Задания второго уровня предполагают более глубокие знания о программном обеспечении компьютера и предполагают умение сопоставлять, думать, размышлять. Учащиеся должны сопоставить предложенные программы с классом ПО, ответить на вопросы, такие как: в чем принципиальное отличие прикладного программного обеспечения от иных видов программного обеспечения?

Задания третьего уровня имеют практико-ориентированный характер. Для решения какой-то определенной задачи, необходимо подобрать конкретное программное обеспечение и составить краткое руководство по использованию этих программ. Здесь учащиеся смогут заработать от 7-8 баллов.

Все задания (Таблица 4) и критерии оценивания (Таблица 3) предоставляются ученикам в печатном виде в начале мега-урока.

*Таблица 4. Разноуровневые задания по изучению ПО*

<b>№</b>	<b>Задание</b>	<b>Вес</b>
<b>Задания первого уровня (1-3 балла)</b>		<b>36</b>
1.1	<i>Как называют совокупность всех программ, предназначенных для выполнения на компьютере?</i>	1
1.2	<i>Приведите классификацию программного обеспечения современных компьютеров.</i>	1
1.3	<i>Какие задачи выполняет прикладное программное обеспечение?</i>	2
1.5	<i>Какие разновидности программного обеспечения имеются на современных компьютерах?</i>	3
1.6	<i>Какие основные функции выполняет операционная система?</i>	2
1.7	<i>Для чего предназначены системы программирования? Кто с ними работает?</i>	2
1.8	<i>Какие программы управляют работой внешних устройств</i>	1

1.9	<i>Что такое система программирования?</i>			1
1.10	<i>Комплекс программ, обеспечивающих совместное функционирование всех устройств компьютера и предоставляющих пользователю доступ к его ресурсам?</i>			3
1.11	<i>Где хранится исполняемая в данный момент времени программа и данные, с которой она непосредственно работает?</i>			2
1.12	<i>Опишите последовательность команд, которую выполняет компьютер в процессе обработки данных</i>			3
1.13	<i>Совокупность программ, которые используют при разработке программ прикладного и системного ПО.</i>			2
1.14	<i>Для каких целей необходимо системное ПО?</i>			3
1.15	<i>Программа для обеспечения работы внешних устройств?</i>			2
1.16	<i>Для чего нужны инструментальные программы?</i>			2
1.17	<i>Программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений?</i>			1
1.18	<i>Программа, обеспечивающая пооператорный анализ, одновременную обработку и выполнение исходной программы или запроса?</i>			2
1.19	<i>Перечислите все, что находится на панели задач?</i>			3
1.20	<i>Приведите в пример как минимум две интегрированные среды программирования.</i>			2
<b>Задания второго уровня (4-6 баллов)</b>				<b>19</b>
2.1	<i>В чем заключается принцип организации диалога компьютер-пользователь с помощью меню?</i>			4
2.2	<i>Сравните, чего больше: А) Графических редакторов или прикладных программ? Б) Антивирусных программ или системных программ?</i>			5
2.3	<i>В чем принципиальное отличие прикладного программного обеспечения от иных видов программного обеспечения?</i>			4
2.4	<i>Определите, разновидностью системного или прикладного ПО является вид программного обеспечения:</i> <ul style="list-style-type: none"> <li>• Системы автоматизированного проектирования</li> <li>• Архиваторы</li> <li>• Бухгалтерские программы</li> <li>• Геоинформационные системы</li> <li>• Программы обслуживания дисков</li> <li>• Системы управления базами данных</li> <li>• Мультимедиа проигрыватели</li> <li>• Операционная система</li> <li>• Офисные пакеты</li> <li>• Электронные учебники</li> </ul> <i>Антивирусная программа</i>			6
<b>Задания третьего уровня (7-8)</b>				<b>15</b>
3.1	<i>Укажите, какое ПО необходимо в следующих ситуациях:</i>			7
	Ситуация	Системное ПО	Прикладное ПО	
	Ландшафтные дизайнеры создают проект нового городского ландшафта			

	Профессиональный программист пишет компьютерную программу по заказу крупной фирмы				
	Ученые научно-исследовательского института расшифровывают записи, переданные марсоходом				
3.2	<i>Определите ПО для данной операционной системы.</i>				8
	<b>Программное обеспечение</b>	<b>Пример</b>			
	Операционная система	MSWindows 7			
	Архиватор				
	Антивирусная программа				
	Коммуникационная программа				
	Система программирования				
	Текстовые редакторы				
	Графические редакторы				
	Редактор презентаций				
	Электронные таблицы				
	Электронное учебное издание				
	Игра				
	Редакторы работы со звуком				
	Редакторы работы с видеoinформацией				

Вторая часть основана на изучении темы «Алгоритм и его свойства» и закреплении первичных знаний об алгоритме. Задание представляет собой канонический тест из 20 вопросов, каждый вопрос оценивается в 1 балл. В каждом вопросе имеется 4 варианта ответа и содержится только один верный вариант:

Тест по теме «Алгоритм и его свойства»

- 1) алгоритм — это:
  - а) правила выполнения определенных действий;
  - б) ориентированный граф, указывающий порядок исполнения некоторого набора команд;

c) понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;

d) набор команд для компьютера;

2) укажите наиболее полный перечень способов записи алгоритмов:

a) словесный, графический, псевдокод, программный;

b) словесный;

c) графический, программный;

d) словесный, программный;

3) суть такого свойства алгоритма как результативность заключается в том, что ...?

a) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);

b) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;

c) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;

d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;

4) суть такого свойства алгоритма как массовость заключается в том, что:

a) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);

b) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;

c) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;

d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;

5) суть такого свойства алгоритма как дискретность заключается в том, что ...?

a) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);

b) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;

c) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;

d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;

6) суть такого свойства алгоритма как понятность заключается в том, что ...?

a) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);

b) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;

c) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;

d) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;

7) суть такого свойства алгоритма как детерминированность (точность) заключается в том, что ...?

a) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);

b) записывая алгоритм для конкретного исполнителя можно использовать лишь те команды, что входят в систему его команд;

c) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;

d) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма;

- 8) алгоритм называется линейным:
- a) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
  - b) если ход его выполнения зависит от истинности тех или иных условий;
  - c) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
  - d) если он представим в табличной форме;
- 9) алгоритм называется циклическим, если ...?
- a) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
  - b) если ход его выполнения зависит от истинности тех или иных условий;
  - c) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
  - d) если он представим в табличной форме;
- 10) алгоритм включает в себя ветвление, если...?
- a) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
  - b) если ход его выполнения зависит от истинности тех или иных условий;
  - c) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
  - d) если он представим в табличной форме;
- 11) какой тип алгоритма должен быть выбран при решении квадратного уравнения?
- a) линейный;
  - b) циклический;
  - c) разветвляющийся;
  - d) циклически-разветвляющийся;

- 12) разветвляющийся алгоритм – это ...?
- a) присутствие в алгоритме хотя бы одного условия;
  - b) набор команд, которые выполняются последовательно друг за другом;
  - c) многократное исполнение одних и тех же действий;
  - d) другое;
- 13) графическое задание алгоритма (блок/схемы) – это ...?
- a) способ представления алгоритма с помощью геометрических фигур;
  - b) представление алгоритма в форме таблиц и расчетных формул;
  - c) система обозначения правил для единообразной и точной записи алгоритмов их исполнения;
  - d) схематическое изображение в произвольной форме;
- 14) исполнитель алгоритмов – это:
- a) человек или автомат (в частности компьютер), умеющий выполнять некоторый, вполне определенный набор действий
  - b) понятное и точное предписание;
  - c) связи между этапами при помощи стрелок;
  - d) определенные условия;
- 15) какой из документов является алгоритмом?
- a) правила техники безопасности;
  - b) инструкция по приготовлению пищи;
  - c) расписание движения поездов;
  - d) список книг в школьной библиотеке;
- 16) какой вид алгоритма используется для вычисления площади треугольника по трем сторонам?
- a) линейный;
  - b) циклический;
  - c) разветвляющийся;
  - d) любой;

- 17) отдельное указание исполнителю - это...?
- a) программа;
  - b) алгоритм;
  - c) команда;
  - d) исполнитель;
- 18) форма организации действий, при которой один и тот же блок команд выполняется несколько раз, называется...?
- a) следованием;
  - b) циклом;
  - c) ветвлением;
  - d) алгоритмом;
- 19) совокупность всех команд, которые может выполнить конкретный исполнитель - это...?
- a) система программ;
  - b) система алгоритмов;
  - c) система команд;
  - d) система задач;
- 20) в расчете на кого должен строиться алгоритм?
- a) в расчете на ЭВМ;
  - b) в расчете на умственные способности товарища;
  - c) в расчете на конкретного исполнителя;
  - d) на всех одновременно.

Ответы на тест приведены ниже (см.Таблицу 5).

*Таблица 5. Ответы к тесту по теме «Алгоритм и его свойства»*

№ вопроса	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ответ	с	а	d	с	а	b	e	с	а	b	с	а	а	а	b	а	с	b	с	с

Далее представлена таблица с критериями оценивания теста (см. Таблицу 6)

Таблица 6. Критерии оценивания теста

Оценка	«2»	«3»	«4»	«5»
Сумма баллов	Менее 10	От 10 до 15	От 16 до 18	19 и выше

Тест предоставляется учащимся в виде ссылки на гугл-диске после самостоятельного изучения темы «Алгоритм и его свойства» [32,34].

В каждой школе дети разбиваются на группы по 2-3 человека. Из них необходимо организовать межшкольные группы. Желательно за несколько дней до проведения мега-урока, предоставить группам возможность познакомиться друг с другом в социальных сетях.

Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для данных групп тьюторами создаются интерактивные доски в online приложении Linoit.com. Доска разбита на три части (область работы над задачами, область – вопросы тьютору, область для общения членов группы). Определившись с задачей, в соответствующую область доски необходимо вывесить стикер с ее номером в поле для работы с задачей. Завершив работу над задачей и представив ответ, цвет стикера изменяется на красный (это будет сигналом тьютору для начала проверки задачи). Вопросы тьютору и общение членов группы осуществляются строго на своём поле. Оценка тьютора будет вывешена на отдельном стикере, поверх стикера с задачей.

Каждая группа учащихся работает в первой половине урока с печатным вариантом вопросов. Они имеют возможность выбирать сложность задачи, при решении которой могут использовать уже имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом.

Во второй половине занятия мега-учитель с помощью презентации на гугл-диске объясняет учащимся новый материал по теме «Алгоритм и его свойства». Затем учащимся предстоит самостоятельное, более подробное

изучение материала. Использовать для этого ученики могут имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала тьютор выкладывает на доску link ссылку на тест для закрепления и систематизации полученных знаний. При выполнении теста на гугл-диске учащиеся не могут обращаться за помощью к тьютору. Тьютор следит за выполнением задания, проверяет ответы.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в режиме просмотра. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп (Рис.16). Данный рейтинг позволяет учащимся на протяжении всего урока отслеживать успехи своей группы.

B4		fx =СУММ(D4:BP4)									
	A	B	C	D	E	F	G	H	I	J	K
1	Номер группы	Всего баллов	Номер задания	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8
2			Баллы	1	1	2	2	3	2	2	2
3	1	4		1	1	2					
4	2	0									
5	3	0									
6	4	3						3			
7	5	0									
8	6	0									
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											

Рис.16. Рейтинг групп

Главная особенность урока состоит в необычной форме дистанционного взаимодействия, где присутствуют различные сетевые способы взаимодействия и способы управления деятельностью учащихся: учитель организует группы, управляет деятельностью учащихся на уроке, студенты работают как эксперты и тьюторы, ученики - работа в разных группах (межшкольные группы, взаимодействие через интерактивные доски).

Следующей особенностью является выступление педагога из вуза (модератора) в конце мега-урока, в котором анализируются информационные процессы, происходящие в компьютере при решении конкретных информационных задач.

В это время эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. На этапе подведения итогов предполагается выступление тьюторов: их выводы об отрицательных и положительных моментах взаимодействия с учащимися, предварительные результаты выполнения заданий, демонстрация рейтинговой таблицы в Google. В конце занятия мега-учитель предлагает в качестве домашнего задания найти информацию об истории возникновения языка программирования Python.

Данный урок поможет обучающимся закрепить и углубить знания по теме «Программное обеспечение компьютера», а именно

- знание о назначении системного ПО, назначении ОС и программ, которые относятся к системному ПО, о назначении программ, которые относятся к прикладному ПО, а также систем программирования.
- понятие и термины: программное обеспечение (ПО), операционная система, программа.

А также развить некоторые умения связанные с этой темой:

- определять, к какому программному обеспечению относится конкретная программа;
- классифицировать ПО на прикладные, системные и системы программирования.

Также данное занятие поможет учащимся постепенно перейти к новому разделу «Алгоритмизация и программирование», изучить новую тему «Алгоритм и его свойства», проверить усвоение новых знаний об алгоритмах построения программы, их свойств.

### Сценарий мега-урока «Простейшие программы. Вычисления»

Мега-урок основан на изучении простейших программ и вычислений с помощью языка программирования Python.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.17).

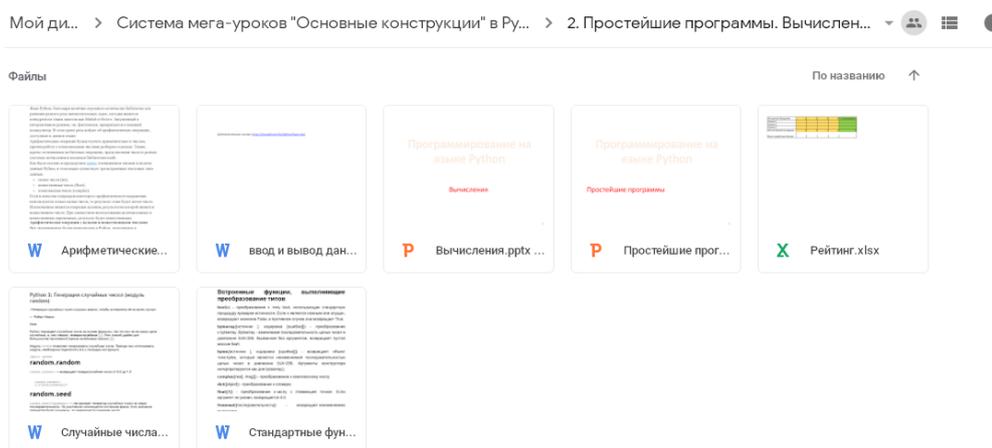


Рис.17. Теоретический материал к уроку

Подробнее можно посмотреть по ссылке:

[\[https://drive.google.com/open?id=1UUwDAVA4k0pBvDanmMdThOUF-](https://drive.google.com/open?id=1UUwDAVA4k0pBvDanmMdThOUF-r6V2ssr)

[r6V2ssr\]](https://drive.google.com/open?id=1UUwDAVA4k0pBvDanmMdThOUF-r6V2ssr). Теоретический материал состоит из семи документов в форматах docx и pptx, а именно:

- 1) 2 документа в формате pptx: «Вычисления», в котором имеется краткая теория об основных вычислениях в Питоне, а также «Простейшие программы», где представлено описание и примеры простейших программ на Питон;
- 2) «Ввод и вывод» - даны дополнительные ссылки на углубленное изучение материала;
- 3) «Арифметические выражения и операции» - представлено описание основных арифметических операций в Питон, а также дополнительные ссылки на подробное изучение подтемы;

- 4) «Случайные числа», где дано подробное описание модуля random и примеры программ, а также имеются дополнительные ссылки по данной подтеме;
- 5) «Стандартные функции» - имеется описание основных стандартных функций в Питон и дополнительные ссылки;
- 6) «Рейтинг» - рейтинговая таблица с баллами для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в режиме просмотра. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп (рис.18). Данный рейтинг позволяет учащимся на протяжении всего урока отслеживать успехи своей группы.

	A	B	C	D	E	F
1	Не группы/ Незадания	1	2	3	4	4.2 (программа)
2	Группа 1					
3	Группа 2					
4	Группа 3					
5	Кол-во баллов за задание	1	1	1	2	5
6	Всего заработано баллов					
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

Рис18. Рейтинг групп

Важное значение имеет содержание задач при их решении для повышения мотивации познавательной деятельности учащихся. Задания должны быть доступными и интересными. Данные требования и стали определяющими содержания заданий в системе задач для мега-урока. Система задач включает в себя 2 части и дополнительное задание

повышенного уровня сложности. Все задания представлены ниже:

*Задания на изучение темы «Простейшие программы. Вычисления»*

Часть 1. Задания на составление простейших программ.

Группа №1:

1. Что будет выведено на экран в результате выполнения фрагмента программы

$a = 5$

$b = 3$

```
print( a, "Z(", b, ")", sep="" )
```

Ответ:

введите строку.

2. Запишите оператор для вывода значений переменных  $a=5$  и  $b=3$  в следующем формате:

$Z( 5 )=F( 3 )$

В ответе не используйте лишние пробелы.

3. Запишите оператор для вывода значений переменных  $a=6$  и  $b=7$  в следующем формате:

$6 + 7 = ?$

4. Создать программу, которая будет спрашивать у пользователя его имя и возраст. Вывести на экран сообщение, в котором говорится, сколько лет ему было бы через век.

Группа №2:

1. Что будет выведено на экран в результате выполнения фрагмента программы

$a = 3$

$b = 13$

```
print( 'F( ', a, ' )=( ', a+b, ' )'
```

2. Запишите оператор для вывода значений переменных  $a=50$  и  $b=13$  в следующем формате:

$50 + 13 = ?$

В ответе не используйте пробелы.

Ответ:

введите строку

3. Запишите оператор для вывода значений переменных  $a=23$  и  $b=21$  в следующем формате:

$23*21=?$

4. Запросите у пользователя четыре числа. Отдельно сложите первые два и отдельно вторые два. Разделите первую сумму на вторую. Выведите результат на экран так, чтобы ответ содержал две цифры после запятой.

Группа №3:

1. Что будет выведено на экран в результате выполнения фрагмента программы

$a = 19$

$b = 2$

`print( b, "Z(", a, ")", sep="" )`

Ответ:

введите строку

2. Запишите оператор для вывода значений переменных  $a=43$  и  $b=33$  в следующем формате:

$a= 43 ;b= 33$

3. Запишите оператор для вывода значений переменных  $a=2$  и  $b=5$  в следующем формате:

$2^5=?$

4. Напишите программу, которая предлагала бы пользователю решить пример  $4 * 100 - 54$ . Потом выводила бы на экран правильный ответ и ответ пользователя. Подумайте, нужно ли здесь преобразовывать строку в число.

Часть 2. Вычисления

Группа №1 (Арифметические операции и выражения):

Написать программу, которая выполняет различные (по вариантам) математические действия над двумя числами, выводя на экран развёрнутую

форму записи действия.

*На входе:* два целых положительных числа.

*На выходе:* строка, содержащая целые (или вещественные) числа.

- 1) сумма двух чисел;
- 2) возведение в куб разности двух чисел;
- 3) произведение двух чисел;
- 4) возведение числа в степень;
- 5) возведение в квадрат суммы двух чисел;
- 6) возведение в квадрат разности двух чисел;
- 7) возведение в квадрат произведения двух чисел;
- 8) разность двух чисел;
- 9) возведение в квадрат частного двух чисел;
- 10) возведение в куб суммы двух чисел;
- 11) возведение в куб частного двух чисел;
- 12) возведение в куб произведения двух чисел;
- 13) частное двух чисел (без учёта деления на ноль);
- 14) экспонента числа.

Группа №2 (Стандартные функции):

С начала суток прошло  $N$  часов,  $M$  минут,  $S$  секунд ( $0 \leq N < 12$ ,  $0 \leq M < 60$ ,  $0 \leq S < 60$ ). По данным числам  $N$ ,  $M$ ,  $S$  определите угол (в градусах), на который повернулась часовая стрелка с начала суток и выведите его в виде действительного числа.

Составьте программу для решения данной задачи.

Группа №3 (Случайные числа):

Получить случайное трехзначное число и вывести через запятую его отдельные цифры.

Составьте программу для решения данной задачи.

Часть 3. Задача на закрепление.

Сила сопротивления воздуха движению любого объекта может быть описана так:

$$F_d = \frac{1}{2} C_D \rho A V^2$$

В этом выражении:

$\rho$  - плотность воздуха

$V$  — скорость объекта

$A$  — его поперечная площадь (для мяча находится как  $\pi r^2$ )

$C_D$  — коэффициент, зависящий от формы объекта и неровности его поверхности.

Напишите программу, которая рассчитывает силу сопротивления и силу тяжести и выводит:

а) их значения с одинаковой точностью в одних и тех же единицах (Ньютонах);

б) отношение этих значений. В тексте программы представить изменяющиеся величины как переменные, сопровождаемые комментариями.

Для мяча принять:  $\rho = 1.2$  кг/м<sup>3</sup>, масса мяча 0.43 кг,  $C_D = 0.2$ .

Сравните результаты при сильном сопротивлении, когда скорость мяча достигает 120 км/ч и слабом - с  $V = 10$  км/ч. Не забудьте, что в физике обычно используют одну систему, СИ.

Первая часть посвящена составлению простейших программ на Python, включает 4 задания с двумя уровнями сложности [25,27].

Первый уровень состоит из 3 заданий, сложность которых оценивается в 1 балл. Задания первого уровня состоят в форме задач с готовыми программами, предполагающих воспроизведение известной информации о вводе и выводе данных, типах данных, значениях переменных.

Сложность второго уровня оценивается в 2 балла. Второй уровень включает всего 1 задание, которое предполагает более глубокие знания о составлении простейших программ и предполагают умение думать и составлять. Учащиеся должны написать программу по предложенной задаче.

Вторая часть предполагает использование полученных знаний по

предложенному для каждой группы пункту изучаемой темы при разработке программы по задаче. Задание для каждой группы соответствует изучаемому материалу. Правильное составление программы предполагает оценку в 5 баллов.

Последнее задание представлено в форме сложной задачи, для решения которой необходимы знания по всему пройденному материалу, а также дополнительно знания физики.

Все задания первой и второй части предоставляются ученикам на гугл-диске в виде рабочей тетради. Последнее задание представлено в виде ссылки на отдельный документ в рабочей тетради. Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в с). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к рабочей тетради.

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к своему документу.

В каждой школе дети разбиваются на 3 группы. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с учащимися в рабочей тетради имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению той или иной задачи, а также давать сигнал на проверку выполненного задания.

Перед изучением нового материала мега-учитель проводит актуализацию знаний обучающихся в форме фронтального опроса, вопросы которого можно посмотреть ниже.

*Вопросы фронтального опроса по теме «Алгоритм и его свойства»:*

- 1) дайте определение алгоритма;

- 2) дайте определение блок-схемы;
- 3) перечислите способы описания алгоритмов;
- 4) дайте определение линейного алгоритма;
- 5) перечислите оператора вывода и ввода.

После актуализации опорных знаний каждая группа учащихся работает с первой частью заданий в своей рабочей тетради на гугл-диске. Они имеют возможность выбирать сложность задачи, при решении которой могут использовать уже имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет ответы и выставляет баллы в рейтинг.

Вторая половина занятия основана на нелинейном подходе к обучению. Мега-учитель с помощью презентации на гугл-диске объясняет учащимся новый материал по теме «Вычисления». Затем учащимся предстоит самостоятельное, более детальное изучение материала: группе №1 предлагается подробное изучение пункта «Арифметические операции и выражения», группе №2 – «Стандартные функции», группе №3 – «Случайные числа». Использовать для этого ученики могут имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению второй части заданий – составлению программ, для закрепления и систематизации полученных знаний. Тьютор следит за временем выполнения задания, проверяет готовую программу и выставляет баллы в рейтинговую таблицу.

По завершении второй части заданий группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного пункта по теме «Простейшие программы и вычисления» (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала).

Для закрепления полученных знаний по всей пройденной теме урока учащимся предоставляется более сложное задание в отдельном документе, ссылка на который представлена в рабочих тетрадях. Задача связана с физикой и выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 группа занимается физическими расчетами, 3 группа переводит физические расчеты в машинный язык. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа.

Особенностью урока также является необычная форма дистанционного взаимодействия, где присутствуют различные сетевые способы взаимодействия и способы управления деятельностью учащихся: учитель организывает группы, управляет деятельностью учащихся на уроке, студенты работают как эксперты и тьюторы, ученики - работа в разных группах (межшкольные группы, взаимодействие через гугл-документы).

Также особенностью является и выступление педагога из вуза (модератора) в конце мега-урока, который в необычной форме анализирует межгрупповые взаимодействия при выполнении последней задачи.

В это время эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. На этапе подведения итогов предполагается выступление тьюторов: их выводы об отрицательных и положительных моментах взаимодействия с учащимися, предварительные результаты выполнения заданий, демонстрация рейтинговой таблицы в Google. В конце занятия мега-учитель предлагает в качестве домашнего задания решение задач с использованием арифметических операций. Текст задания представлен ниже:

Домашнее задание по теме «Простейшие программы. Вычисления»:

1. Напишите программу, которая моделирует бросание двух игральных кубиков: при запуске выводит случайное число в диапазоне от 2 до 12.

2. Напишите программу, которая случайным образом выбирает дежурных: выводит два различных случайных числа в диапазоне от 1 до  $N$ , где  $N$  – количество учеников вашего класса.

С какой проблемой вы можете столкнуться?

3. Напишите программу, которая вводит два вещественных числа,  $a$  и  $b$  ( $a < b$ ), и выводит на экран 5 случайных вещественных чисел в полуинтервале  $[a, b)$ .

Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Простейшие программы и вычисления, а именно:

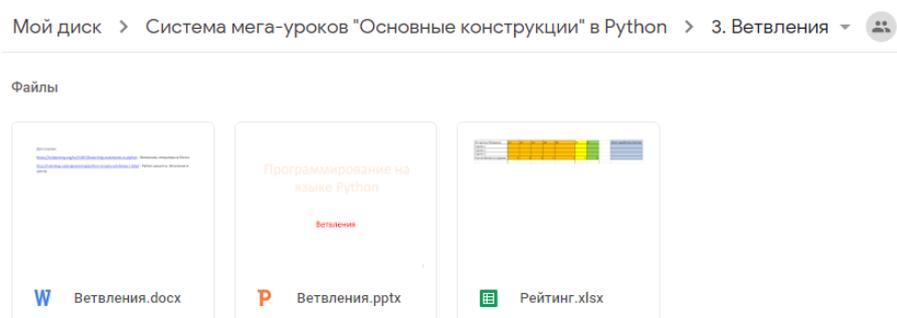
1. Знание основных операторов в Питоне, типов переменных, типов данных, арифметических выражений и операций.

2. Формирование умений применять различные арифметические операции и выражения, а также стандартные функции и случайные числа при составлении программ.

### Сценарий мега-урока «Ветвления»

Данный мега-урок посвящен изучению оператора ветвления на языке Python.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.19), который подробнее можно посмотреть по ссылке: [https://drive.google.com/open?id=1RuBS9bkYiddADTUf67fZem8GTFDTAoXH ]



*Рис.19. Теоретический материал к уроку*

Материал состоит из трех документов в форматах docx, pptx и xlsx, а именно:

- 1) презентация «Ветвления», в которой имеется описание основных конструкций ветвления в Питон, а также представлены различные примеры программ с их применением;
- 2) «Ветвления», где представлены ссылки на самостоятельное изучение материала;
- 3) «Рейтинг» - рейтинговая таблица с баллами для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в режиме просмотра. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп (Рис.20). Данный рейтинг позволяет учащимся на протяжении всего урока отслеживать успехи своей группы.

№ группы/ №задания	A1	A2	A3	A4	A5	B	C	Всего заработано баллов
Группа 1								
Группа 2								
Группа 3								
Кол-во баллов за задание	1	1	1	1	1	1	3	5

Рис.20. Рейтинг групп

Урок состоит из трех этапов: актуализации знаний, теоретико-практической части и закрепления.

Теоретико-практическая часть предоставляется группам в виде рабочей тетради на гугл-диске. Задания включают 3 уровня сложности (смотрите ниже).

*Задания на изучение темы «Ветвления»:*

- А. 1) как называются алгоритмы, содержащие ветвления?
- 2) какое значение, должна иметь переменная s, чтобы в ветвлении с условием  $s \neq 0$  выполнен блок else?
- 3) какой блок ветвления в языке Python реализует разделение кода более, чем на две ветви?

4) определите значение переменной А после выполнения фрагмента программы:

```
a = 11;
  if a > 5: a += 13
  else: a -= 9
  if a > 5: a += 13
  else: a -= 9
```

5) определите значение переменной А после выполнения фрагмента программы:

```
a = 11
b = 4
  if a > b: b += 13
  else: a -= 8
  if a > b: a += 13
  else: a -= 8
```

В.Задача: из двух случайных чисел, одно из которых четное, а другое нечетное, определить и вывести на экран нечетное число.

Необходимо исправить ошибки в программе:

```
from random import random
a = float(random() * 100)
b = bool(random() + 100)
  if a%2 and b%2 and a%2==0 and b%2==0:
a += 1
    print(a,b)
  else a%2:
    print(a)
  else:
    print(b)
```

С. Напишите условие, которое определяет заштрихованную область (рис.21).

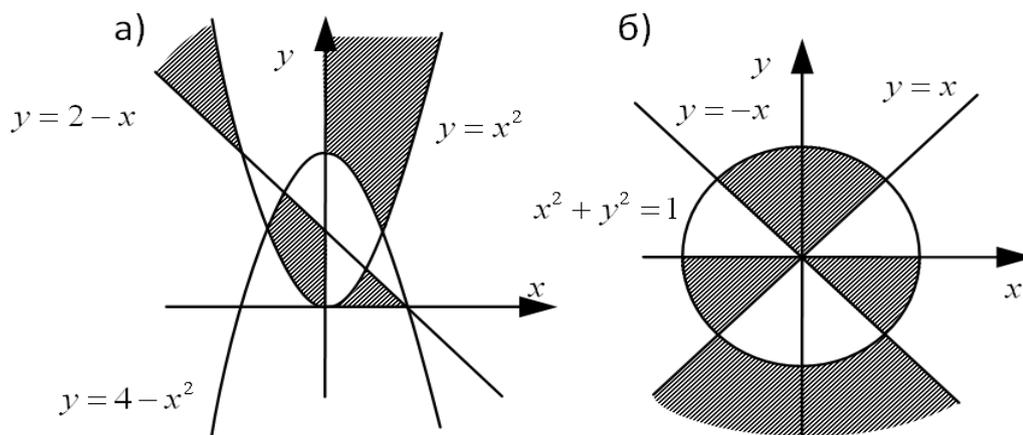


Рис.21.

Задания первого уровня включают в себя 5 вопросов на понимание изученной подтемы. Каждый вопрос оценивается в 1 балл.

Второй уровень включает в себя одну программу с ошибками, которые необходимо исправить. Данное задание направлено на практическое применение полученных знаний и оценивается в 3 балла.

Задание третьего уровня представлено в форме задачи, по которой нужно составить программу. Данное задание также способствует формированию навыков и умений применять полученные знания в практической деятельности. Задача оценивается в 5 баллов.

Этап закрепления знаний составляет одна задача высокого уровня сложности. Данное задание способствует повышению познавательных и творческих способностей учащихся, развивает логическое мышление.

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к рабочей тетради.

В каждой школе дети разбиваются на 3 группы. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с

учащимися в рабочей тетради имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению той или иной задачи, а также давать сигнал на проверку выполненного задания.

В начале занятия мега-учитель проводит фронтальный опрос по изученной теме на прошлом уроке (вопросы представлены ниже), после чего он проверяет выполнение домашнего задания: один ученик рассказывает об истории возникновения языка программирования Python (возможно наглядное представление материала в виде презентации), остальные учащиеся слушают и дополняют доклад.

*Вопросы фронтального опроса по теме «Простейшие программы. Вычисления»:*

1. Какие типы данных вы знаете?
2. Какие данные записываются в логические переменные?
3. Расскажите об особенностях переменных в языке Python. Почему может получиться, что изменение одной переменной автоматически приводит к изменению другой?
4. Что такое приоритет операций? Зачем он нужен?
5. В каком порядке выполняются операции, если они имеют одинаковый приоритет?
6. Зачем используются скобки?
7. Что происходит, если в выражения входят переменные разных числовых типов? Какого типа будет результат?
8. Опишите операции // и %.
9. Расскажите о проблеме вычисления остатка от деления в различных языках программирования. Обсудите в классе этот вопрос.
10. Какие стандартные математические функции вы знаете? В каких единицах задается аргумент тригонометрических функций?
11. Как выполнить округление вещественного числа к ближайшему целому?
12. Какие числа называют случайными? Зачем они нужны?
13. Как получить «естественное» случайное число? Почему такие числа

почти не используются в цифровой технике?

По завершении этапа актуализации и проверки домашнего задания мега-учитель объявляет новую тему урока и излагает основные тезисы по данной теме с помощью презентации на гугл-диске.

Далее мега-урок проходит на основе нелинейного подхода. Учащимся предлагается решить следующую задачу:

Задача: есть две коробки, первая размером  $A1 \times B1 \times C1$ , вторая размером  $A2 \times B2 \times C2$ . Определите, можно ли разместить одну из этих коробок внутри другой, при условии, что поворачивать коробки можно только на 90 градусов вокруг ребер. Формат входных данных:

Программа получает на вход числа  $A1, B1, C1, A2, B2, C2$ . Формат выходных данных:

Программа должна вывести одну из следующих строчек:

- boxes are equal – если коробки одинаковые;
- the first box is smaller than the second one – если первая коробка может быть положена во вторую;
- the first box is larger than the second one – если вторая коробка может быть положена в первую;
- boxes are incomparable, во всех остальных случаях.

Для решения данной задачи обучающимся предлагается решить более простые задания, изучив материал на гугл-диске. Также можно использовать электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в рабочей тетради для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет ответы и выставляет баллы в рейтинг. Команда, быстрее всех выполнившая все задания в рабочей тетради, получает дополнительные баллы в рейтинговую таблицу.

Учащиеся должны убедиться в правильности разработки программ для

того, чтобы наглядно продемонстрировать готовую программу и прокомментировать каждый шаг. Далее группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного материала (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала). Мега-учитель, при необходимости, комментирует работы групп и указывает на недочеты.

Для закрепления полученных знаний по всей пройденной теме урока учащиеся возвращаются к заданию, представленному в начале урока. Задание находится в отдельном документе, ссылка на который имеется в рабочих тетрадях:

Задача выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 группа занимается составлением блок-схемы решения задачи, 3 группа работает над составлением множественных ветвлений. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа [4].

К концу занятия традиционно выступает педагог из вуза (модератор), который в необычной форме анализирует межгрупповые взаимодействия при выполнении последней задачи, а также делает комментарии и поправки в самом коде программы.

В это время эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. На этапе подведения итогов предполагается выступление тьюторов: их выводы об отрицательных и положительных моментах взаимодействия с учащимися, предварительные результаты выполнения заданий, демонстрация рейтинговой таблицы в Google. Группа, набравшая наибольшее количество баллов за все прошлые уроки и данный урок, получает оценки «отлично». В конце занятия мега-учитель предлагает в качестве домашнего задания решение трех задач с

использованием ветвлений и циклов:

Домашнее задание по теме «Ветвления»

1) даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень – отрицательны;

2) даны две точки  $A(x_1, y_1)$  и  $B(x_2, y_2)$ . Составить программу, определяющую, которая из точек находится ближе к началу координат.

Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Ветвления», а именно:

1. Знания основных понятий: ветвления, каскадные ветвления, тернарная операция.

2. Формирование умений применять полученные знания о ветвлениях языка программирования Питон в практической деятельности и при решении творческих задач.

### Сценарий мега-урока «Циклы»

Данный мега-урок посвящен изучению циклов на языке Python.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.22), который подробнее можно посмотреть по ссылке: [\[https://drive.google.com/open?id=1b6F7ixamxMDNvMv2je9pHSGDL-9ZzrfN\]](https://drive.google.com/open?id=1b6F7ixamxMDNvMv2je9pHSGDL-9ZzrfN)

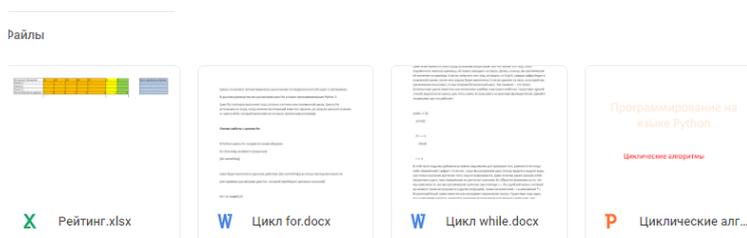


Рис.22. Теоретический материал к уроку

Материал состоит из четырех документов в форматах docx и pptx, а именно:

- 1) презентация: «Циклические алгоритмы», в которой имеется описание основных ветвлений в Питон, а также представлены различные примеры программ с их применением;
- 2) «Цикл for», «Цикл while», в которых описано применение данных циклов в Питон, а также имеются наглядные примеры и дополнительные ссылки;
- 3) «Рейтинг» - рейтинговая таблица с баллами для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в режиме просмотра. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп (Рис.23). Данный рейтинг

позволяет учащимся на протяжении всего урока отслеживать успехи своей группы.

	A	B	C	D	E	F	G	H	I	J
1	№ группы/ Незадания	A1	A2	A3	A4	A5	B	C		Всего заработано баллов
2	Группа 1									
3	Группа 2									
4	Группа 3									
5	Кол-во баллов за задание	1	1	1	1	1	3	5		
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										

Рис.23. Рейтинг групп

Урок состоит из трех этапов: актуализации знаний, теоретико-практической части и закрепления.

Теоретико-практическая часть предоставляется группам в виде рабочей тетради на гугл-диске. Задания включают 3 уровня сложности (смотрите ниже).

### *Задания на изучение темы «Циклы»*

#### 1.Задания для 1 межсетевой группы (цикл for):

A. 1) для чего в Питоне предназначен цикл for?

2) алгоритм, содержащий циклы, называется циклическим алгоритмом или ...?

3) в цикле с параметром количество повторений ("оборотов") цикла заранее известно?

4) определите, что будет напечатано в результате выполнения следующего скрипта:

```
s = 0
```

```
for k in range(3,11):
```

```
    s = s + k
```

```
print(s)
```

5) определите, что будет напечатано в результате выполнения

следующего скрипта:

```
s = 0
for k in range(-5,11):
    s = s + 2 * k
print(s)
```

В. Задача: Поиск квадратных уравнений, имеющих решение.

Программа принимает от пользователя диапазоны для коэффициентов a, b, c квадратного уравнения:

$$ax^2 + bx + c = 0$$

Перебирает все варианты целочисленных коэффициентов в указанных диапазонах, определяет квадратные уравнения, которые имеют решение.

Необходимо исправить ошибки в программе:

```
import math
a1 = int(input('a1: '))
a2 = float(input('a2: '))
b1 = float (input('b1: '))
b2 = float (input('b2: '))
c1 = float (input('c1: '))
c2 = float (input('c2: '))
a = bool(a1, a2 + 1)
b = range(b1, b2 + 1)
c = bool (c1, c2 + 1)
i in a:
if i == 0:
    continue
for j in b:
    k in c:
print(i, j, k, end='')
D = j * j - 4 * i * k
if D >= 0:
```

```

x1 = (-j - math.sqrt(D)) / (2 * i)
x2 = (-j + math.sqrt(D)) / (2 * i)
input('Yes', round(x1, 2), round(x2, 2))
else:
print('No')

```

С. Задача: Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Найдите ее, зная номера оставшихся карточек.

Дано число N, далее N – 1 номер оставшихся карточек (различные числа от 1 до N). Программа должна вывести номер потерянной карточки.

Массивами и аналогичными структурами данных пользоваться нельзя.

2. Задания для 2 межсетевой группы (цикл while):

A. 1) для чего в Питоне предназначен цикл while?

2) алгоритм, содержащий циклы, называется циклическим алгоритмом или ...?

3) в цикле с условием при каждом следующем повторении что требуется делать?

4) определите, что будет напечатано в результате выполнения следующего скрипта:

```

s = 0
m = 123
while m > 0:
    d = m % 10
    s = s + d
    m = m // 10
print(s)

```

5) Определите, что будет напечатано в результате выполнения следующего скрипта:

```

c = 0

```

```

m = 123
while m > 1:
    d = m % 10
    c = (c + d) * 10
    m = m // 10
print(c)

```

В. Задача: Перевод десятичного числа в любую систему счисления с основанием до 9.

Необходимо исправить ошибки в программе:

```

num = float(input())
base = int(print("Base (2-9): "))
if (2 <= base <= 10):
    quit()
newNum = ""
while num < 0:
    newNum = int(num % base) + newNum
input(newNum)
num //= base

```

С. Задача: Последовательность состоит из натуральных чисел и завершается числом 0. Определите индекс наибольшего элемента последовательности. Если наибольших элементов несколько, выведите индекс первого из них. Нумерация элементов начинается с нуля.

Задания первого уровня включают в себя 5 вопросов на понимание изученной подтемы. Каждый вопрос оценивается в 1 балл.

Второй уровень включает в себя одну программу с ошибками, которые необходимо исправить. Данное задание направлено на практическое применение полученных знаний и оценивается в 3 балла.

Задание третьего уровня представлено в форме задачи, по которой нужно составить программу. Данное задание также способствует формированию навыков и умений применять полученные знания в

практической деятельности. Задача оценивается в 5 баллов.

Этап закрепления знаний составляет одна задача высокого уровня сложности. Данное задание способствует повышению познавательных и творческих способностей учащихся, развивает логическое мышление.

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к рабочей тетради.

В каждой школе дети разбиваются на 2 группы. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с учащимися в рабочей тетради имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению той или иной задачи, а также давать сигнал на проверку выполненного задания.

В начале занятия мега-учитель проводит фронтальный опрос по изученной теме на прошлом уроке (вопросы представлены ниже), после чего он проверяет выполнение домашнего задания: один ученик рассказывает об истории возникновения языка программирования Python (возможно наглядное представление материала в виде презентации), остальные учащиеся слушают и дополняют доклад.

*Вопросы фронтального опроса по теме «Ветвления»:*

1. Чем отличаются разветвляющиеся алгоритмы от линейных?
2. Как вы думаете, почему не все задачи можно решить с помощью линейных алгоритмов?
3. Как вы думаете, хватит ли линейных алгоритмов и ветвлений для разработки любой программы?
4. Почему нельзя выполнить обмен значений двух переменных в два шага:  
 $a=b; b=a$ ?
5. Чем отличаются условные операторы в полной и неполной формах? Как

вы думаете, можно ли обойтись только неполной формой?

6. Какие отношения вы знаете? Как обозначаются отношения «равно» и «не равно»?

По завершении этапа актуализации и проверки домашнего задания мега-учитель объявляет новую тему урока и излагает основные тезисы по данной теме с помощью презентации на гугл-диске.

Далее мега-урок проходит на основе нелинейного подхода. Учащимся предстоит самостоятельное, более детальное изучение материала: группе №1 предлагается подробное изучение пункта «Цикл for», группе №2 – «Цикл while». Использовать для этого ученики могут имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в рабочей тетради для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет ответы и выставляет баллы в рейтинг.

Учащиеся должны убедиться в правильности разработки программ по своим для того, чтобы наглядно продемонстрировать готовую программу и объяснить изученный материал другим группам. Далее группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного пункта по теме «Циклы» (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала).

Для закрепления полученных знаний по всей пройденной теме урока учащимся предоставляется более сложное задание в отдельном документе, ссылка на который представлена в рабочих тетрадях:

*Задача на закрепление (для всех групп):*

В магазине продается мастика в ящиках по 15 кг, 17 кг, 21 кг. Как купить ровно 185 кг мастики, не вскрывая ящики? Сколькими способами

можно это сделать?

Задача выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 группа занимается составлением ветвлений, 3 группа занимается составлением циклов. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа [4].

К концу занятия традиционно выступает педагог из вуза (модератор), который в необычной форме анализирует межгрупповые взаимодействия при выполнении последней задачи, а также делает комментарии и поправки в самом коде программы.

В это время эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. На этапе подведения итогов предполагается выступление тьюторов: их выводы об отрицательных и положительных моментах взаимодействия с учащимися, предварительные результаты выполнения заданий, демонстрация рейтинговой таблицы в Google. Группа, набравшая наибольшее количество баллов за все прошлые уроки и данный урок, получает оценки «отлично». В конце занятия мега-учитель предлагает в качестве домашнего задания решение трех задач с использованием ветвлений и циклов:

Домашнее задание по теме «Циклы»:

1. Алгоритм Евклида для вычисления наибольшего общего делителя двух натуральных чисел, формулируется так: нужно заменять большее число на разность большего и меньшего до тех пор, пока одно из них не станет равно нулю; тогда второе и есть НОД. Напишите программу, которая реализует этот алгоритм. Какой цикл тут нужно использовать?

2. Напишите программу, использующую модифицированный алгоритм Евклида: нужно заменять большее число на остаток от деления

большее на меньшее до тех пор, пока этот остаток не станет равен нулю; тогда второе и есть НОД.

3. Добавьте в решение двух предыдущих задач вычисление количества шагов цикла. Заполните таблицу (Таблица 7) (шаги-1 и шаги-2 означают количество шагов двух версия алгоритма Евклида) [12]:

*Таблица 7. Таблица для заполнения шагов цикла*

a	64168	358853	6365133	17905514	549868978
b	82678	691042	11494962	23108855	298294835
НОД(a,b)					
Шаги-1					
Шаги-2					

Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Циклы», а именно:

3. Знания основных понятий: циклы, циклы с условием и постусловием, циклы с переменной.

4. Формирование умений применять полученные знания о ветвлениях и циклах языка программирования Питон в практической деятельности и при решении творческих задач.

### *Сценарий мега-урока «Вложенные циклы»*

Данный мега-урок посвящен изучению вложенных циклов на языке Python и закреплению изученного материала прошлого урока.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.24), который подробнее можно посмотреть по ссылке: [\[https://drive.google.com/open?id=1b6F7ixamxMDNvMv2je9pHSGDL-9ZzrfN\]](https://drive.google.com/open?id=1b6F7ixamxMDNvMv2je9pHSGDL-9ZzrfN)



*Рис.24. Теоретический материал к уроку*

Материал состоит из трех документов в форматах docx и pptx, а именно:

- 1) презентация: «Вложенные циклы», в которой имеется описание вложенных циклов в Питон, а также представлены различные примеры программ с их применением;
- 2) «Вложенные циклы», где представлены ссылки на самостоятельное изучение материала;
- 3) «Рейтинг» - рейтинговая таблица с баллами для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в

режиме просмотра. В этой таблице по мере выполнения заданий группами, эксперты обновляют текущий рейтинг групп (Рис.25). Данный рейтинг позволяет учащимся на протяжении всего урока отслеживать успехи своей группы.

	A	B	C	D	E	F	G	H	I	J
1	№ группы	Задача 1	Доп.баллы	Всего баллов за урок						
2	Группа 1									
3	Группа 2									
4										
5	макс. Кол-во баллов	3	5	8						
6										
7										
8										
9										
10										
11										

Рис.25. Рейтинг групп

Урок состоит из трех этапов: актуализации знаний, теоретико-практической части и закрепления.

Теоретико-практическая часть предоставляется группам в виде рабочей тетради на гугл-диске. Задания включают 3 уровня сложности (смотрите ниже).

*Задания на изучение темы «Вложенные циклы»:*

Группа 1: Дан список чисел, найти НОД каждого из них с введенным числом. Написать программу, используя вложенные циклы.

Группа 2: Написать программу поиска простых чисел от 1 до 10, используя вложенные циклы.

Задание для каждой группы включает в себя одну задачу на понимание изученной подтемы. Правильно выполненное задание оценивается в 3 балла.

Этап закрепления знаний составляет одна задача высокого уровня сложности. Данное задание способствует повышению познавательных и творческих способностей учащихся, развивает логическое мышление.

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к рабочей тетради.

В каждой школе дети разбиваются на 2 группы. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с учащимися в рабочей тетради имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению той или иной задачи, а также давать сигнал на проверку выполненного задания.

В начале занятия мега-учитель проводит фронтальный опрос по изученной теме на прошлом уроке (вопросы представлены ниже), после чего он проверяет выполнение домашнего задания: один ученик рассказывает об истории возникновения языка программирования Python (возможно наглядное представление материала в виде презентации), остальные учащиеся слушают и дополняют доклад.

*Вопросы фронтального опроса по теме «Циклы»:*

1. Что такое цикл?
2. Сравните цикл с переменной и цикл с условием. Какие преимущества и недостатки есть у каждого из них?
3. Что означает выражение «цикл с предусловием»?
4. В каком случае цикл с предусловием не выполняется ни разу?
5. В каком случае программа, содержащая цикл с условием, может зациклиться?
6. В каком случае цикл с переменной не выполняется ни разу?
7. Верно ли, что любой цикл с переменной можно заменить циклом с условием? Верно ли обратное утверждение?

По завершении этапа актуализации и проверки домашнего задания мега-учитель объявляет новую тему урока и излагает основные тезисы по данной теме с помощью презентации на гугл-диске.

Далее мега-урок проходит на основе нелинейного подхода. Учащимся предлагается решить следующую задачу:

Аркадий играет в «Морской бой» (большую часть правил этой игры вам знать необязательно).

Вам дано поле, состоящее из  $n \times n$  клеток. На поле должен находиться ровно один  $k$ -палубник, т. е. корабль длиной  $k$  клеток, ориентированный горизонтально или вертикально. Но Аркадий не знает, где этот корабль находится. Для каждой клетки Аркадий знает, точно ли она пустая или же она может содержать часть корабля.

Рассмотрим все возможные расположения корабля. Найдите такую клетку, которая принадлежит максимально возможному числу разных расположений корабля.

Входные данные:

В первой строке содержится два целых числа  $n$  и  $k$  ( $1 \leq k \leq n \leq 100$ ) — размер поля и размер корабля. Следующие  $n$  строк содержат поле. В каждой строке содержится  $n$  символов, каждый из которых либо «#» (обозначает определённо пустую клетку) или «.» (обозначает клетку, которая может принадлежать кораблю).

Выходные данные:

Выведите два целых числа - номер строки и номер столбца клетки, которая принадлежит максимальному числу возможных разных расположений корабля.

Если есть несколько возможных вариантов ответа, выведите любой из них. В частности, если корабль не может располагаться на поле, вы можете вывести любую клетку [22].

Для решения данной задачи обучающимся предлагается решить более простые задания, изучив материал на гугл-диске. Также можно использовать электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в рабочей тетради для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ в сети был правильно воспринят и оценен экспертом. По завершении выполнения заданий тьютор проверяет написанные программы и выставляет баллы в рейтинг. Команда, быстрее всех выполнившая задание в рабочей тетради,

получает дополнительные баллы в рейтинговую таблицу.

Учащиеся должны убедиться в правильности разработки программ для того, чтобы наглядно продемонстрировать готовую программу и прокомментировать каждый шаг. Далее группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного материала (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала). Мега-учитель, при необходимости, комментирует работы групп и указывает на недочеты.

Для закрепления полученных знаний по всей пройденной теме урока учащиеся возвращаются к заданию, представленному в начале урока. Задание находится в отдельном документе, ссылка на который имеется в рабочих тетрадях.

Учащиеся разбиваются на 4 межсетевых группы: 1 группа работает над вводом и выводом данных, 2 группа занимается составлением блок-схемы решения задачи, 3 группа работает над составлением множественных ветвлений, 4 группа работает над разработкой циклов. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа [4].

К концу занятия традиционно выступает педагог из вуза (модератор), который в необычной форме анализирует межгрупповые взаимодействия при выполнении последней задачи, а также делает комментарии и поправки в самом коде программы.

В это время эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. На этапе подведения итогов предполагается выступление тьюторов: их выводы об отрицательных и положительных моментах взаимодействия с учащимися, предварительные результаты выполнения заданий, демонстрация рейтинговой таблицы в

Google. Группа, набравшая наибольшее количество баллов за все прошлые уроки и данный урок, получает оценки «отлично». В конце занятия мегачитель предлагает в качестве домашнего задания решение трех задач с использованием ветвлений и циклов:

Домашнее задание по теме «Вложенные циклы»:

Выведите на экран (в строку)  $n$  первых простых чисел.

Подсказка: Внешний цикл должен отвечать за проверку, сколько чисел найдено, и генерацию следующего числа для проверки. Внутренний цикл проверяет, является ли простым текущее число. В качестве алгоритма проверки «на простоту» используйте обычный перебор (простое число не делится ни на что кроме себя и единицы).

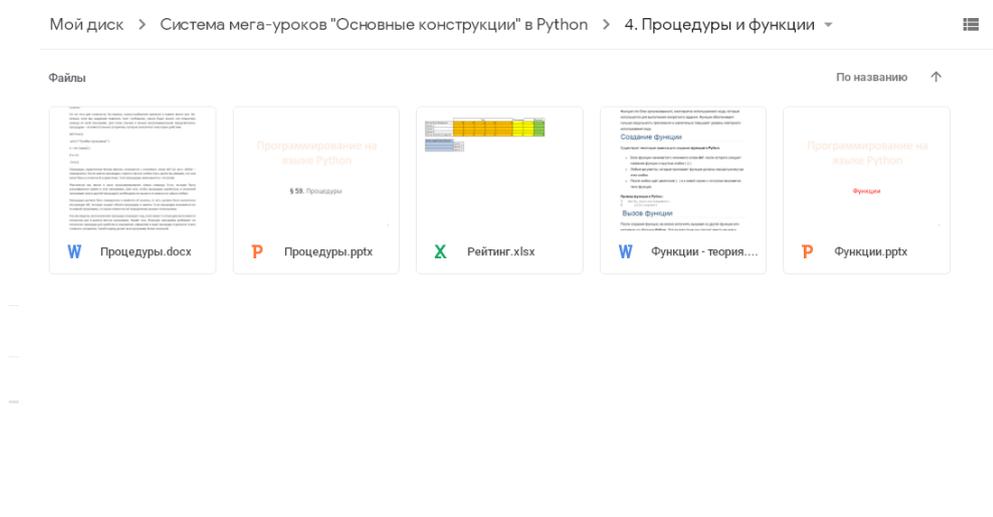
Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Вложенные циклы», а именно:

1. Знания основных вложенных конструкций: вложенные циклы, сложные условия.
2. Формирование умений применять полученные знания о ветвлениях и циклах языка программирования Питон в практической деятельности и при решении творческих задач.

**Сценарий мега-урока «Подпрограммы в Python»**

Данный мега-урок посвящен изучению процедур и функций на языке Python.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.26) который подробнее можно посмотреть по ссылке: [\[https://drive.google.com/open?id=1z6ig7wi0vVPiIrRRI0lsMZ2EIG5XO1sN\]](https://drive.google.com/open?id=1z6ig7wi0vVPiIrRRI0lsMZ2EIG5XO1sN)



*Рис.26. Теоретический материал к уроку*

Он состоит из пяти документов в форматах docx и pptx, а именно:

- 1) 2 презентации: «Процедуры» и «Функции», где представлены основные тезисы и примеры по данным темам;
- 2) «Процедуры» - документ в формате doc, в котором описана процедура в Питон с примерами программ, а также имеются дополнительные ссылки по теме;
- 3) «Функции» - представлено описание функций в Питон с наглядными примерами. Дополнительные ссылки на углубленное изучение присутствуют;
- 4) «Рейтинг» - рейтинговая таблица для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в режиме просмотра (Рис.27).

	A	B	C	D	E	F	G	H	I	J	
1		Тест					Процедуры	Функции			
2	Не группы/ задания	1	2	3	4	5	1	2	1		
3	Группа 1										
4	Группа 2										
5	Группа 3										
6	Кол-во баллов за задание	1	1	1	1	1	2	2	4		
7											
8	Всего заработано баллов										
9		Группа 1									
10		Группа 2									
11		Группа 3									
12											
13											
14											
15											
16											
17											

Рис.27. Рейтинг групп

Урок состоит из трех этапов: актуализации знаний, теоретико-практической части и закрепления.

Этап актуализации знаний представлен в форме фронтального опроса для проверки усвоения ранее изученного материала по ветвлениям и циклам в Питоне. Вопросы представлены ниже:

*Вопросы фронтального опроса по темам «Ветвления» и «Циклы».*

1. Чем отличаются разветвляющиеся алгоритмы от линейных?
2. Как вы думаете, почему не все задачи можно решить с помощью линейных алгоритмов?
3. Как вы думаете, хватит ли линейных алгоритмов и ветвлений для разработки любой программы?
4. Что такое сложное условие?
5. Сравните цикл с переменной и цикл с условием. Какие преимущества и недостатки есть у каждого из них?
6. Верно ли, что любой цикл с переменной можно заменить циклом с

условием? Верно ли обратное утверждение?

7. В каком случае цикл с предусловием не выполняется ни разу?

8. В каком случае программа, содержащая цикл с условием, может заиклиться?

9. В каком случае цикл с переменной не выполняется ни разу?

Теоретико-практическая часть предоставляется группам в виде документа на гугл-диске. Документ состоит из двух частей: первая часть включает в себя тест из пяти вопросов на понимание изученной группой темы (тест одинаков для обеих групп, каждый вопрос которого оценивается в 1 балл); вторая часть содержит задачи на систематизацию и углубление изученного материала: «Процедуры в Питоне» - 2 задачи, каждая задача оценивается в 2 балла; «Функции в Питоне» - 1 задача, оцениваемая в 4 балла.

Закрепление знаний составляет одна задача высокого уровня сложности. Данное задание способствует повышению познавательных и творческих способностей учащихся, развивает логическое мышление. Все задания представлены в ниже.

#### *Задания на изучение темы «Подпрограммы в Python»*

##### Часть 1. Тест: Подпрограммы в Python.

Вопрос 1. Какое ключевое слово стоит первым в заголовке описания функции в языке Python?

Вопрос 2. Какое служебное слово используется в описании функции в языке Python для возвращения результата работы функции?

Вопрос 3. Какое количество информации возвращает логическая функция?

Вопрос 4. В каких строках описания подпрограммы допущены ошибки?

```
1 def fib(n)
2     a, b = 0, 1
3     while a < n
```

```

4      print(a, end = "")
5      a, b = b, a + b
6      print()

```

Вопрос 5. При условии исправления ошибок в подпрограмме, определить, что выведет на экран эта подпрограмма при вызове ее в фактическим параметром 6?

Записать этот полный результат работы в строке ответа

```

def fib(n)
a, b = 0, 1
while a < n
    print(a, end = "")
a, b = b, a + b

```

Ответы на тест представлены в таблице (Таблица 8):

*Таблица 8. Ответы к тесту по теме «Подпрограммы в Python»*

№ вопроса	1	2	3	4	5
Ответ	def	return	b	1,3	011235

## Часть 2. Задания для групп

Группа №1 (процедуры):

1) напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления;

2) напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с первой.

Группа №2 (функции): Простое число называется гиперпростым, если любое число, получающееся из него откидыванием нескольких цифр, тоже является простым. Например, число 733 – гиперпростое, так как и оно само, и числа 73 и 7 – простые. Напишите логическую функцию, которая определяет, верно ли, что переданное ей число – гиперпростое. Используйте уже готовую функцию isPrime, которая приведена в презентации на гугл-диске.

Подсказка: Эту программу можно еще немного усовершенствовать: после числа 2 имеет смысл проверять только нечётные делители, увеличивая на каждом шаге значение  $k$  сразу на 2.

Задача на закрепление: написать программу, которая преобразует русскоязычный текст в звуки азбуки Морзе.

Подсказка: Для этого необходимо воспользоваться таблицей соответствий символов «точка» и «тире» с кириллическими буквами [35].

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к своему документу.

В каждой школе дети разбиваются на 2 группы. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с учащимися в документе имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению задачи, а также давать сигнал на проверку выполненного задания.

В начале занятия мега-учитель проводит фронтальный опрос по изученной теме на прошлом уроке. По завершении этапа актуализации мега-учитель объявляет новую тему урока и излагает основные тезисы по данной теме с помощью презентации на гугл-диске.

Далее учащимся предстоит самостоятельное, углубленное изучение отдельных пунктов: группе №1 предлагается подробное изучение пункта «Процедура в Питоне», группе №2 – «Функции в Питоне». Ученики могут использовать имеющиеся материалы: теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в своих документах для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ был правильно

воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет готовые программы и выставляет баллы в рейтинг.

Для того, чтобы наглядно продемонстрировать готовый продукт и объяснить изученный материал другим группам, учащиеся должны убедиться в правильности разработки программ по своим. Далее группы обмениваются готовыми программами. Каждая группа объясняет остальным ход выполнения своей программы на основе изученного пункта по теме «Процедуры и функции» (код программы каждой группы выводится на экран для более понятного и подробного объяснения материала).

Для закрепления полученных знаний по всей пройденной теме урока учащимся предоставляется более сложное задание в отдельном документе, ссылку на который группы получают по завершении теоретико-практической части урока. Учащиеся делятся на 4 межсетевые группы. Задача выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 занимается составлением блок-схемы решения задачи, 3 группа занимается составлением ветвлений, 4 группа занимается составлением циклов, 5 группа занимается составлением функций. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа.

В конце занятия выступает педагог из вуза (модератор) с анализом межгрупповых взаимодействий при выполнении последней задачи, а также делает комментарии и поправки в самом коде программы. Эксперты (студенты) завершают проверку заданий и подводят итоги решения задач в межшкольных группах. Далее идет подведение итогов: тьюторы объявляют рейтинг групп: 2 команды, набравшие наибольшее количество баллов, получают оценки «хорошо» и «отлично». В конце занятия мега-учитель предлагает в качестве домашнего задания решить задачу и составить по ней программу. Текст задачи представлен ниже.

Домашнее задание: напишите программу, которая вводит натуральное число  $N$  и находит все числа в интервале  $[0, N]$ , сумма цифр которых не меняется при умножении на 2, 3, 4, 5, 6, 7, 8 и 9 (например, число 9). Используйте функцию для вычисления суммы цифр числа.

Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Процедуры и функции», а именно:

1. Знание основных понятий: процедуры, процедуры с параметром, локальные и глобальные переменные, функции, логические функции;
2. Формирование умений и навыков применения полученных знаний о процедурах и функциях на практике и при решении интересных, познавательных задач.

### Сценарий мега-урока «Рекурсия в Питоне»

Данный мега-урок посвящен изучению рекурсии на языке Python.

Для подготовки к уроку изучения нового материала и решения задач подбирается теоретический материал, необходимый для успешного выполнения заданий. Данный материал находится на гугл-диске (рис.28), который подробнее можно посмотреть по ссылке: <https://drive.google.com/open?id=1ogmaS-2bicfUmT2mpbAzuGfabtiiZbxc>.

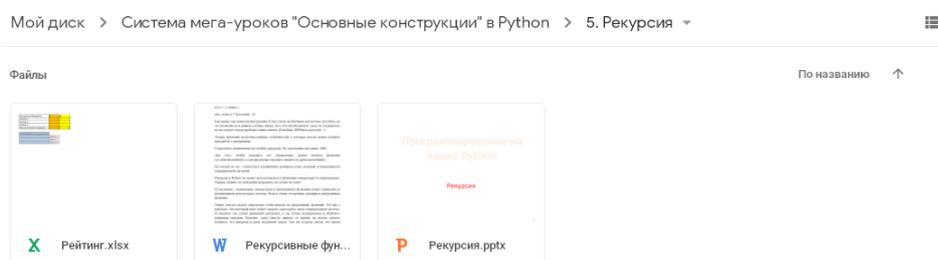


Рис.28. Теоретический материал к уроку

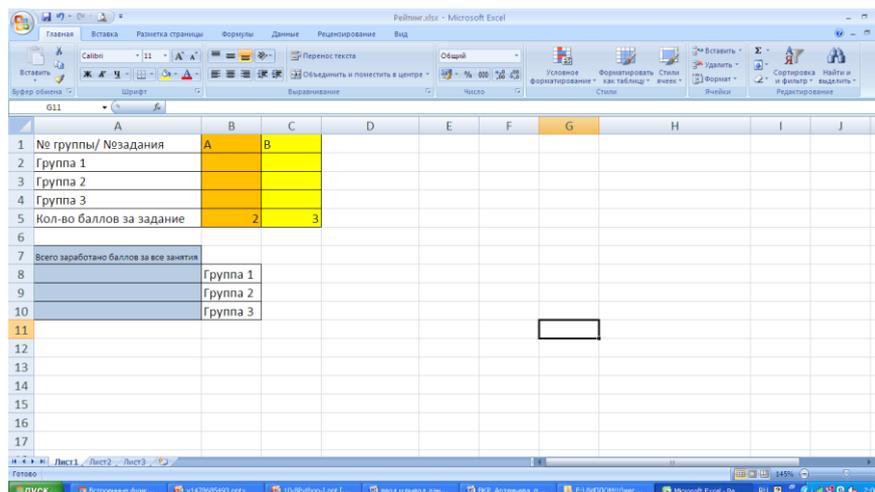
Он состоит из трех документов в форматах docx и pptx, а именно:

- 1) презентация в формате pptx «Рекурсия» - в которой представлены основные тезисы по изучаемой теме, а также приведены примеры программ;
- 2) документ «Рекурсивные функции», где более подробно описана рекурсия в Питон, имеются примеры с использованием рекурсивных функций, а также представлены дополнительные ссылки по данной теме урока;
- 3) «Рейтинг» - рейтинговая таблица для всех групп.

Доступ к материалу предоставляется в начале мега-урока. Этим материалом не запрещается пользоваться во время выполнения заданий. Также разрешается пользоваться учебником и интернетом.

Для поддержки соревновательного элемента экспертам необходимо подготовить электронную таблицу и предоставить ученикам к ней доступ в

режиме просмотра (Рис.29).



	A	B	C	D	E	F	G	H	I	J
1	№ группы/ №задания	A	B							
2	Группа 1									
3	Группа 2									
4	Группа 3									
5	Кол-во баллов за задание	2	3							
6										
7	Всего заработано баллов за все занятия									
8		Группа 1								
9		Группа 2								
10		Группа 3								
11										
12										
13										
14										
15										
16										
17										

Рис.29. Рейтинг групп

Урок состоит из трех этапов: актуализации знаний, теоретико-практической части и закрепления.

Этап актуализации знаний представлен в форме фронтального опроса для проверки усвоения ранее изученного материала по процедурам и функциям в Питоне.

Теоретико-практическая часть предоставляется группам в виде документа на гугл-диске. Документ состоит из двух задач различного уровня сложности. Задача 1 уровня (A) оценивается в 2 балла, задача уровня 2 (B) – в 3 балла

Закрепление знаний составляет одна задача высокого уровня сложности. Данное задание способствует развитию логического мышления учащихся. Все задания можно посмотреть ниже.

*Задачи на изучение материала по теме «Рекурсия»:*

A. Дано натуральное число N. Требуется получить и вывести на экран количество всех возможных различных способов представления этого числа в виде суммы натуральных чисел (то есть,  $1 + 2$  и  $2 + 1$  – это один и тот же способ разложения числа 3). Решите задачу с помощью рекурсивной процедуры.

В. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите среднее значение элементов этой последовательности (без учета последнего нуля).

В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке Python функция возвращает кортеж из пары чисел: число элементов в последовательности и их сумма. Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля).

*Задача на закрепление (для всех групп):*

Дана последовательность натуральных чисел (одно число в строке), завершающаяся двумя числами 0 подряд. Определите, сколько раз в этой последовательности встречается число 1. Числа, идущие после двух нулей, необходимо игнорировать.

В этой задаче нельзя использовать глобальные переменные и параметры, передаваемые в функцию. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметров.

Разработка программ ведется в объектно-ориентированной среде программирования Alice (описание работы с данной средой изложено в Главе 2, п. 2.3. «Формы организации работы и контроль»). Коды написанных программ учащиеся прикрепляют в виде файла с подписанными фамилиями к своему документу.

В каждой школе дети разбиваются на несколько групп. Для проведения мега-урока создается несколько межшкольных групп, за каждой из которых модератор закрепляет тьютора-студента. Для общения тьюторов с учащимися в документе имеется специальное диалоговое окно, где каждый учащийся может задавать вопросы по решению задачи, а также давать сигнал на проверку выполненного задания.

В начале занятия мега-учитель проводит фронтальный опрос по изученной теме на прошлом уроке, вопросы которого представлены ниже.

Вопросы фронтального опроса по теме «Процедуры и функции»:

1. Что такое процедуры? В чем смысл их использования?
2. Как оформляются процедуры в Python? Достаточно ли включить процедуру в текст программы, чтобы она «сработала»?
3. Что такое параметры? Зачем они используются?
4. Какие переменные называются локальными? глобальными?
5. Как в процедуре прочитать и изменить значение глобальной переменной?
6. Как оформляются процедуры, имеющие несколько параметров?
7. Что такое функция? Чем она отличается от процедуры?
8. Как по тексту программы определить, какое значение возвращает функция?
9. Какие функции называются логическими? Зачем они нужны?

По завершении этапа актуализации мега-учитель объявляет новую тему урока и излагает основные тезисы по данной теме с помощью презентации на гугл-диске.

Далее каждая группа получает ссылку на рабочую тетрадь, где находятся задачи. Учащиеся самостоятельно изучают новую тему и могут использовать теоретический материал на Гугл-диске, электронные учебники и ресурсы Интернет. После изучения материала каждая группа приступает к выполнению заданий в своих рабочих тетрадях для закрепления и систематизации полученных знаний. Учащиеся могут обратиться за помощью к модератору урока, чтобы их ответ был правильно воспринят и оценен экспертом. По завершении выполнения всех заданий тьютор проверяет готовые программы.

Затем группы обмениваются программами между собой. Тьюторы путем обсуждения выявляют наиболее качественную и простую программу и ставят дополнительные баллы в рейтинг той группы, написавшей код данной программы.

После обсуждения выполнения заданий учащимся на закрепление предлагается написать программу к задаче более высокого уровня

сложности. Задача предоставляется всем группам в виде ссылки на отдельный гугл-документ и выполняется совместными усилиями всех групп: 1 группа работает над вводом и выводом данных, 2 занимается составлением блок-схемы решения задачи, 3 группа занимается составлением ветвлений, 4 группа занимается составлением циклов, 5 группа занимается составлением функций, 6 группа составляет рекурсии к задаче. За выполнением задания следят все тьюторы групп. В процессе выполнения задания учащиеся могут обмениваться кодами программ, прикрепляя их документу с заданием, а также общаться между собой и задавать вопросы тьюторам в диалоговом окне гугл-документа.

В конце урока готовая программа выводится на экран, где педагог вуза (модератор) комментирует код программы и подсказывает другие варианты решения. Тьюторы также излагают отрицательные и положительные моменты взаимодействия с учащимися, подсчитывают рейтинг групп, а также выделяют команду с самым высоким баллом. За урок данная группа получает оценки «отлично». В конце занятия мега-учитель предлагает в качестве домашнего задания для закрепления материала по теме «Рекурсия» и самостоятельного изучения Списков в Питоне.

#### *Домашнее задание по теме «Рекурсия»*

Имеется список со списками. Необходимо найти вложенный список с максимальной суммой элементов. Решить данную задачу на Python можно тремя способами:

- 1) Обойти вложенные списки
- 2) Обойти внешний список.
- 3) Сочетание функций `sum()` и `max()`

Для решения данной задачи необходимо изучить списки в Питон. Попробуйте решить задание тремя способами и выясните, какой способ по вашему наиболее простой и понятный. Решать задание можно как в группах по 2-3 человека, так и индивидуально. Текст с кодами программ выложите не позднее трех дней после окончания данного урока на гугл-диск в папке

«Домашнее задание №7» и подпишите свои фамилии. Ссылка на папку: Данная ссылка будет закрыта для доступа по истечении трех дней после окончания этого занятия.

Данный урок поможет обучающимся получить, закрепить и углубить знания по теме «Рекурсия», а именно:

1. Формирование представлений об основных понятиях: рекурсия, рекурсивный алгоритм, итерационный алгоритм, глубина рекурсии.

2. Формирование навыков и умений использовать полученные знания на предыдущих занятиях, а также нового материала на практике.