

МИНИСТЕРСТВО НАУКИ ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ им. В.П. Астафьева» (КГПУ им. В.П. Астафьева)

Институт математики, физики и информатики

Кафедра Информатики и информационных технологий в образовании

Никитина Лидия Викторовна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**УЧЕБНЫЕ КАРТОЧКИ В ФОРМАТЕ ПЕРЕВЕРНУТОГО УЧЕБНИКА КАК
СРЕДСТВО АКТИВИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ШКОЛЬНИКОВ
В УСЛОВИЯХ ДИСТАНЦИОННОГО ОБУЧЕНИЯ**

Направление подготовки: 44.03.05 Педагогическое образование

Направленность (профиль) образовательной программы:
Математика и информатика

ДОПУСКАЮ К ЗАЩИТЕ

Зав.кафедрой:

д-р пед. наук, профессор каф. ИИТО
Пак Н.И.

(дата, подпись)

Руководитель:

канд. пед. наук, доцент каф. ИИТО
Бархатова Д.А.

23.06.2021 _____
(дата, подпись)

Дата защиты 28.06.2021

Обучающийся:

Никитина Л.В.

23.06.2021 _____
(дата, подпись)

Оценка

Отлично

Красноярск 2021

Содержание

Введение	3
Глава 1. Активизация самостоятельной деятельности школьников в условиях дистанционного обучения	7
1.1. Особенности организации дистанционного обучения в основной школе.....	7
1.2. Понятие активизации самостоятельной деятельности.....	15
1.3. Особенности восприятия учебного материала современными школьниками	19
Глава 2. Перевернутый учебник в дистанционном обучении на примере темы «Основы программирования» в 8 классе.....	26
2.1. Концепция и требования к представлению учебного материала в формате перевернутого учебника	26
2.2. Модель и содержание учебных карточек в формате перевернутого учебника на примере темы «Основы программирования» в основной школе	29
2.3. Методические рекомендации по использованию учебных карточек в формате перевернутого учебника в процессе обучения школьников.....	37
2.4. Анализ результатов экспертизы учебных карточек в формате перевернутого учебника	38
Заключение.....	47
Библиографический список.....	48
Приложение А	52
Приложение Б.....	99

Введение

Современные условия информационного бума и роста количества образовательных программ в области подготовки кадров в различных сферах деятельности, обеспечивающих экономическую конкурентоспособность страны, актуализируют дистанционное образование, как общедоступный источник доступа к знаниям всех слоев населения. Дистанционное обучение позволяет не только решить проблемы обеспеченности кадрами и ресурсами учебные заведения, но и организовать индивидуальные траектории обучения одаренных детей и детей с ОВЗ, многие из которых выбирают домашнее обучение, включающее дистанционные технологии.

Кроме того, одним из неоспоримых факторов растущей популярности дистанционного обучения является нынешняя обстановка, связанная с пандемией COVID-19. Здесь очень актуальна такая форма проведения занятий, так как с ее помощью обучение осуществляется дистанционно, без прямого контакта преподавателя и ученика. Процесс дистанционного обучения полностью соответствует требованиям ФГОС и имеет ряд преимуществ: способствует проведению занятий в доступной для студентов форме, дает возможность организовать различные формы студенческой деятельности, связанной с самостоятельным усвоением знаний, создать условия для индивидуального обучения школьников и т.д. Кроме того, наблюдается рост обучающихся, перешедших на домашнее обучение. Как отмечает Петр Сергеев: «...В России пока не ведется регулярная официальная статистика по количеству детей на домашнем обучении. На конференции Edcrunch (2019) было заявлено, что около 100 тыс. школьников выбрало такую форму образования в 2019 году. По данным Министерства образования и науки, на начало 2015-2016 учебного года в форме семейного образования в России проходило обучение всего около 8500 человек (меньше одной десятой процента от количества всех школьников). Ассоциация семейного образования провела собственное исследование и выяснила, что количество детей, находящихся на семейном обучении в России, выросло с 8452 до 15 тыс. человек на начало 2017-2018 учебного года. Но несмотря на то, что

цифры разнятся, рост интереса к семейному обучению – притом взрывной – заметен» [7].

Однако технологии дистанционного обучения требуют от обучаемого высокого уровня самоорганизации, где важная роль отводится самостоятельной работе, что вызывает необходимость разработки таких образовательных ресурсов, которые способны удержать интерес и внимание учащихся до конца обучения. Согласно исследованию 2015 года, средний процент тех, кто закончил MOOC-курсы (массовые открытые онлайн-курсы), составил приблизительно 15% [24]. Одной из причин такого низкого процента М. Спиридонов отмечает информационную перегруженность обучаемых [31]. Приходя за знаниями, обучаемые скорее хотят получить ответы на конкретные вопросы, а не «багаж знаний», из которых они сами должны выбирать необходимое им. Именно поэтому современному поколению проще ввести запрос в сети Интернет и выбрать тот контент, который отвечает их требованиям к восприятию информации.

Безусловно, современная молодежь предъявляет новые требования к представлению учебного материала, что связано с необходимостью постоянно взаимодействовать с большими потоками информации и критической их обработкой. Здесь большое значение приобретают технологии микрообучения и визуализации, которые в большей степени отвечают когнитивным способностям поколения зуммеров, являющиеся преобладающей долей обучаемых сегодня.

На основе приведенных выше утверждений, можно сделать вывод о том, что с одной стороны, тема выпускной квалификационной работы актуальна, а с другой – констатировать **противоречие** между необходимостью учителей в современных образовательных ресурсах, отвечающих запросам нового поколения, позволяющие реализовать продуктивное обучение в условиях дистанционного взаимодействия, и наличием, как правило, традиционных средств обучения, разработанных для очной классно-урочной системы.

В связи с этим возникает **проблема исследования**, какими должны быть учебные ресурсы в условиях дистанционного обучения, повышающие самостоятельную активность обучающихся и обеспечивающие ее продуктивность.

Указанную проблему можно решить через внедрение нового формата учебников, а именно перевернутых. Под перевернутым учебником в нашем исследовании понимается представление учебного материала в виде проблемных вопросов, изучение которых требует от обучаемых 5-10 минут с максимальной вовлеченностью в практическую деятельность, что позволяет активизировать и поддержать их самостоятельную работу. Для выделения границ микроблоков необходимо использовать учебные карточки, в которых материал представлен в визуализированной форме с чередованием практических заданий, что отвечает современным запросам поколения зуммеров

Таким образом, **объект исследования** заключается в процессе активизации самостоятельной работы школьников в условиях дистанционного обучения.

Предмет исследования: учебные карточки в формате перевернутого учебника как средство активизации самостоятельной работы школьников в условиях дистанционного обучения

Цель исследования: разработать учебные карточки в формате перевернутого учебника (на примере изучения основ программирования в основной школе), обеспечивающие активизацию самостоятельной работы школьников за счет визуализации и применения вопросно-задачного подхода, а также описать методику их применения.

Для достижения цели поставлены следующие **задачи**:

1. Провести анализ особенностей организации дистанционного обучения в основной школе.
2. Описать понятие и средства активизации самостоятельной деятельности школьников в условиях дистанционного обучения.
3. Провести анализ и выявить современные запросы школьников к представлению учебного материала.

4. Описать концепцию и требования к представлению учебного материала в перевернутом формате.

5. Построить модель и описать содержание учебных карточек в формате перевернутого учебника на примере темы «Основы программирования» в основной школе в 8 классе.

6. Описать методические рекомендации по использованию учебных карточек в формате перевернутого учебника в процессе обучения школьников в условиях дистанционного обучения.

Область применения полученных результатов: в работе описаны модель и содержание перевернутого учебника на примере темы школьного курса информатики «Основы программирования», реализованного через учебные карточки, а также представлены рекомендации по применению полученных результатов исследования в обучении информатике в средней школе. Разработанные учебные карточки в формате перевернутого учебника могут быть использованы, как средство организации самостоятельной работы школьников в рамках дистанционного обучения, также и как вспомогательный материал при организации очного обучения в условиях класса. Разработанная модель перевернутого учебника и содержание учебных карточек могут быть использованы в качестве руководства для создания аналогичных средств.

Глава 1. Активизация самостоятельной деятельности школьников в условиях дистанционного обучения

1.1. Особенности организации дистанционного обучения в основной школе

Дистанционное обучение (удаленное) – форма получения знаний, в том числе образования, на расстоянии, с сохранением компонентов учебного процесса и использованием интерактивных и интернет-технологий образования [6].

Взаимодействия в рамках дистанционного обучения делятся на два основных типа [2]:

- синхронные взаимодействия между участниками образовательных отношений – это средства общения ученика и учителя в данный момент времени;
- асинхронные взаимодействия между участниками образовательных отношений – это средства общения ученика и учителя с задержкой во времени.

Основными видами синхронной связи, распространёнными в Российской Федерации, являются [2]:

1. Аудиоконференция – средство связи, при котором голосовое общение учителя и учеников (ученика) осуществляется через компьютер в данный момент времени.
2. Видеоконференция – средство связи, обеспечивающее одновременно двухстороннюю передачу, обработку, преобразование и представление видеоинформации и голосовую связь учителя и учеников (ученика) через компьютер в данный момент времени.
3. Чат – это средство общения, посредством обмена сообщениями между учителем и учениками (учеником) через компьютер в данный момент времени.
4. Интерактивная виртуальная доска – виртуальное подражание классной доске, для передачи информации в данный момент времени.
5. Виртуальный класс – виртуальное подражание в совместном обучении во время встречи в классе. Программное обеспечение, одновременно реализующее большинство компонентов синхронного взаимодействия. Наиболее

близкий тип синхронного виртуального взаимодействия к фактическому (в непосредственной близости), включающий в себя ряд основополагающих функций [2]:

- демонстрация любых учебных данных, материалов, например, презентация;
- универсальный аудио чат или видео чат;
- функция «поднятая рука»;
- личный взаимообмен данными между учеником (учениками) и учителем;
- интерактивная виртуальная доска.

Основными видами асинхронной связи, распространёнными в Российской Федерации, являются [2]:

1. Веб-форум – специализированные базы данных на сайтах (разделы) школ, созданные и назначенные для онлайн-обсуждений, где каждый участник может оставить или передать сообщение по теме, которая соответствует специальному разделу в образовательном процессе.

2. Электронный почтовый ящик – это механизм обмена или сетевое обслуживание для пересылки и получению электронных писем между участниками образовательного процесса. Данный тип асинхронного взаимодействия является наиболее простым и удобным для участников образовательного процесса, так как практически у любого пользователя сети Интернет есть своя учетная запись (свой аккаунт).

3. Обмен файлами – специализированная площадка, созданная специально для школы, способствующая хранению файлов и передачи информации всех участников образовательного процесса в процессе учебной деятельности.

Кроме того существует, синхронно-асинхронное взаимодействие между участниками образовательных отношений – третий тип, который поддерживает возможности смешанного дистанционного обучения. Лучшее решение – это разнообразие типов взаимодействия. Чтобы обучающиеся были как можно

интенсивнее вовлечены в образовательную деятельность, необходимо стремиться смешивать различные типы и формы контента. Заинтересовать обучающихся поможет сочетание компонентов таких как, видео, тексты и изображения [28].

В современном мире существуют различные классификации моделей дистанционного обучения, рассмотрим одну из них. В. И. Снегурова описывает в своей работе базовые модели для построения методической системы удаленного обучения [30].

В соответствии с основными характеристиками, рассмотрим модели дистанционного обучения, рис.1 [30]:

- недоступность индивидуального общения между участниками образовательного процесса в режиме реального времени;
- применение сетевого ресурса в образовательном процессе, как основа для учебной деятельности;
- общение в сети между участниками образовательного процесса, через Интернет.



Рисунок 1. Классификация моделей дистанционного обучения

1. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса и возможностью легкой адаптации в образовательный процесс, с использованием двусторонней связи в режиме асинхронного взаимодействия.

2. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса и возможностью легкой адаптации в образовательный процесс, с использованием двусторонней связи в режиме синхронных (единовременных) индивидуальных консультаций.

3. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса и возможностью легкой адаптации в образовательный процесс, с использованием двусторонней связи в режиме синхронно-асинхронного (смешанного) взаимодействия.

4. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса, кроме того, применение дополнительных учебных материалов, с использованием двусторонней связи в режиме асинхронного взаимодействия.

5. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса, кроме того, применение дополнительных учебных материалов, с использованием двусторонней связи в режиме синхронных (единовременных) индивидуальных консультаций.

6. Дистанционное онлайн-обучение, проходящее на фундаменте специально созданного сетевого ресурса, кроме того, применение дополнительных учебных материалов, с использованием двусторонней связи в синхронно-асинхронного (смешанного) взаимодействия.

Разница между описанными выше моделями складывается из следующих рассматриваемых критериев: тип взаимодействия; применение дополнительных учебных материалов одновременно с сетевым ресурсом, с помощью которых происходит связь между участниками учебного процесса: информационные ресурсы из сети Интернет, цифровые образовательные ресурсы (фотографии, видеофрагменты, объекты виртуальной реальности, статические и динамические

модели, аудиозаписи и др.); инструменты информационных технологий, которые оказывают преимущественное влияние на построение образовательного процесса: отбор таких форм проведения учебных консультаций и занятий, которые позволят активизировать познавательную деятельность обучающихся, установление наиболее интенсивных и результативных методов обучения.

Вышенаписанное свидетельствует о том, что акцентируется внимание на необходимости построения и реализации такой методической системы дистанционного обучения, на основе которой могут быть реализованы все данные модели.

Адаптация образовательного процесса к дистанционному обучению в системе образования ставит под вопрос способность цифровых образовательных технологий обеспечить полноценную реализацию учебного процесса, предложить наиболее эффективные, адекватные ресурсы, сервисы и инструменты для организации плодотворной и удобной работы в цифровой оболочке,

На основе экспресс-анализа сервисов для организации образовательного процесса и цифровых образовательных ресурсов школ в дистанционном режиме можно вывести следующий перечень проблем [36]:

1. Школы, учителя, школьники имеют доступ к разнообразному и обширному комплексу учебных материалов по многим предметам школьной программы (видео, тексты, иллюстрации и др.). Вопреки тому, что часть материалов уступает по качеству на мировом уровне, в целом она содержит почти все программы, ступени овладения и освоения по большинству предметов. Многие учебные материалы доступны бесплатно.

2. В этом комплексе текстовых, иллюстративных материалов отсутствует хорошая система навигации, удобные инструменты для их включения в учебный процесс. Их разработка является срочной.

3. Практическое отсутствие современных систем управления образованием в школе, которые переводят учебный процесс в онлайн-режим. Ряд предлагаемых на рынке решений может быть модифицирован под необходимый функционал. Задача также является срочной.

4. Для активного распространения дистанционного обучения недостаточным является предложение комплексных тренажеров, интерактивных заданий с аналитикой и обратной связью. Существующие на сегодняшний день задания представляют собой тесты с несколькими вариантами ответов или с одним ответом. Они лишены яркости, технологичности и увлекательности. Как правило, в них не используются современные технологические решения. Этот недостаток не позволяет ни реализовать функцию самоучителя, ни облегчить повседневные задачи учителя по проверке выполнения заданий. Причем большая часть этих ресурсов не бесплатная. Учителя в большинстве случаев не имеют опыта их использования в образовательном процессе.

5. Преодоление этого недостатка возможно за счет ускоренной разработки заданий для самостоятельной работы и новых тренажеров, дополняющих существующие наборы. Также становится важным интенсивное повышение уровня квалификации учителей по применению этих ресурсов.

6. Механизм образования обладает правом доступа к большому количеству универсальных инструментов для общения, работы с документами, обратной связи и организации групповой работы. Этот доступ используется очень редко. В то же время существует явная нехватка совокупного решения для школ, где эти ресурсы были бы сформированы в некоторую целостную систему реализации учебного процесса, так как дистанционное обучение может вызвать путаницу и недопонимание среди участников образовательного процесса.

7. Методы дистанционного обучения развиты недостаточно. Практически отсутствуют теоретические и практические разработки по возрастному характеру использования цифровых материалов как для начальной, так и для основной, старшей школы.

8. Не существует общего решения для полной адаптации школ к дистанционному формату обучения. Но можно собрать набор, который в определенной степени удовлетворяет потребности ученика и его родителей, класса, образовательной организации, региона с временной организацией дистанционного обучения. Однако, сейчас нет времени разрабатывать различные

передовые практики. Серьезность ситуации требует создания сети экспериментальных школ, где будет прорабатываться интеграция различных ресурсов, а также разработка новых ресурсов и инструментов, методов их использования и оценки их эффективности.

9. Существенным препятствием для реализации будущих, проанализированных сервисов, инструментов и ресурсов значатся нормативные границы на утилизацию образовательных услуг, сетей и санитарных стандартов. При переходе школ на дистанционное обучение необходимо максимально снизить эти барьеры.

10. Самым главным барьером является ресурсный. Большинство современных интерактивных сервисов разрабатываются и подаются коммерческими компаниями. Они не могут в течение длительного времени поддерживать эти сервисы бесплатно, так как им нужны средства на разработку. Поэтому наиболее рациональным сейчас является выделение школам дополнительных средств, «раскрашенных» специально на закупку цифровых образовательных ресурсов, позволяющих строить программы дистанционного формата обучения. По мнению экспертов, достаточно выделять 2 тысячи рублей в год на обучающегося, чтобы рынок современных образовательных ресурсов в России действительно заработал и обеспечивал учителей и обучающихся качественными цифровыми продуктами.

Кроме того, существует ряд требований к организации дистанционного формата обучения установленных Положениями: о государственной аккредитации учреждений среднего общего образования Российской Федерации, о лицензировании учреждений среднего общего образования в Российской Федерации и об образовательном учреждении среднего общего образования Российской Федерации, которые ориентированы на качество, к ним относится обеспечение [19]:

– телекоммуникационное – скорость передачи канала учреждений, надлежит быть вполне хватающей для построения учебного процесса в полном

объеме, с использованием предусмотренного календарного графика и учебным планом, которые регламентируют дистанционный формат обучения;

– материальная база – реализация учебного процесса в образовательных учреждениях с использованием дистанционного формата обучения, должна соответствовать требованиям по санитарно-гигиеническим нормам охраны здоровья учеников и сотрудников образовательных организаций, оснащению учебных аудиторий, лабораторий и средств телекоммуникаций, за исключением требований по предоставлению учебных площадок, литературы где обязательно должны выполняться условия качественного технического оснащения, то есть присутствие компьютерной техники, кроме того пользование коммерческим программным обеспечением должно быть лицензировано;

– кадровое – учителя должны периодически проходить повышение квалификации или переподготовку в области новых образовательных и информационных технологий, кроме того, аттестоваться;

– информационное – учреждения, осуществляющие образовательный процесс с использованием дистанционного формата обучения, должны иметь оперативный доступ к информационным ресурсам, которые в свою очередь обязаны обеспечить полноценные условия для передачи знаний обучающимся и должны быть основаны на компьютерной сети.

Таким образом, дистанционное обучение – это процесс взаимодействия ученика и учителя посредством расстояния с сохранением всех присущих обучению компонентов (целей, содержания, организационных форм, методов и средств обучения) и с использованием технических средств (интерактивных средств, интернет-технологий и др.). Особенности организации дистанционного обучения включают в себя: требования к организации образовательного процесса и разработку цифровых образовательных ресурсов школ в дистанционном режиме, применение актуальных форм, методов и средств, которые следует применять в дистанционном формате обучения.

1.2. Понятие активизации самостоятельной деятельности

Активизация самостоятельной деятельности учеников является одной из основных проблем современной педагогики. Актуальность обусловлена поиском и необходимостью разработки оптимальных методических приёмов и учебных средств. Познавательная деятельность и самостоятельность – это два компонента, которые являются обязательным условием для успешного и эффективного обучения, их формирование способствует закладыванию стремлению к развитию и совершенствованию своих знаний и умений, с использованием их на практике, а также способности самостоятельного построения своей учебной деятельности. Главная задача школы отводится на становление самостоятельной личности и ориентирование на подготовку к дальнейшему успешному социальному общению в условиях адаптации к информационному обществу. Для начала разберемся, что такое «самостоятельная деятельность ученика».

Ученые отмечают высокую активность, обучающихся, если они выполняют задания без непосредственной помощи учителя. Поэтому этот показатель рассматривается в качестве главного критерия самостоятельной работы.

В Российской педагогической энциклопедии дается следующее определение самостоятельной деятельности обучающихся: «многообразные виды индивидуальной и коллективной учебной деятельности школьников, осуществляемой ими на классных и внеклассных занятиях или дома по заданиям без непосредственного участия учителей» [25].

В статье О. А. Бекетовой можно встретить следующее толкование термина самостоятельная деятельность обучающихся: «форма организации их учебной деятельности, осуществляемая под прямым или косвенным руководством преподавателя, в ходе которой учащиеся преимущественно или полностью самостоятельно выполняют различного вида задания с целью развития знаний, умений, навыков и личностных качеств» [4].

И. В. Муштавинская, О. Б. Даутова и другие в своей работе дают следующее определение самостоятельной деятельности – «...это вид познавательной деятельности, в котором предполагается определенный уровень

самостоятельности во всех структурных компонентах деятельности – от постановки проблемы до осуществления контроля, самоконтроля и коррекции с диалектическим переходом от выполнения простых видов работы к более сложным, носящим поисковый характер, с постоянной трансформацией руководящей роли педагогического управления в сторону ее перехода в формы ориентации и коррекции с передачей всех функций самому обучающемуся, но лишь по мере овладения методикой самостоятельной работы» [13].

Исходя из данных точек зрения, можно сделать вывод, что всякая организованная учителем активная деятельность учеников, ориентированная на выполнение дидактической цели урока: поиск знаний, их понимание, закрепление, формирование умений и навыков, обобщение и систематизация знаний – это самостоятельная деятельность.

Кроме того, самостоятельная деятельность обучающихся – это средство обучения, которое [26]:

- соответствует назначенной дидактической цели и задаче урока в соответствующей ситуации;

- развивает умения и навыки для конкретного класса познавательных задач, повышает уровень (от низкого к высокому) знаний на этапе перехода от незнания к знанию;

- играет роль управления самостоятельной деятельностью обучающегося и служит средством учительского руководства в образовательном процессе;

- является одним из главных аспектов в формировании психологической заинтересованности в самостоятельном и систематическом пополнении знаний и умений, которые помогают определять и разграничивать информацию при выходе из затруднительной ситуации в решении новых познавательных задач.

Развитие активности и формирование познавательной самостоятельности на сегодняшний день является значимым и на уроках информатики, так как связано с большим потоком и непрерывностью в увеличении объема информации, а также

её быстрым «старением». Перед учителями стоит вопрос о том, как формировать умения и навыки, которые направлены на самостоятельность, самообразование и выход на самостоятельное приобретение знаний, чтобы обучающиеся смогли реагировать на любые трудности жизни.

На практике в педагогике используются различные способы повышения познавательной активности, основными из которых являются разнообразные формы, методы, средства обучения, выбор нескольких их сочетаний, которые в возникших ситуациях стимулируют активность и самостоятельность учеников.

Для успешного развития интереса к самостоятельной деятельности в процессе дистанционного обучения информатике необходимы средства обучения, под которыми понимают любые объекты, созданные человеком и используемые в образовательном процессе в качестве носителей учебной информации, инструмента деятельности учителя и обучающихся для достижения поставленных целей обучения и развития [32].

Также одной из важнейших задач в современном образовании выступает предъявление высоких требований к образовательным результатам, в особенности к результатам освоения основной образовательной программы.

Отсюда вытекает следующая система средств обучения информатике, которая сводится к следующим основным пособиям [33]:

- учебник по информатике;
- учебные пособия (дополняют учебник) – это карточки - задания самостоятельной работы обучающихся, сборники задач для вычислений, материалы для проверки знаний обучающихся и т.п.;
- материальные предметы (иллюстративные модели) – это различного рода таблицы, раздаточный материал и счетный материал, приборы, измерительные инструменты.

Учебник как основное средство обучения информатике, полно и структурировано раскрывают содержание курса информатике, отражают уровень знаний, умений и навыков, которыми должны овладеть обучающиеся в каждом классе.

Учебники выполняют следующие дидактические функции:

- помогают сознательно усваивать знания;
- учат приемам умственной деятельности;
- способствуют формированию определенных умений и навыков;
- в том числе и навыков самостоятельной работы;
- контроля и самоконтроля;
- помогают учителю воспитывать и развивать обучающихся.

Учебник, содержащий в себе иллюстрации, дает большие возможности для проведения разнообразной по форме и содержанию коррекционной работы.

Расположение в учебниках иллюстраций и упражнений способствует развитию абстрактного мышления у детей, так как постепенный переход от предметной наглядности к условной, даёт возможность успешнее формировать у обучающихся навыки моделирования. Не меньшее значение имеет система представленных в учебнике иллюстраций для развития конкретного мышления детей.

Содержащиеся в учебнике рисунки и сюжетные материалы, чертежи, схемы, таблицы, образцы программ помогают обучающимся не только осознавать многие зависимости, но и дают основу и материал для обобщений, знакомят их с различными сторонами окружающей действительности.

Одно из важных мест среди обучения занимают карточки с заданиями по информатике. Эти пособия предназначены для того, чтобы помочь учителю в организации самостоятельной работы учеников на различных этапах урока. Они могут быть использованы для проведения контрольных и обучающих самостоятельных работ, организации фронтальной, групповой и индивидуальной работы в классе, восполнение пробелов в знаниях детей.

В современных школах используются еще информационные средства обучения, такие как:

- ПК обучающегося;
- электронный учебник;
- электронная доска, заменившая меловую доску;

- обучающие интерактивные игры, тесты, упражнения;
- электронные журналы – научные, методические, художественные, общеобразовательные (в дополнение к учебнику) и т.п.

На основе изученной литературы, методических рекомендаций и личного педагогического опыта можно сделать вывод, что их использование в практике на уроках информатики не обеспечивает должный уровень мотивации и стремления обучающихся к дальнейшему изучению нового знания, поскольку данные средства обучения не гарантируют:

- наглядность явлений и объектов, которые недоступны для наблюдения в реальной жизни;
- облегчить ученику процесс понятия;
- передачу более полного и точного представления о существующей действительности;
- осмысления и запоминания фактов и закономерностей.

Таким образом, активизация самостоятельной деятельности является ключевым понятием в обучении, так как определяется как процесс активности обучающихся, направленный на определенный предмет деятельности, напрямую зависящий от взаимодействия учеников с окружающей действительностью, а также обеспечивает сложную систему осознанных действий для продуктивного применения знаний в новых условиях в соответствии с образовательной целью.

1.3. Особенности восприятия учебного материала современными школьниками

В современном информационном обществе новые технологии изменяют и преобразовывают окружающую среду человека, при этом меняется его жизнь и деятельность. Начинается формирование цифрового общества, которое превозносит в нашу жизнь цифровую и электронную среды, оказывая этим значимое воздействие на растущие поколения, что влечет за собой новые информационные и образовательные запросы молодежи.

Система образования обязана быстро и гибко реагировать на запросы цифрового общества, а также организация профессиональной деятельности в условиях системных вызовов в недалеком будущем должна отвечать его потребностям [15]:

- создание условий для перспективного будущего и обновленного содержания обучения;
- изменение большинства форм образовательной деятельности и межличностного взаимодействия;
- совершенствование образовательного процесса, за счет управления;
- оптимизация процесса получения социального опыта, путем вхождения в социальную среду и социальные сети;
- полная адаптация учеников к реальной жизни.

На сегодняшний день в современной школе обучаются дети, которые являются представителями поколения «Z». Именно поэтому школа должна не только формировать определенный набор знаний у школьников, но и пробуждать у них стремление к самообразованию.

К одним из особенностей их познавательной сферы относятся способы восприятия информации: обучающиеся легко ориентируются в потоке информации, но теряются, когда учебный материал представлен в большом объеме, особенно, требующий внимательного прочтения, т.к. имеют слабую концентрацию внимания (8 секунд) и высокую переключаемость. Представители данного поколения склонны к потребности восприятия визуализированной информации, клиповому восприятию информации, «чтению заголовками» [17].

Поколение «Z» характеризуется следующим образом [18]:

- главная мотивация - интерес (если обучающимся не интересно, они не будут учиться, даже за большое вознаграждение);
- низкая концентрация внимания (из-за отсутствия интереса);
- важность понимания, что они делают и зачем они это делают (например, обучающийся не будет учиться хорошо или осваивать предмет только

потому, что это важно для родителей, ему нужны конкретные знания для реализации сегодняшних задач);

- не принимают иерархии и авторитетов (уважения заслуживают личные качества человека, а не его возраст и статус);

- не любят расписания и нормированный учебный или рабочий день (хотят заниматься и выполнять задачи по собственному расписанию, не стремятся перевыполнять план и ставить амбициозные цели);

- мыслят краткосрочными перспективами и не готовы ждать (не думают о будущем, так как настоящее меняется слишком быстро);

- цифровые люди, не видят мир без компьютеров и технологий (большинство имеет бесперебойный доступ в Интернет, к гаджетам и мобильным устройствам. Они не могут просто отказаться от гаджетов, это и есть их жизнь).

В образовательной среде современные дети не хотят быть бездейственными и инертными, они увлечены поисковыми системами, отсюда вытекает заключение о том, что учителя должны приспособиться к их потребностям и изменить свои установки в обучении.

Более того, ученики современных школ не способны различать правдивую информацию от ложной, они дезориентированы в потоке контента. Основатель Школы программирования для детей CODDY Оксана Селендеева, рассказывает об отличии таких детей от других поколений и предоставляет доступ к рекомендациям по их обучению, которые следует использовать при разработке и подаче учебного материала [8]:

1. Оставляйте отзывы (взаимодействие с учеником). Мотивируйте обучающихся, критикуйте с полным обоснованием, умейте правильно направить и побудить их к более глубокому изучению предмета.

2. Гаджеты. Используйте в своей работе гаджеты для организации обучения, на которых доступен образовательный контент. Перспектива системного обучения, то есть там, где удобно ребенку.

3. Не забывайте про свои устойчивые медиапривычки. Большинство своего времени дети проводят в социальных сетях, стройте короткие обучающие модули, чтобы увлечь детей.

4. Внесите изменения в процесс обучения. Электронная почта (email, gmail и др.) является неактуальной, скучной для обучающихся, кроме того, они ждут быстрых ответов на свои сообщения. Используйте в своей работе мессенджеры и видеосвязь для взаимодействия с детьми, ведите групповые чаты для обсуждения.

5. Возможность свободного самообразования. Детям нравится независимость, выбор выполнения блоков (в каком-либо порядке) по собственному желанию. Это может послужить на пользу учителю в развитии уверенности в себе и мотивации. Например, в CODDY модульное обучение — ученик может комбинировать учебные модули, чтобы осваивать интересующие его предметы.

6. Используйте визуализацию данных. Смайлики, мемы и картинками отличный способ для визуального, добавляйте в занятия аудиоролики или видеоролики.

7. Делите информацию по блокам. Структурированность информации позволяет ученикам быстрее вовлечься в процесс обучения. Подача монотонного материала увеличивает риск отвлечения на социальные сети. Следует разделить теорию, диалог и практику (например, начало урока с разговора, актуализации, потом дайте задание, а затем снова обсуждение).

8. Используйте цифровой формат. С рождения дети привыкли к технологиям, им хочется получить быстрый доступ к нужной информации на уроках, поэтому онлайн-сервисы, которые служат инструментом хранения и передачи материалов будут особо полезны.

Процесс урока становится более эффективным. Медленное «хождение» по классу и монотонная речь учителя выводят детей из равновесия. Они лучше воспринимают быструю информацию и материал в оптимистичном тоне.

Несомненным вкладом в современных условиях в развитие российского образования и педагогики стало издание русского перевода книги Дж. Коатс [9], где автор рекомендует педагогам:

- хорошо подготовить и организовать процесс обучения (структурируйте его так, чтобы ученик смог точно определить, что от него требуется и в какие сроки он должен выполнить задание);

- обеспечить взаимодействие с учениками, более эффективно использовать предоставленное время (интервалы 10-15 минут, в течении которых меняется вид деятельности);

- информация, которую предоставляет педагог, не должна быть излишней (обучающиеся нацелены на конкретизированные знания), учебный материал должен быть интересным и заметным.

Именно поэтому в системе школьного образования происходят серьезные изменения, основной целью обучения становится ориентация на усвоение опыта творческой деятельности обучающихся. При таком подходе к обучению меняется и его содержание. На первый план выходят приемы и методы обучения, требующие активной познавательной деятельности, развивающие умения сравнивать, анализировать, видеть проблемы, обобщать, искать решения, формулировать гипотезы и корректировать полученные результаты. Изменяется и роль учителя, которая заключается в педагогической поддержке индивидуального развития обучающегося, а не в передаче знаний.

Это значит, что требования к представлению информации изменились, следует внедрять такие педагогические технологии, которые способствуют развитию мышления, а также включать в познавательную деятельность некоторые компоненты состязательности, использовать систему вознаграждений», формировать навыки обсуждения, продвигать свои идеи и предоставлять информацию небольшими порциями [17].

При дистанционном формате обучения в целях повышения качества образования и его эффективности организация образовательного процесса должна соответствовать таким современным формам и методам обучения, как [11]:

- метод «Дерево решений» – способ оценивания преимуществ и недостатков различных вариаций решений;
- метод ПОПС-формула – применяется для обратной связи, позволяющий ученикам легко сформировать и предоставить свое мнение в четкой и краткой форме, при этом аргументируя его конкретными примерами или контрпримерами;
- метод «Займи позицию» – применяется для выявления имеющегося мнения, увидеть сторонников и противников какой-либо позиции, начать обоснованное обсуждение спорных вопросов;
- метод «Портфолио» – современный метод наблюдения и фиксации результатов выполненных заданий обучающимися, укомплектование творческой и учебной копилки;
- метод проектов – это способ независимого получения не хватающих знаний из разных источников, развитие у обучающихся навыков исследования (построение гипотез, сбор информации, наблюдательность, выявление проблем, эксперимент, анализ, общение и др.);
- интерактивная экскурсия – форма обучения, при которой обучающиеся воспринимают и приобретают знания в месте расположения изучаемых объектов (музеи, исторические места, памятники, предприятия, выставки и др.);
- творческие задания – задания, требующие от учеников не воспроизведения информации, а творческого подхода, так как содержат элемент неопределенности и имеют несколько подходов.

Рассмотрим виды интерактивных средств для реализации данных форм и методов в дистанционном формате обучения [11]:

- электронное учебное пособие (электронный учебник), электронная рабочая тетрадь;
- Интернет-ресурсы, сайты и энциклопедии;
- интерактивные постеры, презентации, хроники и ленты времени, видео, контент дополненной реальностью;

- интерактивные задания и упражнения;
- интерактивные викторины, игры, тренажеры, квесты, тесты и т.д.

Используя элементы современных технологий, такие как коллективный способ обучения (КСО), теория решения изобретательных задач (ТРИЗ), развивающего, проектного, проблемного обучения, а также технологии интерактивного и мобильного обучения и др., такие интерактивные форматы проведения уроков, как мозговая атака (брейнсторм), деловые и ролевые игры, ситуационный анализ, круглый стол (дебаты, дискуссия) [27], способствуют преодолению отрицательных тенденций развития «цифрового поколения».

Таким образом, опираясь на современные запросы обучающихся для представления материала, следует вносить в его изложение изменения, а также подбирать или составлять такие средства обучения, которые помогут учителю активизировать и затронуть больший спектр деятельности в самостоятельной работе.

Глава 2. Перевернутый учебник в дистанционном обучении на примере темы «Основы программирования» в 8 классе

2.1. Концепция и требования к представлению учебного материала в формате перевернутого учебника

Как уже отмечалось в первой главе, в зависимости от требований, которые предъявляются к средствам обучения и в связи с особенностями восприятия школьников к представлению учебного материала видится актуальным разработка новых учебных ресурсов для обучения современного поколения обучающихся в основной школе.

Педагогу следует обучиться модифицировать собственную деятельность с целью обучения цифрового поколения Z с применением соответствующего учебно-методического обеспечения.

В данной взаимосвязи цифровые образовательные ресурсы должны отображать указанную специфику [16]:

- технологичность организации обучения за счет четких условий к результатам обучения, стремительной обратной взаимосвязи, рационального применения цифровой техники и технологий, в том числе мобильные;
- индивидуализированная направленность обучения за счет возможности подбора своего сценария обучения;
- свободный и доступный формат организации учебного процесса и учебной деятельности обучаемого за счет цифровых технологий;
- прагматичность и минимизирование обучения точно по времени за счет предоставления стремительной, важной и никак не излишней информации.

Перевернутый учебник подразумевает замену стратегии обучения, порождает потребность пересмотра структурной композиции и содержания учебной информации в учебниках. В них рационально принять во внимание на ученике подход, от вопросов и задач к обучению [3] и «перевернутость» учебного материала.

Система перевернутого электронного учебника состоит в замене целой последовательной формы представления учебного материала (рис. 2.) в нелинейную, сетевую структуру с вопросно-задачной основной линией (рис. 3.) [3].

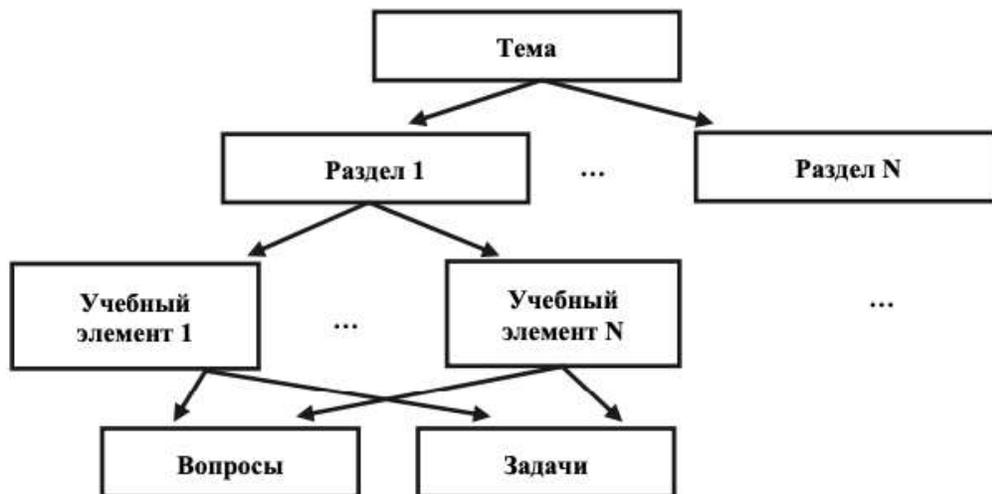


Рисунок 2. Конструкция классического представления содержания учебника

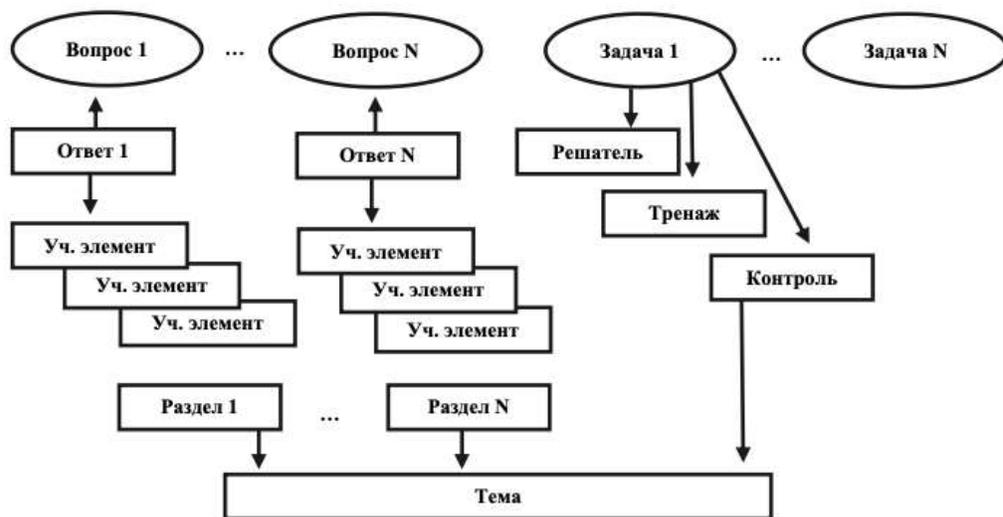


Рисунок 3. Конструкция представления содержания перевернутого электронного учебника

Работая в рамках вопросно-задачного подхода, учитель ставит обучающихся перед необходимостью самостоятельно искать способы решения проблемной задачи, для которой у них нет заранее рассказанного, готового учителем способа, но при этом есть достаточно знаний, применяя которые в

нестандартных ситуациях, обучающиеся способны прийти к правильным выводам [12].

Вопросно-задачный подход – есть специально организованное и систематически осуществляемое обучение в виде разрешения разнообразных учебных задач с помощью направляющих вопросов [14].

О функциональной многогранности вопросно-задачного подхода и его роли в повышении эффективности образования можно судить по широкому кругу научных монографий и диссертационных исследований, устанавливающих, что задачи в обучении [35]:

- являются средством: углубленного усвоения материала, систематизации и обобщения учебных знаний, организации различных видов мыслительной деятельности, реализации прикладной направленности образовательного процесса и его деятельностной и личностной направленности;

- обуславливают развитие: алгоритмического и логического мышления, информационной культуры учеников, активности, самостоятельности и творческих качеств личности;

- выступают в качестве формы представления учебного содержания;

- определяют формирование: общеучебных умений, навыков и способов деятельности, мировоззренческой и мотивационной сфер личности, познавательной и общей культуры обучающихся, широкого спектра компетенций;

- решают задачи: методической направленности обучения, реализации внутрипредметных и межпредметных связей, преемственности в образовательном процессе, осуществления индивидуализации и дифференциации в обучении.

Таким образом, обучение с использованием перевернутого учебника в условиях вопросно-задачного подхода должно строиться по следующему алгоритму:

1. Постановка вопросов перед обучающимися, где первым ставится основополагающий вопрос, на который нельзя ответить одним предложением, и который заключает в себе суть изучаемого блока учебного материала и позицию в практической стороне его применения.

2. Основополагающий вопрос разделяется на серию проблемных вопросов, которые являются конкретными предметными и тематическими предпосылками к формированию ответов на основополагающий вопрос [38].

3. Решение на конкретных примерах: используются задания на проблематизацию, упражнения-вызовы, показывающие отсутствие у обучающихся опыта или навыков, необходимых для решения актуальных для них задач.

4. Расширение практики до теоретических знаний: изучение материала начинается не с теории, а с решения конкретной практической задачи, в ходе которого обучаемые обогащают свои знания, а полученная информация преобразуется в понимание и опыт.

5. Закрепление полученных знаний, систематизация и установка межпонятийных, межпредметных связей.

Таким образом, аналогичные ресурсы и сервисы никак не предоставляют знания ради знаний, а дают только нужную информацию обучающемуся по появившемуся у него запросу.

2.2. Модель и содержание учебных карточек в формате перевернутого учебника на примере темы «Основы программирования» в основной школе

Одной из основных тем информатики, практически с начала ее появления в школах в нашей стране, продолжает оставаться тема «Основы программирования» [10].

По мере развития информатики в школьном плане количество изучаемых тем увеличилось. С темой «Программирование» обучающиеся знакомились только поверхностно, получая знания только об основах. И лишь в профильных классах эту тему изучали в полном объеме.

Изучение в школе того или иного общеобразовательного предмета преследует три основополагающих цели [1]:

- развитие определенных форм мышления (развивающая цель);

– вносить вклад в научные взгляды учеников (теоретико-мировоззренческая цель);

– передавать обучающимся конкретные практические умения и способности, полезные в обучении, в жизни и еще в предстоящей профессиональной деятельности (прагматическая цель).

Содержательное направление линии «Основы программирования» имеет отношение к абсолютно всем трем целям.

Проанализируем развивающий аспект обучения программированию. Бесспорно, развитие алгоритмического мышления у обучающегося совершается тем эффективнее и успешнее, нежели раньше оно наступает. Целью обучения алгоритмизации считается освоение структурной методологией построения алгоритмов учащимися. Это обозначает, что обучающиеся обязаны обучиться применять главные управляющие структуры в практике построения алгоритмов: иметь навык разделять задачу в подзадачи, использовать способ последовательной детализации алгоритма, следование, ветвление и цикл. Педагогичный инструмент для этого ранее изобретен — это разнообразные обучающие исполнители алгоритмов: роботы, чертежники, черепахи, кенгуру.

Использование данных исполнителей в заданиях весьма результативно с методической точки зрения. Главные достоинства – показательность работы исполнителя, четкость, доступность разрешаемых вопросов и поддержка методологии структурной алгоритмизации.

При изучении алгоритмизации в пропедевтическом курсе информатики главным считается аспект развития. Помимо этого, в базовом курсе информатики к нему прибавляются другие подходы, которые необходимо причислить к теоретическим целям школьной информатики. Таковых подходов два.

Первый подход – кибернетический. Ознакомление с информационными основами процессов управления. Роль алгоритмов во этой проблеме обуславливается соответствующим способом: алгоритм управления – это информационная составная часть каждой системы управления. В ходе управления совершается предоставление сведений об состоянии контролируемого объекта по

линии обратной связи, а по линии прямой связи – управляющая информация, т.е. указания управления. Очередность установок управления составляет алгоритм управления. Управляющий объект обязан это «знать».

Следующий подход состоит в связи линии компьютера с линией алгоритмизации и программирования, с наиболее глубочайшим раскрытием концепции программного управления электронно-вычислительной машины.

Исследование алгоритмизации в аспекте программирования сопряжено с внедрением последней группы определений (кроме алгоритмических структур): величина, тип и структура величины, переменной и константы, присваивания значения переменной, выражения (арифметические, логические, строковые), операции (действия) над величинами. Если вплоть до изучения этой проблемы обучающиеся работали с базами данных и электронными таблицами, то они уже умеют представление об величинах и их свойствах. Возможно начать с этих представлений, вводя понятие величины в языках программирования.

При наличии незначительного количества учебного времени, программирование в базовом курсе возможно изучать только на вводном уровне. Главная цель ограничена рамками этой же линии компьютера: раскрывается понятие программного управления работой компьютера. Исследование проходит на примерах обычных программ. Представлено, как организован простой диалог компьютера с человеком: компьютер задает вопросы, обучающийся отвечает, компьютер откликается на ответ в соответствии с его содержанием. Показывает, как сформированы простые расчеты, к примеру, вводится числовая последовательность, на вывод получаем ее среднее арифметическое значение, либо вводятся два числа, на вывод принимаем их максимальный общий делитель (алгоритм Евклида). Этого в целом в достаточной мере с точки зрения установленной учебной цели.

Проанализируем прагматическую цель изучения программирования, которая состоит в изучении и овладении основными принципами профессионального программирования. С практическими программами возможно совершить больше, нежели с языками программирования в школьном уровне. Эта цель может

быть поставлена только лишь с целью специального либо факультативного курса информатики.

Ведущей целью темы «Основы программирования» является формирование умений и навыков программирования, развитие у школьника основ алгоритмического и системного мышления. Умение мыслить последовательно (алгоритмически) принимается за способность решать задачи различного уровня и происхождения, требующие от ученика построения плана действий для достижения требуемого результата [34], а под систематическим мышлением - способность человека познавая мир, способен устанавливать связи между явлением и предметами, выявлять существующие закономерности, прогнозировать их развитие и решать возникающие проблемы [29].

В результате изучения темы обучающиеся должны овладеть основными приемами программирования. Рассмотрим конкретные знания, умения и получаемые обучающимися для темы «Основы программирования» в 8 классе [21].

На выбранном языке программирования обучающиеся должны знать:

- основные понятия «алгоритм», «исполнитель» и «система команд исполнителя»;
- построение главных алгоритмических структур в выбранном языке программирования;
- главные алгоритмические структуры, к которым относятся следование, ветвление и цикл.

На выбранном языке программирования обучающиеся должны уметь:

- программировать стандартные, несложные линейные, разветвляющиеся и циклические алгоритмы;
- на алгоритмическом языке составлять алгоритмы для решения простых задач в словесной форме;
- выполнять трассировку алгоритма с использованием трассировочных таблиц.

Таким образом, модель перевернутого учебника выглядит следующим образом, рис. 4 [23, 40].

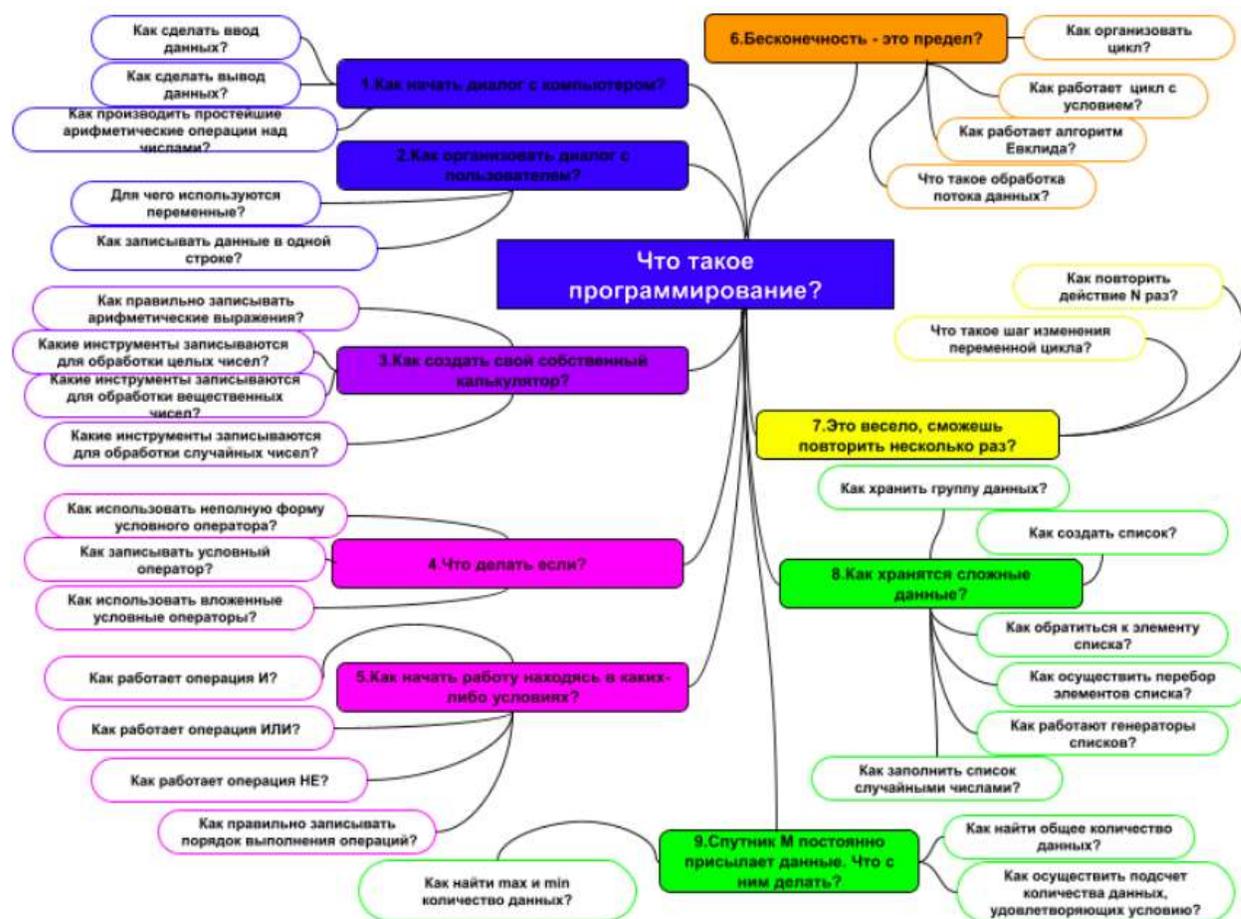


Рисунок 4. Рисунок-схема учебника

Программирование является одной из самых сложных тем для понимания обучаемыми, не смотря на ее актуальность и востребованность в нашей жизни. Большая часть современных источников в данной сфере требуют внимательного и тщательного прочтения и изучения, в то время как обучаемые хотят сразу решать практические задачи и видеть конкретный результат. Использование карточек с пошаговой инструкцией выполнения тех или иных команд, в конечном итоге приведут к ответу на главный вопрос урока. Обучающиеся сразу же погружаются в процесс программирования на компьютерах, поэтапно отвечая на проблемные вопросы карточки.

Таким образом, у учителя нет необходимости объяснять материал у доски, а потом консультировать каждого ученика за компьютером. В рамках данного подхода учитель сразу же выступает в роли консультанта, что позволяет реализовать индивидуальный подход в процессе обучения. В конце урока подводятся итоги и систематизация знаний совместно со всем классом.

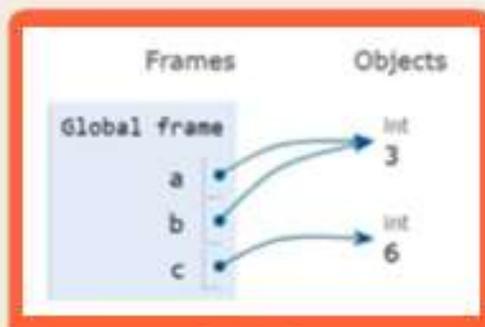
Python – это язык программирования общего назначения, направленный в первую очередь на повышение производительности самого программиста, а не кода, который он пишет. Программный код во многом лаконичен и понятен даже тем, кто на нем никогда не писал. Благодаря этому обучение программированию на языке Python происходит намного быстрее, то есть обучающимся физически приходится усваивать меньше материала. Это особенно важно в случае юной аудитории, которым не свойственна усидчивость. Именно поэтому в разработке комплекта учебных карточек будем использовать язык программирования Python.

Приведенный пример карточки урока по программированию содержит в себе основополагающий вопрос: для чего в программе используют переменные. Проблемные вопросы (как связаны переменные и данные, как узнать адрес объекта и как можно называть переменные) подводят обучаемых к пониманию необходимости использовать переменные в программе, рис. 5 [5, 37].

ДЛЯ ЧЕГО В ПРОГРАММЕ ИСПОЛЬЗУЮТ ПЕРЕМЕННЫЕ?

ССЫЛКА НА ПАМЯТЬ В КОМПЬЮТЕРЕ

Переменная в Python – это ссылка на значение, которое хранится в памяти компьютера. Ссылку можно менять на другое значение.



Любые данные в Python это объект, любая переменная — ссылка на объект. Не существует никаких данных, которые не являются объектами.

Для программы:

```
a=3  
b=3  
c=a+b
```

Переменные `a` и `b` будут ссылаться на один и тот же объект.

АДРЕС ОБЪЕКТА

Адрес хранения значения можно получить с помощью функции `id()`, где в скобках указывается переменная, которая ссылается на данное значение.

КАК МОЖНО НАЗЫВАТЬ ПЕРЕМЕННЫЕ

В переменных можно использовать латинские буквы (`a..z`, `A..Z`), цифры (`0..1`) и нижнее подчеркивание (`_`). При этом, заглавные и строчные буквы различаются. Название переменной не может начинаться с цифры.

```
>>> a=3  
>>> A=4  
>>> id(a)  
2074073296  
>>> id(A)  
2074073312  
>>> c1=A  
>>> id(c1)  
2074073312
```

Рисунок 5. Фрагмент урока в условиях вопросно-задачного подхода

В конце обучаемым предлагается провести самоконтроль полученных знаний и обсудить материал с учителем, рис. 6 [37].

The image displays three educational components. At the top left is a yellow card titled 'ВЫВОДЫ' (Conclusions) with two numbered points. At the top right is a white card with a red border titled 'ПРОВЕРЬ СЕБЯ' (Check Yourself) containing a code snippet and its output. At the bottom is a computer monitor graphic with a Python logo in the bottom-left corner, displaying a terminal window with Python code and its output.

ВЫВОДЫ

1. Любые данные в Python это объекты.
2. Если объекты «одинаковы», то они хранятся по одному адресу в памяти. Иными словами, $a == b$ и $id(a) == id(b)$ это эквивалентные утверждения.

ПРОВЕРЬ СЕБЯ

После выполнения программы какое значение и какой адрес в памяти компьютера будет иметь переменная c .

```
a=2
b=3
c=a*b-4
```

если $id(a)=2074073280$
 $id(b)=2074073296$

Загрузите среду программирования Python.

В консоле наберите следующий текст. Есть ли ошибки в названиях переменных? Если есть исправьте их. Чему будет равна переменная C_sum ? Какой адрес имеет объект C_sum в памяти компьютера?

```
>>>1_c=10234
>>>1_c
>>>id(_c)
>>>C_sum=0
>>>C_sum=_c+_c
>>>C_sum
```

Рисунок 6. Завершение урока в условиях вопросно-задачного подхода

Следующей карточкой, с которой буду работать ребята, ставит перед ними вопрос, как организовать диалог с компьютером, объясняющий суть команд `input()` и `print()`.

Таким образом, разработанные обучающие ресурсы предоставляют возможность активного обучения, так как соответствуют требованиям, которые запрашивают дети поколения зуммеров. Они способствуют разработке таких

форм деятельности для обучения учеников, которые относятся к самостоятельному освоению знаний, а также созданию условий для индивидуального обучения школьников, проведению занятий в дистанционном формате обучения.

2.3. Методические рекомендации по использованию учебных карточек в формате перевернутого учебника в процессе обучения школьников

Рассмотрим проведение занятий в дистанционном формате обучения, поскольку она является актуальной на сегодняшний день.

Для того, чтобы провести плодотворный дистанционный урок, к нему нужно подготовиться. Нужно выбрать обучающую цифровую платформу, организовать доступ на выбранную платформу своим ученикам. На соответствующей платформе размещается вспомогательный дидактический материал в рамках перевернутого учебника для проведения урока.

Учитель организывает учебный процесс, при котором обучающиеся переходят к самостоятельной работе, ученик выполняет задание в соответствии с карточкой и общепоставленной целью урока.

Порядок работы на занятии заключается в том, что обучающиеся индивидуально разбирают представленный материал и прикрепляют свою работу в оболочку.

Курируя учеников в онлайн режиме, учитель при необходимости корректирует их деятельность в формате удаленного общения. Осознание учеником изучения и закрепления нового материала происходит индивидуально.

На этапе завершения основной работы на уроке проводится рефлексия в онлайн режиме, в ходе которой обучающиеся под курированием учителя должны прийти к общему выводу и включить полученную информацию на уроке в общую систему знаний.

На основе дистанционной формы обучения, несложно применять учебные карточки в очной системе обучения и сконструировать дидактику проведения такого урока.

На первом этапе необходимо качественно подготовить вспомогательный дидактический материал для проведения урока в рамках перевернутого учебника.

Контроль, понимание и общие концепции обучающихся, также является немаловажным фактором для контроля и организации проведения урока данного типа.

Характерной чертой проведения урока в рамках перевернутого учебника является самостоятельная работа обучающихся, ученик выполняет задание в соответствии с карточкой и общепоставленной целью урока. Учитель в свою очередь выполняет роль куратора, контролирующего общеобразовательный процесс. При этом учитель всегда может скорректировать деятельность учеников в формате диалога.

При завершении основной работы на уроке проводится рефлексия, в ходе которой обучающиеся под кураторством учителя должны прийти к общему выводу и включить полученную информацию на уроке в общую систему знаний.

2.4. Анализ результатов экспертизы учебных карточек в формате перевернутого учебника

К средствам обучения имеется ряд требований, которые показывают качество и эффективность в обучении. Рассмотрим группы требований, которые предъявляются к средствам обучения (таблица 1) [33].

Таблица 1 – Группа требований к средствам обучения

Группа	Характеристика
Эргономические требования	– высокая пригодность средств к деятельности учителя и обучающихся; – удобное и простое использование в обращении; – соответствие требованиям безопасности жизнедеятельности и гигиены труда.
Эстетические	Средства в соответствии с требованиями должны

требования	иметь товарный вид: – единство и целостность композиции; – гармонизированные формы и т.п.
Экономические требования	Соответствие критерию «цена – качество», средства должны быть недорогие, либо долгосрочные в использовании.
Методические требования	Соответствие возможностей средств обучения формам и методам обучения, которые согласуются с современными требованиями.
Дидактические требования	Соответствие средств обучения с: – психофизиологическим закономерностям обучения обучающихся; – конкретным определением педагогических задач; – учебными программами и учебниками; – дидактическими принципами; – целям обучения и воспитания.

В соответствии с данной группой средств была разработана анкета, которая позволяет оценить на сколько данное средство имеет возможность активизировать самостоятельную деятельность обучающихся. Вследствие этого был создан и осуществлен опрос в Google Формах, рис. 7, рис. 8.

Опрос включал в себя 4 блока вопросов [39]:

1. Оформление авторского средства обучения программированию.
2. Оценка предметной составляющей авторского средства.
3. Практикоориентированность авторского средства обучения.

4. Возможность использования данного средства в дистанционном обучении.

Блок 1. Оформление авторского средства обучения программированию

Уважаемые эксперты, просим вас пройти по ссылке <https://drive.google.com/drive/folders/1ElbwDTD0QUIT-p-sK8XTVUjI9wcJNOGi?usp=sharing> и оценить авторское средство.

*** Обязательно**

1.1. Соответствие авторского средства изложению учебного материала в визуализированной форме: *

- Полностью не соответствует
- Частично соответствует
- В большей части соответствует
- Полностью соответствует

Рисунок 7. Опрос в Google Форме

Блок 2. Оценка предметной составляющей авторского средства

Уважаемые эксперты, просим вас пройти по ссылке <https://drive.google.com/drive/folders/1ElbwDTD0QUIT-p-sK8XTVUjI9wcJNOGi?usp=sharing> и оценить авторское средство.

2.1. Соответствие разработки содержанию школьного курса информатики по теме «Программирование» в 8 классе: *

- Полностью не соответствует
- Частично соответствует
- В большей части соответствует
- Полностью соответствует

2.2. Как вы оцениваете форму изложения в авторском средстве от 1 до 5? *

1 2 3 4 5

Не соответствует методическим требованиям Полностью соответствует методическим требованиям

Рисунок 8. Фрагмент критериев оценивания

В экспертизе приняло участие всего 15 человек. Выпускников университета КГПУ им. В.П. Астафьева было 12 человек, остальные – учителя математики и информатики, физики и информатики.

В целом эксперты считают, что представленное авторское средство полностью соответствует изложению материала в визуализированной форме, лишь 3 человека выступают за «в большей части соответствует», 1 человек проголосовал за «частично» и 1 человек «полностью не соответствует» (рис. 9).

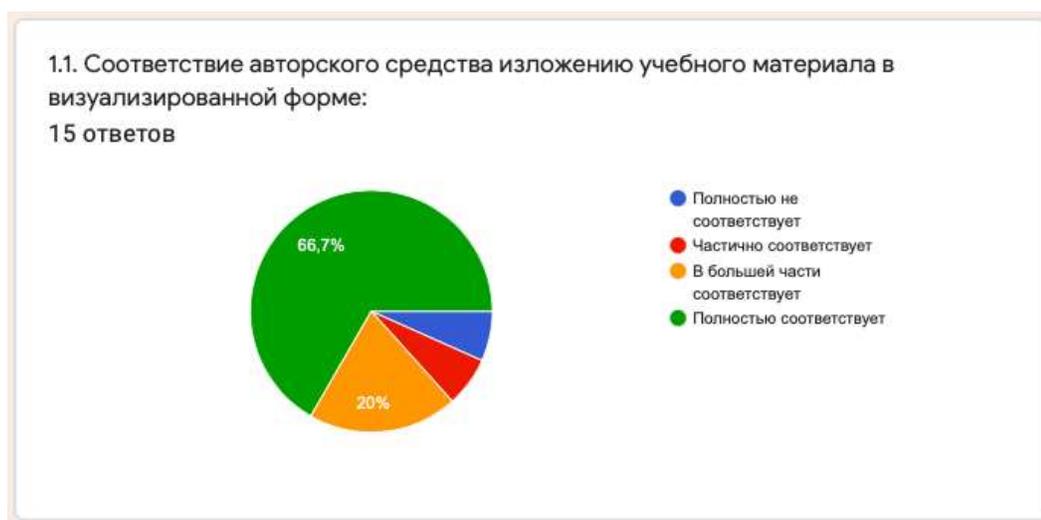
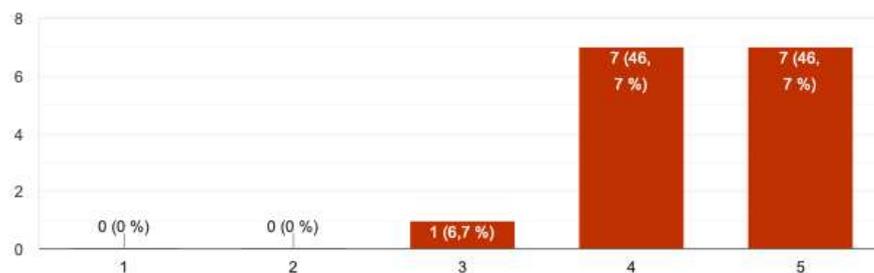


Рисунок 9. Оценка визуализации авторского средства

Опрошенные считают, что оформление авторского средства представлено в целостной и структурированной форме, а также, что его просто и удобно использовать на уроках, лишь 1 человек считает, что данное средство требует доработки в оформлении, рис. 10.

1.2. Соответствие авторского средства целостности и структурированности изложения учебного материала:

15 ответов



1.3. Как вы оцениваете простоту и удобство в обращении авторским средством?

15 ответов

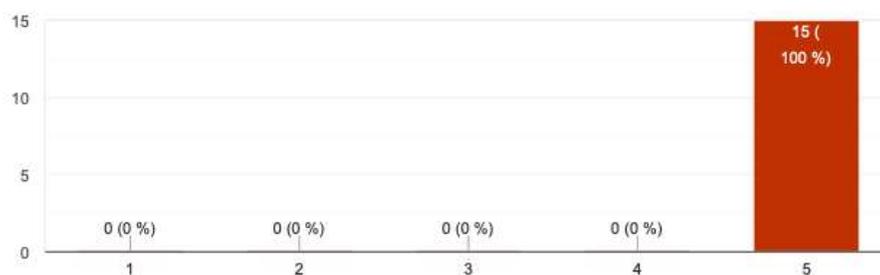


Рисунок 10. Оценка оформления авторского средства

На основе обработанных результатов были получены данные о предметной составляющей авторского средства. Многие из экспертов оценили это средство, как соответствующее содержанию школьного курса информатики по теме «Программирование» в 8 классе, что видно из круговой диаграммы, рис. 11.

Блок 2. Оценка предметной составляющей авторского средства

2.1. Соответствие разработки содержанию школьного курса информатики по теме «Программирование» в 8 классе:

15 ответов

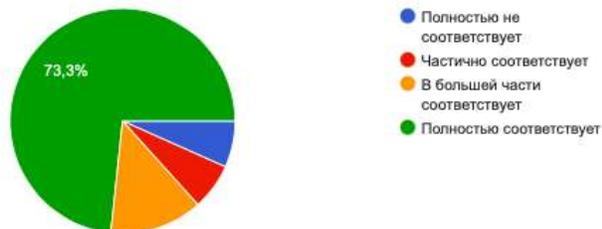


Рисунок 11. Оценка предметной составляющей авторского средства

Для большинства участников представленное авторское средство соответствует методическим и дидактическим требованиям, значительно выигрывает по сравнению с традиционными учебниками приходится на 4 и 5 уровень, что является хорошим показателем. Но все же требует некоторых доработок (рис. 12).

2.2. Как вы оцениваете форму изложения в авторском средстве от 1 до 5?

15 ответов

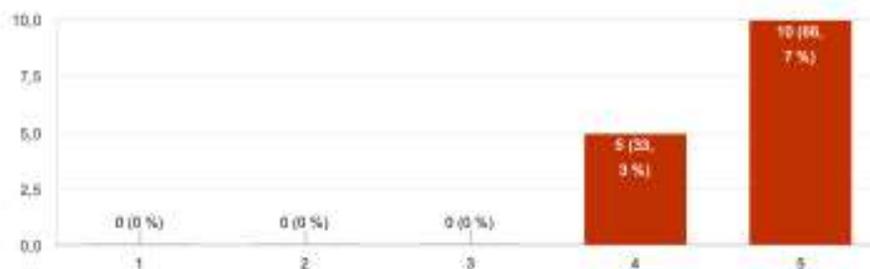




Рисунок 12. Результаты оценки предметной составляющей авторского средства

Получены данные о практикоориентированности авторского средства на основе обработанных результатов. Из столбчатой гистограммы видно, что большая часть экспертов оценили средство, как активизирующее практическую деятельность обучающихся, лишь 1 человек выставил «практическая деятельность практически отсутствует». Кроме того большая часть экспертов склоняются к тому, что «возможно организовать самостоятельную деятельность обучаемых, но учитель при этом будет активным консультантом с каждым учеником», 4 человека отметили, что «возможно организовать самостоятельную деятельность для всех обучаемых (возможна консультация учителя)», 4 человека

проголосовали за «возможно организовать самостоятельную деятельность для всех обучаемых» и 2 человека за «возможно организовать самостоятельную деятельность для сильных обучаемых», что видно из круговой диаграммы, это свидетельствует о разносторонней практикоориентированности авторского средства, рис. 13.



Рисунок 13. Результаты оценки практикоориентированности авторского средства

В целом эксперты рекомендуют представленное авторское средство как «самостоятельное полноценное дидактическое средство обучения без использования дополнительных средств» в дистанционном обучении, 6 человек

выступают за «возможно, но с привлечением дополнительных средств обучения» и 1 человек проголосовал за «возможно, но только с учителем», рис. 14.

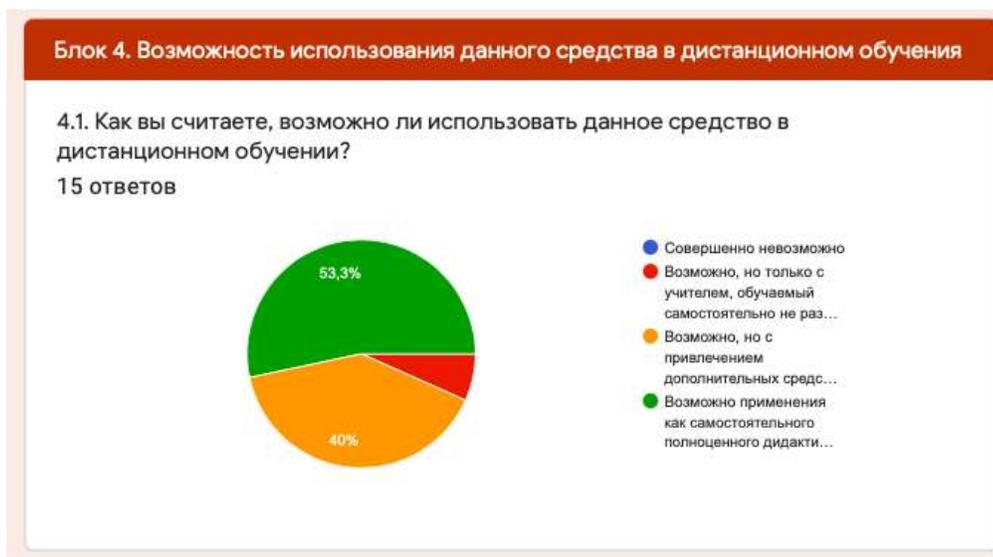


Рисунок 14. Результаты оценки возможности использования авторского средства в дистанционном обучении

Основным результатом данного опроса была положительная оценка авторского средства обучения, кроме того, следует учитывать мнение участников опроса, которые определили, что некоторые аспекты требуют изменений. Аспекты, требующие изменений, в частности, указали на то, что разработанное средство дает возможность выявить и продемонстрировать значительные возможности решения учителем познавательных и дидактических задач в дистанционном режиме. Таким образом, возможно с большей степенью утверждать, что цель работы достигнута.

Заключение

Представим выводы и результаты, которые были обнаружены в ходе выполнения исследовательской работы. В завершении основными результатами работы являются следующие.

Во-первых, проанализировав научно-педагогические источники, удалось определить такие понятия, как «дистанционное обучение», «активизация самостоятельной деятельности», «средства обучения» «поколение Z», «вопросно-задачный подход», «перевернутый учебник». Были выделены особенности и модели взаимодействия в дистанционном обучении.

Во-вторых, были конкретизированы дидактические средства обучения, требования к ним и современные запросы школьников, для активизации самостоятельной работы в основной школе, на основании которых выбран основной подход для реализации учебных карточек в формате перевернутого учебника.

В-третьих, были разработаны структура и содержание демонстрационного комплекта для обучения программированию в основной школе в дистанционном формате обучения. В данной работе продемонстрированы средства для представления и закрепления нового материала, организации контроля и коррекции образовательных результатов. Комплект включает в себя учебные карточки в формате перевернутого учебника, а также методические рекомендации по темам программирования в основной школе с помощью сервиса Canva.com.

Помимо этого, была проведена оценка разработанного авторского средства и проанализированы ее результаты, которые в большей степени являются положительными и доказывают возможность использования демонстрационного комплекта в дистанционном формате обучения для решения дидактических задач учителем информатики.

Перечисленные выше выводы и результаты, позволяют сделать вывод о том, что все задачи исследовательской работы выполнены, а поставленная цель достигнута.

Библиографический список

- 1) Алгоритмизация и программирование [Электронный ресурс] URL: https://lbz.ru/metodist/authors/informatika/2/files/Algoritmizatsiya_i_programmirovani_e.pdf (дата обращения: 29.04.2021).
- 2) Андреев С.Е., Воронов М.П. Виды синхронных и асинхронных взаимодействий между участниками образовательной деятельности // Научное обозрение. Технические науки. – 2017. – № 2. – С. 5-10.
- 3) Асауленко Е.В. Анализ процесса развития методов контроля знаний с позиции теории черного ящика // Педагогическое образование в России. 2016. № 5. С. 41–46.
- 4) Бекетова О.А. Активация самостоятельной работы учащихся [Электронный ресурс]. URL: <https://multiurok.ru/files/aktivatsiia-samostoiatiel-noi-raboty-uchashchikhsia.html> (дата обращения: 30.04.2021).
- 5) Брайсон П. Python для детей и родителей / П. Брайсон ; [пер. с англ. М.А. Райтмана]. — М: Издательство «Э», 2017. — 352 с.
- 6) Дистанционное обучение: суть, терминология и особенности [Электронный ресурс]. URL: <https://vuz24.ru/news/o-dstantsionnom-obrazovanii/distancionnoe-obuchenie-sut-terminologiya-osobennosti#expir> (дата обращения: 30.04.2021).
- 7) Как устроено домашнее обучение в РФ и стоит ли переводить на него ребенка [Электронный ресурс] URL: <https://trends.rbc.ru/trends/education/5e7cc6fd9a79471501bb1e2e> (дата обращения: 05.05.2021).
- 8) Как учить поколение Z, чтобы школьные годы не прошли даром [Электронный ресурс] URL: <https://rb.ru/opinion/uchit-pokolenie-z/> (дата обращения: 30.04.2021).
- 9) Коатс, Дж. Поколения и стили обучения // Дж.Коатс – М.: МАПДО; Новочеркасск: НОК, 2011. 121 с.
- 10) Макарова Н.В. Информатика. 10-11 классы. / Н.В. Макарова, Ю.Ф. Титова, Ю.Н. Нилова. — М: БИНОМ. Лаборатория знаний, 2016. — 336 с.

11) Методические рекомендации по подготовке к занятиям в интерактивной форме [Электронный ресурс] URL: http://www.rusacad.ru/docs/polozeniia/Metod_Podgot_Inter_Form.pdf (дата обращения: 29.04.2021).

12) Минаева М.Ю. Технологии задачного обучения [Электронный ресурс] URL: <https://xn----dtbhtbbrhebfpirq0k.xn--plai/other/articles/file/77459-soobshchenie-po-teme-tekhnologiya-zadachnogo-obucheniya> (дата обращения: 27.04.2021).

13) Муштавинская И.В. Современные педагогические технологии основной школы в условиях ФГОС / И.В. Муштавинская, О.Б. Даутова, О.Н. Крылова. — СПб.: ЛитРес, 2019. — 180 с.

14) Нахман А.Д. Задачный подход как технологическая основа процесса обучения // Международный журнал экспериментального образования. 2018. № 2. С. 34-39.

15) Носкова Т.Н. Дидактика цифровой среды / Т.Н. Носкова. — СПб.: ЛитРес, 2020. — 738 с.

16) Пак Н.А. Концепция трансформационных и перевернутых электронных учебников / Н.А. Пак, Е.Г. Потупчик, Л.Б. Хегай // Вестник Российского университета дружбы народов. Серия: Информатизация образования. — 2020. — № 2. — С.153-168.

17) Пестерева О. А. Особенности обучения поколения Z: проблемы и пути решения [Электронный ресурс] URL: <https://www.bsu.ru/content/page/21087/4-osobennosti-obucheniya-«pokoleniya-Z»-problemi-i-puti-resheniya.PDF> (дата обращения: 27.04.2021).

18) Поколение Z: как его учить [Электронный ресурс]. URL: <https://lala.lanbook.com/pokolenie-z-kak-ego-uchit>(дата обращения: 29.04.2021).

19) Положение о дистанционном образовании [Электронный ресурс] URL: <https://docs.google.com/document/d/1KBbpWbqKCl6SuDq76R6gwP01iACHWBQg5U2UxYKt4Io/edit?usp=sharing> (дата обращения: 30.04.2021).

- 20) Поляков К.Ю. Информатика. 7-9 классы / К.Ю. Поляков, Е.А. Еремин. — Москва: БИНОМ, 2016. — 80 с.
- 21) Поляков К.Ю. Информатика. 7-9 классы: методическое пособие / К.Ю. Поляков, Е.А. Еремин. — М.: БИНОМ. Лаборатория знаний, 2016. — 80 с.
- 22) Поляков К.Ю. Информатика, 8 класс / К.Ю. Поляков, Е.А. Еремин. — М: БИНОМ. Лаборатория знаний, 2019. — 104 с.
- 23) Поляков К.Ю. Методические указания к проведению уроков в для 7-9 классах / К.Ю. Поляков, Е.А. Еремин. — М: БИНОМ. Лаборатория знаний, 2019. — 78 с.
- 24) Почему никто не завершает онлайн-курс и почему это неважно [Электронный ресурс] URL: <https://www.influencive.com/no-one-finishes-online-course-doesnt-matter/> (дата обращения: 05.05.2021).
- 25) Российская педагогическая энциклопедия [Электронный ресурс]. URL: <http://www.otrok.ru/teach/enc/txt/18/page16.html> (дата обращения: 30.04.2021).
- 26) Самостоятельная работа учащихся на уроке [Электронный ресурс]. URL: https://ciur.ru/glr/glr_spon/DocLib2/Методические%20и%20иные%20документы%20для%20обеспечения%20образовательного%20процесса/Самостоятельная%20работа%20учащихся.pdf (дата обращения: 30.04.2021).
- 27) Секреты эффективного обучения поколения Z [Электронный ресурс]. URL: <http://cmro.ru/?p=5397> (дата обращения: 30.04.2021).
- 28) Синхронное и асинхронное обучение: что подходит вашим ученикам? [Электронный ресурс]. URL: <https://antitreningi.ru/info/online-obrazovanie/sinhronnoe-i-asinhrонное-obuchenie/> (дата обращения: 06.05.2021).
- 29) Системное мышление. Развитие системного мышления [Электронный ресурс] URL: <https://b-trainika.com/blog/myshlenie-sistemnogo-myshleniya/> (дата обращения: 29.04.2021).
- 30) Снегурова В.И. Модели дистанционного обучения в системе среднего образования [Электронный ресурс]. URL: <http://www.unibob.ru/upload/files/18.pdf> (дата обращения: 30.04.2021).

31) Спиридонов М.Ю. Ваши курсы не проходят до конца? Вот как это исправить [Электронный ресурс] URL: <https://rb.ru/opinion/ne-zakonchili-onlajn-kursy/> (дата обращения: 05.05.2021).

32) Средства обучения и требования к ним [Электронный ресурс]. URL: http://college4.ru/attachments/article/177/sredstva_ob.pdf (дата обращения: 29.04.2021).

33) Федеральный государственный общеобразовательный стандарт среднего общего образования: текст с изм. и доп. на 2014 г. / М-во образования и науки Рос. Федерации. – М.: Просвещение, 2020. – 33 с.

34) Чебурина О.В. Формирование алгоритмического мышления в обучении программированию игр / О.В. Чебурина // Наука и перспективы. — 2017. — № 2. — С.1-5.

35) Шмигирилова И.Б. Задачный подход как основа эффективного обучения школьников / И.Б. Шмигирилова // Актуальные проблемы методики обучения информатике и математике в современной школе. — 2019. — С.449-456.

36) Экспресс-анализ цифровых образовательных ресурсов и сервисов для организации учебного процесса школ в дистанционной форме / И.А. Карлов [и др.]. — М: НИУ ВШЭ, 2020. — 56 с.

37) Canva [Электронный ресурс] URL: <https://www.canva.com> (дата обращения: 15.04.2021).

38) Intel Обучение для будущего [Электронный ресурс] URL: <http://window.edu.ru/resource/002/55002> (дата обращения: 27.04.2021).

39) Google Формы [Электронный ресурс] URL: <https://www.google.ru/forms/about/> (дата обращения: 26.05.2021).

40) Mindomo [Электронный ресурс] URL: <https://www.mindomo.com/ru/login> (дата обращения: 15.05.2021).

Учебные карточки в формате перевернутого учебника по теме «Основы программирования»

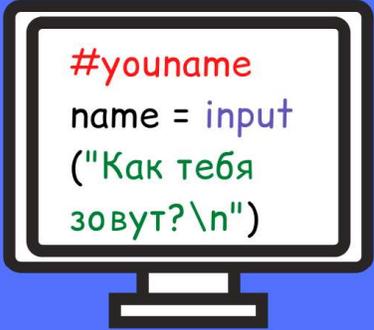
Python

Урок I. Как начать диалог с компьютером?

Как сделать ввод данных?

Давайте напишем программу на Python. В новом, пустом, окне введите две строки кода:

Первая строка называется комментарием. Комментарии начинаются с символа решетки (#), представляют собой заметки программиста или какие-то напоминания, игнорируемые компьютером. В данном примере это лишь заметка. Вторая строка просит пользователя ввести свое имя и запоминает его как name.



```
#youname
name = input
("Как тебя
зовут?\n")
```

Зачем пишут комментарии в программах? Подумайте, как комментирование можно использовать при поиске ошибок в программе.

Команда `input ()` по умолчанию воспринимает входные данные как строку символов.

- Целочисленное значение: `a = int (input())`
- Вещественные числа: `a = float (input())`

Как сделать вывод данных?

Теперь давайте введем третью строку кода:

```
print ("Привет, ", name)
```



Таким образом, осуществляется вывод данных при помощи команды `print(список вывода)`. Обратите внимание, что текст "Привет, " и `name` разделяет запятая.

!!! Проверь, что выведет программа.

```
Задача
a = input()
b = input()
c = a + b
print(c)
```

Протокол:

21

33

2133

Почему?

Из математики мы знаем, что:

$x + y$ - сложение

$x - y$ - вычитание

$x * y$ - умножение

x / y - деление

Давайте напишем программу, которая вычисляет сколько вам лет.

В новом, пустом, окне введите три строки кода:

```
a = int (input ("Какой
сейчас год?\n"))
b = int (input ("Год
вашего рождения?\n"))
print ("Вам ", a - b)
```

Что программа знает о вас?

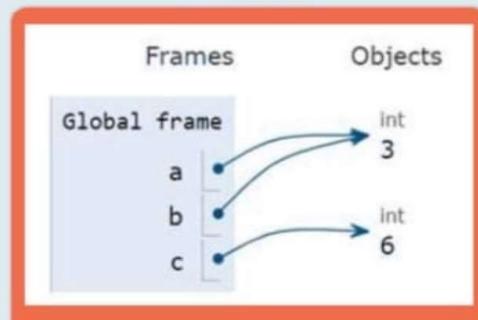
Сделайте так, чтобы программа производила разные арифметические операции.

Урок 2. Что делать, если данных не хватает для реализации программы?

ДЛЯ ЧЕГО В ПРОГРАММЕ ИСПОЛЬЗУЮТ ПЕРЕМЕННЫЕ?

ССЫЛКА НА ПАМЯТЬ В КОМПЬЮТЕРЕ

Переменная в Python – это ссылка на значение, которое хранится в памяти компьютера. Ссылку можно менять на другое значение.



Любые данные в Python это объект, любая переменная — ссылка на объект. Не существует никаких данных, которые не являются объектами.

Для программы:

```
a=3  
b=3  
c=a+b
```

Переменные a и b будут ссылаться на один и тот же объект.

АДРЕС ОБЪЕКТА

Адрес хранения значения можно получить с помощью функции `id()`, где в скобках указывается переменная, которая ссылается на данное значение.

КАК МОЖНО НАЗЫВАТЬ ПЕРЕМЕННЫЕ

В переменных можно использовать латинские буквы (a..z, A..Z), цифры (0..1) и нижнее подчеркивание (`_`). При этом, заглавные и строчные буквы различаются. Название переменной не может начинаться с цифры.

```
>>> a=3  
>>> A=4  
>>> id(a)  
2074073296  
>>> id(A)  
2074073312  
>>> c1=A  
>>> id(c1)  
2074073312
```

ВЫВОДЫ

1. Любые данные в Python это объекты.
2. Если объекты «одинаковы», то они хранятся по одному адресу в памяти. Иными словами, $a == b$ и $id(a) == id(b)$ это эквивалентные утверждения.

ПРОВЕРЬ СЕБЯ

После выполнения программы какое значение и какой адрес в памяти компьютера будет иметь переменная c .

```
a=2
b=3
c=a*b-4
```

если $id(a)=2074073280$
 $id(b)=2074073296$

Загрузите среду программирования Python.

В консоле наберите следующий текст. Есть ли ошибки в названиях переменных? Если есть исправьте их. Чему будет равна переменная C_sum ? Какой адрес имеет объект C_sum в памяти компьютера?

```
>>>1_c=10234
>>>1_c
>>>id(_c)
>>>C_sum=0
>>>C_sum=_c+_c
>>>C_sum
```



Как записывать данные в одной строке?

Иногда нужно вводить несколько значений в одной строке. Попробуйте составить код, чтобы диалог программы с пользователем выглядел так:

Введите два числа:

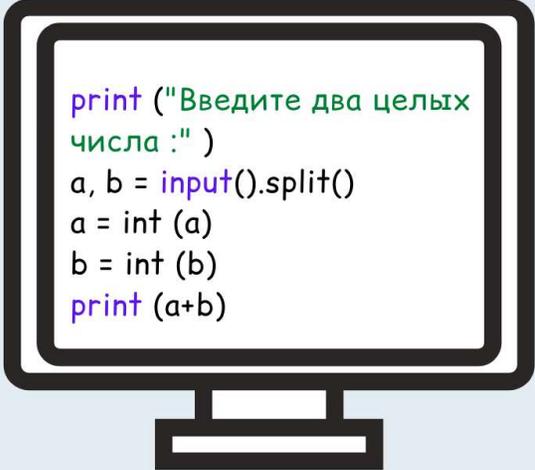
2
3
2+3=5

Чтобы выделить две части, применим функцию `split` (от англ. `split` - расщепить) и, предполагая, что этих частей всего две, запишем их в переменные `a` и `b`: `a, b = input().split()`

Здесь используется так называемое множественное присваивание - в одном операторе присваивания задаются значения двух переменных.

Теперь нужно применить функцию `int` к переменным `a` и `b` - преобразовать строки в целые числа:

```
a = int(a)  
b = int(b)
```



```
print ("Введите два целых  
числа :")  
a, b = input().split()  
a = int (a)  
b = int (b)  
print (a+b)
```

Визуально доработайте программу. Попробуйте использовать три слагаемых.

Обратите внимание!

Количество имён переменных слева от оператора присваивания должно точно соответствовать количеству введённых чисел: если их будет больше или меньше, программа завершится с ошибкой.



Урок 3. Как создать свой собственный калькулятор?

Как правильно записывать арифметические выражения?

Арифметические выражения обычно записываются в одну строку.

Приоритет операций выполняется в следующем порядке:

I.
действия в скобках

II.
умножение и деление,
слева направо

III.
сложение и
вычитание, слева
направо

Язык Python имеет ряд математических (арифметических) операторов для проведения вычислений в арифметических выражениях. Перечень этих операций в порядке убывания приоритета следующий:

1) `**` - возведение в степень;

2) `-x` - унарный минус;

3) `/`, `//` - обычное деление, деление с округлением вниз (одинаковый приоритет);

4) `%` - остаток от деления;

5) `*` - умножение;

6) `-` - вычитание;

7) `+` - сложение.

Какие инструменты используются для обработки целых чисел?

Для целочисленного деления в языке Python используется оператор `//`, для вычисления остатка от деления - оператор `%`.

Воспользуемся возможностями Python для создания чего-то практически полезного: для выполнения вашего домашнего задания!

Напишем короткую программу, в которой используются строки и числа, воспользуемся функцией `eval()`, чтобы превратить математические задачи в ответы. Функция `eval()` может преобразовать строку в число (например, "20" в 20 или "2*10" в 20).

Продемонстрируйте выполнение своего д.з

```
#Попросить пользователя ввести математическую
задачу
problem = input ("Введите задачу: ")
#Выполнить программу, пока пользователь не
введет q
while (problem!= "q"):
    #Показать задачу и ответ с помощью eval()
    print ("Ответ на ", problem, " - это", eval(problem))
    #Запросить еще одну задачу
    problem = input ("Введите еще одну задачу или q,
чтобы выйти: ")
    #Этот цикл будет выполняться пока вы не
введете q для выхода
```



```

x = 1 / 6
print("{:f}".format(x)) # формат f выводит по
                        # умолчанию 6 цифр
                        # в дробной части
print(" {:.3f}".format(x)) # .3 обозначает, что необходимо
                        # вывести 3 знака после точки
print("{:12.4e}".format(x)) # 12.4 - первое число (12)
задает
                        # общее количество позиций на вывод
числа,
                        # второе (4) - количество цифр в дробной
части.
                        # Формат e - выводит число в научном
формате

```

Продемонстрируйте, как работает калькулятор.

При работе с вещественными числами можно использовать уже знакомый нам модуль `math`, который содержит большое число встроенных функций.

`int(x)` - отбрасывание дробной части числа `x`;

`round(x)` - округление вещественного числа `x` к ближайшему целому числу.

Познакомьтесь самостоятельно с функциями, содержащимися в модуле `math`.

Операции с вещественными числами

`int` - целая часть числа

```
x=1.6
print(int(x))
```

→ 1

`round` - ближайшее целое число

```
x=-1.2
print(round(x))
```

→ -1

Какие инструменты используются для обработки вещественных чисел?

Для того, чтобы ввести вещественное число с помощью функции `input()` - необходимо преобразовать символьную строку, которая является результатом работы функции `input()`, в вещественное число с помощью функции `float()`:

```
x = float(input())
```

Если необходимо ввести сразу несколько значений из одной строки, то используем тот же способ, что и для целых чисел:

```
x, y = map(float, input().split())
```

```
Ввод вещественных чисел  
print( "Введите число:" )  
x = float(input())  
или так:  
x = float(input("Введите число:"))
```

При выводе вещественных чисел по умолчанию выводится 16 знаков после запятой. Не всегда такой формат необходим. Если нужно сократить формат вывода, то используется форматный вывод. Для этого применяется метод `format()` к строке, которую мы хотим вывести. А внутри строки записываются форматы в фигурных скобках после двоеточия. В круглых скобках функции `format()`, указываются имена переменных (или константы), значения которых будут подставляться в строку вместо фигурных скобок в указанном формате. Количество переменных должно совпадать с количеством фигурных скобок.



Какие инструменты используются для обработки случайных чисел?

Случайные числа - это последовательность чисел, в которой невозможно предсказать следующее число, даже зная все предыдущие.

Для получения псевдослучайных чисел в заданном диапазоне мы будем использовать функции из модуля `random`:

- `randint(a,b)` - случайное целое число на отрезке $[a; b]$;
- `uniform(a,b)` - случайное вещественное число на отрезке $[a; b]$.

Для того чтобы записать в переменную `n` случайное число на отрезке $[a; b]$, можно использовать такие команды:

```
from random import randint  
n = randint( a, b)
```

В первой строке из модуля `random` импортируется (загружается) функция `randint`, во второй она вызывается для получения случайного числа.

Получите случайное вещественное число на отрезке $[1,5; 2,8]$.



Урок 4. Что делать если?

Как использовать неполную форму условного оператора?

Здесь использован условный оператор в неполной форме, потому что в случае, когда условие ложно, ничего делать не требуется (нет слова `else` и блока операторов после него)

```
if <условие>:  
    <блок операторов>
```

Давай попробуем создать программу, с использованием `if`. Введите следующий программный код в новое окно и сохраните его с именем `OldEnoughOrElse.py`.

```
driving_age = eval (input("Какой  
минимальный возраст для  
получения прав в вашей стране? "))  
your_age = eval (input("Сколько вам  
лет? "))  
if your_age < driving_age:  
    print ("Извините, вы не сможете  
водить еще ", driving_age -  
your_age, "лет.")
```



В нашем случае программа не показывает, что возраст для получения прав подходящий. Как это сделать? Рассмотрим в следующей подтеме.

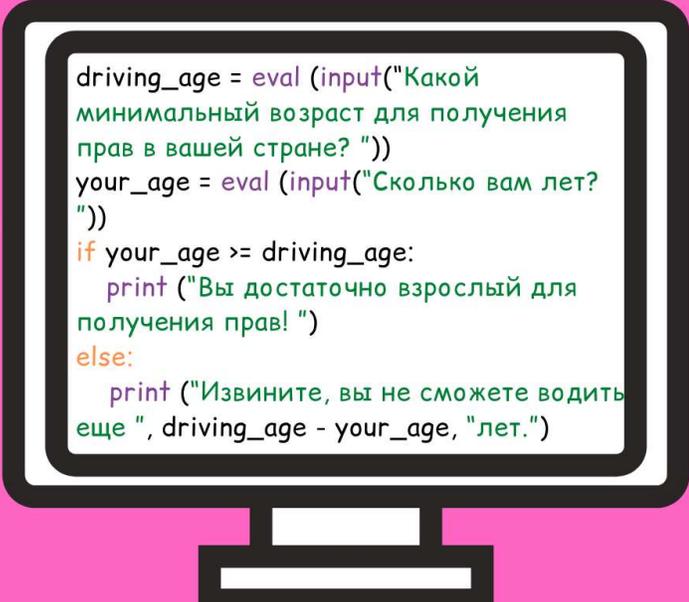
Как правильно записывать условный оператор?

Зачастую нам нужно, чтобы наша программа выполняла определенное действие, если условие расценивается как истинное (True), и какое-то другое действие, если условие является ложным (False). Поэтому в нашем распоряжении есть сокращение: выражение `else`, позволяющее нам проверить истинность условия без необходимости проверять его ложность. Выражение `else` может быть использовано только после выражения `if`, самостоятельное использование выражения `else` является недопустимым. Синтаксис выглядит следующим образом:

```
if условие:  
    табулированные выражения  
else:  
    другие табулированные выражения
```

Давай попробуем создать программу, с использованием `if`. Введите следующий программный код в новое окно и сохраните его с именем `OldEnoughOrElse.py`.

Продемонстрируйте,
как работает
программа.



```
driving_age = eval (input("Какой  
минимальный возраст для получения  
прав в вашей стране? "))  
your_age = eval (input("Сколько вам лет?  
"))  
if your_age >= driving_age:  
    print ("Вы достаточно взрослый для  
получения прав! ")  
else:  
    print ("Извините, вы не сможете водить  
еще ", driving_age - your_age, "лет.")
```

Николай написал функцию `is_alive(health)`, которая проверяет здоровье персонажа в игре. Если оно равно или меньше нуля, то функция возвращает `False`, в противном случае `True`. К сожалению, функция не работает, так как ученик допустил в ней ряд ошибок. Исправьте их и проверьте работоспособность программы (в качестве аргумента всегда передается число).

```
def is_alive(health):  
    if:  
        health < 0  
        False  
    else:  
        return true
```

Как использовать вложенные условные операторы?

Внутри условного оператора могут находиться любые операторы, в том числе и другие условные операторы. Ниже приведен код расширенной версии программы `OldEnoughOrElse.py`, адаптированной для нашего русского языка.

Условный оператор, проверяющий равенство (он выделен шрифтом), находится внутри блока «иначе» (`else`), поэтому он называется вложенным условным оператором. Использование вложенных условных операторов позволяет выбрать один из нескольких (а не только из двух) вариантов.

```
driving_age = eval (input("Какой минимальный возраст для  
получения прав в вашей стране? "))  
your_age = eval (input("Сколько вам лет? "))  
if your_age >= driving_age:  
    print ("Вы достаточно взрослый для получения прав! ")  
else:  
    if driving_age - your_age > 4:  
        print ("Извините, вы не сможете водить еще ", driving_age - your_age,  
            "лет.")  
    if driving_age - your_age == 1:  
        print ("Извините, вы не сможете водить еще ", driving_age - your_age,  
            "год.")  
    if driving_age - your_age <= 4 and driving_age - your_age != 1:  
        print ("Извините, вы не сможете водить еще ", driving_age - your_age,  
            "года.")
```

Урок 5. Как организовать работу находясь в каких-либо условиях?

Как работает операция И?

Иногда единственного выражения ветвления может оказаться недостаточно.

Предположим, что ООО «Неон» набирает сотрудников, возраст которых от 30 до 45 лет включительно. Нужно написать программу, которая запрашивает возраст претендента и выдает ответ: «подходит» он или «не подходит» по этому признаку.

Какое же условие должно быть истинно для того, чтобы человека приняли на работу?

Одного условия «возраст \geq 30» не хватает, это условие соблюдается и для людей старше 45 лет. С другой стороны, условия «возраст \leq 45» тоже не хватает, так как оно выполняется и для школьников. В этой задаче нужно, чтобы два условия выполнялись одновременно: «возраст \geq 30» и «возраст \leq 45».

Операция «И» (and) означает одновременное выполнение двух или нескольких условий.



Ниже приведен код программы с использованием операции И.

```
your_age = eval (input ("Сколько  
вам лет? "))  
if (your_age >= 30 and your_age <=  
45):  
    print("Ваш возраст подходит")  
else:  
    print("Ваш возраст не  
подходит")
```

Решение получилось короткое и понятное. В условном операторе мы записали сложное условие составленное из двух простых с помощью логической операции «И». В языке Python эта операция обозначается словом `and`.

Как работает операция ИЛИ?

Рассмотрим ещё одну задачу. Самолёт из Москвы в Красноярск летает только по понедельникам и четвергам. В переменной `day` записан номер дня недели (1 - понедельник, 7 - воскресенье). Программа должна определить, полетит ли самолёт в этот день.



Если мы напишем условие «day = 1 and day = 4», то это будет неверно, потому что мы потребовали, чтобы значение переменной day было одновременно равно и 1, и 4. Такого быть не может, поэтому это условие всегда будет ложно. Значит, операция «И» не подходит. Вместо неё нужно применить другую операцию – «ИЛИ», которая требует выполнения хотя бы одного из набора условий.

Операция «ИЛИ» означает выполнение хотя бы одного из двух или нескольких условий. В языке Python операция «ИЛИ» обозначается словом «or» (по-английски – «или»).

Ниже приведен код программы с использованием операции ИЛИ.

```
day = eval (input ("Какой сейчас
день?(1-7) "))
if day == 1 or day == 4:
    print( "Самолет полети в этот
день!" )
else:
    print( "Нет рейса." )
```

Напишите
другой
вариант
решения
последней
задачи,
использую
щий
операцию
«И».

Как работает операция НЕ?

Существует ещё одна операция, которую можно использовать в сложных условиях - «НЕ», в Python она обозначается словом «not» (по-английски - «не»).

Если исходное условие истинно, то обратное (противоположное) ему - ложно, и наоборот.

Например, решение задачи с самолётом можно было записать так:

```
day = eval (input ("Какой
сейчас день?(1-7) "))
if not (day == 1 or day == 4):
    print( "Нет рейса.")
else:
    print("Самолет полети в
этот день!")
```

Используя операцию «НЕ», можно записывать условия по-разному, как нам удобнее в каждом случае.

Запишите равносильные условия, не используя операцию «НЕ»:

- а) `not (7 < a and a < 12)`
- б) `not (b != c or d < 5)`



Как правильно записывать порядок выполнения операций?

Операции «И», «ИЛИ» и «НЕ» - это логические операции, то есть они применяются к логическим значениям: «да»/«нет», «истина»/«ложь».

Если в сложном условии встречается несколько разных операций, они выполняются в следующем порядке (во всех случаях - слева направо):

- 1) операции в скобках;
- 2) операции «НЕ»;
- 3) операции «И»;
- 4) операции «ИЛИ».

Изменить порядок действий можно с помощью круглых скобок.

**Определите порядок операций при определении истинности условия:
 $\text{not } (a > 10) \text{ or not } (a < 10) \text{ and } (a < b)$
Определите, истинно или ложно это выражения при $a = 5, b = 10$.**



Урок 6. Бесконечность – это предел?

Как организовать цикл?

Цикл – это многократное выполнение одинаковых действий.

Два вида циклов:

- цикл с **известным** числом шагов (сделать 10 раз)
- цикл с **неизвестным** числом шагов (делать, пока не надоест)

Задача. Вывести на экран 10 раз слово «Привет».



Можно ли решить известными методами?

Допустим, мы хотим вывести 10 раз на экран слово «привет». Можно, конечно, записать 5 одинаковых команд:

```
print( "привет" )  
print( "привет" )  
print( "привет" )  
print( "привет" )  
print( "привет" )
```

```
print( "Привет" )  
print( "Привет" )  
...  
print( "Привет" )
```



Что плохо?



Но что если нужно будет сделать какие-то действия 10000 или 1000000 раз? Давайте разберёмся, как можно организовать цикл в языке Python. Вы знаете, что программа выполняется автоматически. И при этом в любой момент нужно знать, сколько раз уже выполнен цикл и сколько ещё осталось выполнить. Для этого необходимо использовать ячейку памяти (переменную). В ней можно, например, запоминать, сколько раз цикл уже был выполнен. Такую переменную целого типа часто называется счётчиком.

```
счётчик = 0
пока счётчик < 10:
    print("Привет")
    увеличить счётчик на 1
```

результат операции автоматически сравнивается с нулём!

```
счётчик = 10
пока счётчик > 0:
    print("Привет")
    уменьшить счётчик на 1
```

Какой способ удобнее для процессора?

```
count = 0
while count < 10: #
заголовок цикла
    print("привет" )
    count += 1 #
увеличение счётчика
```

Сначала в переменную-счётчик записывают ноль (цикл ещё не выполнен ни разу), а после каждого повторения цикла увеличивают значение счётчика на единицу. Давай попробуем создать программу, с использованием переменной-счётчик записать ноль (цикл ещё не выполнен ни разу), а после каждого повторения цикла увеличить значение счётчика на единицу.

В этой программе используется новое служебное слово `while`, после которого записано условие.

Поскольку цикл связан с повторением, циклические алгоритмы называют итерационными (от лат. *iteratio* - повторение). Каждое выполнение тела цикла называют итерацией.

Все операторы, которые выполняются в цикле (они называются телом цикла) сдвигаются вправо на одинаковое число позиций, так же как и в условном операторе. Этот приём позволяет обойтись без операторных скобок, ограничивающих тело цикла в других языках программирования.

```
print( "привет" )
    count += 1 # увеличение счётчика
```

Нам нужно выполнять цикл 10 раз, то есть пока счётчик не станет равен 10. Об этом говорит заголовок цикла

```
while count < 10:
```

Его можно прочитать как «делай, пока count < 10». После каждой итерации цикла переменная count увеличивается на 1 – цикл выполнен ещё один раз. Если программист забудет написать этот оператор, произойдёт заикливание: программа никогда не остановится, потому что условие count < 10 никогда не станет ложным.

Цикл можно построить и по-другому: сразу записать в счётчик нужное количество итераций, и после каждой итерации цикла уменьшать счётчик на 1. Тогда цикл должен закончиться при нулевом значении счётчика. Запишите цикл со счётчиком, который уменьшается от нужного значения до нуля. Этот вариант несколько лучше, чем предыдущий, поскольку счётчик сравнивается с нулём, а такое сравнение выполняется в процессоре автоматически.

Найдите ошибку в программе:

```
k = 0
while k < 10:
    print( "привет" )
```

Как её можно исправить?

Как работает цикл с условием?

Цикл, в котором проверка условия выполняется при входе (перед выполнением очередного шага) называется циклом с предусловием, то есть циклом с предварительной проверкой условия.

Перед тем, как начать выполнение цикла, мы проверяем, нужно ли это делать вообще. Это можно сравнить с такой ситуацией: перед тем, как прыгнуть в бассейн, нужно проверить, есть ли в нём вода.

Все циклы, записанные в начале параграфа - это циклы с предусловием. У них есть два важных свойства:

- цикл не выполнится ни разу, если условие в самом начале ложно;
- как только нарушается условие в заголовке цикла, его работа заканчивается.

Рассмотрим ещё одну задачу, которая решается с помощью цикла с условием. Требуется ввести с клавиатуры натуральное число и найти сумму цифр его десятичной записи. Например, если ввели число 123, программа должна вывести сумму $1+2+3 = 6$.

```
N = int( input("Введите число: ") )
summa = 0
while N != 0:
    digit = N % 10
    summa += digit
    N = N // 10
print( "Сумма цифр", summa )
```

Сколько раз выполнится цикл, если ввести однозначное число?
двузначное? К-значное? число 0?
Какова может быть сумма цифр двузначного числа?
трёхзначного? К-значного?



Предположим, что число записано в переменной N . Нам нужно как-то разбить число на отдельные цифры.

Вспомним, что остаток от деления числа на 10 равен последней цифре его десятичной записи. Запишем эту цифру в переменную $digit$: $digit = N \% 10$

Сумму цифр будем хранить в целой переменной $summa$. В самом начале, пока ни одну цифру ещё не обработали, значение этой переменной равно нулю:
 $summa = 0$

Для того чтобы добавить к предыдущей сумме новую цифру, нужно заменить значение переменной $summa$ на $summa + digit$, то есть выполнить присваивание $summa += digit$

Для того чтобы затем отсечь последнюю цифру числа N , разделим N на 10 (основание системы счисления): $N = N // 10$

Эти три операции - выделение последней цифры числа, увеличение суммы и отсечение последней цифры - нужно выполнять несколько раз, пока все цифры не будут обработаны (и отсечены!) и в переменной N не останется ноль.

Для введённого числа 123 программа должна выдать ответ 6 (последнее значение переменной sum). Это правильный ответ.

В отличие от предыдущего примера, здесь количество шагов цикла заранее неизвестно, оно определяется количеством цифр введённого числа.



Как работает алгоритм Евклида?

Алгоритм Евклида позволяет найти наибольший общий делитель (НОД) двух натуральных чисел.

Вспомните, что такое НОД?

Алгоритм нахождения НОД делением

1. Больше число делим на меньшее.
2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла).
3. Если есть остаток, то большее число заменяем на остаток от деления.
4. Переходим к пункту 1.

Пример: Найти НОД для 30 и 18.

1. $30 / 18 = 1$ (остаток 12)
2. $18 / 12 = 1$ (остаток 6)
3. $12 / 6 = 2$ (остаток 0)
4. Конец: НОД - это делитель 6.
5. НОД (30, 18) = 6

Запишем программу НОД для 50 и 130

```
a = 50
b = 130

while a != 0 and b != 0:
    if a > b:
        a = a % b
    else:
        b = b % a
print(a + b)
```

Остаётся вывести результат - ненулевое значение переменной a или b . Количество итераций такого цикла зависит от исходных данных. Дополним программу так, чтобы она считала ещё и количество сделанных итераций.





Для этого нужно ввести переменную-счётчик целого типа. Перед началом цикла счётчик обнуляется (в него записывается ноль), и после каждой итерации значение счётчика увеличивается на единицу:

```
count = 0
while a != 0 and b != 0:
...
    count += 1
    print( a + b )
    print( "Шагов:", count )
```

Вместо многоточия нужно вставить условный оператор, как и в предыдущем варианте программы. Обратите внимание, что оператор, увеличивающий значение счётчика, записывается с отступом – он находится в теле цикла. Напишите программу, которая получает с клавиатуры два натуральных числа и находит их НОД с помощью алгоритма Евклида. Программа должна подсчитать количество шагов цикла.

Что такое обработка потока данных?

На вход программы поступает поток данных – последовательность целых чисел, которая заканчивается нулём.

Требуется найти сумму элементов этой последовательности.

В этой задаче не нужно сохранять все данные в памяти, мы можем добавлять их к сумме по одному. Используем две целых переменных: в переменной *x* будем хранить последнее введённое число, а в переменной *summa* – накапливать сумму.

- Какое начальное значение нужно присвоить переменной *summa*?
- Как добавить к значению переменной *summa* значение переменной *x*?



Сначала запишем
основной цикл
программ, «скрыв»
шаги алгоритма в
комментариях:

```
while x != 0:  
    # добавить x к  
    сумме  
    # прочитать  
    следующее число
```

Однако перед таким циклом нужно
прочитать первое число, иначе неясно,
откуда возьмётся значение x при первой
проверке условия. В итоге получается
такая программа:

```
summa = 0  
x = int( input() )  
while x != 0:  
summa += x  
    x = int( input() )  
print( "Сумма", summa )
```

Как нужно
изменить
программу для
того, чтобы она
вычисляла сумму
только
положительных
чисел?

Урок 7. Это весело, сможешь повторить пару раз?

Как повторить действие N раз?

Вернёмся снова к задаче, которую мы уже обсуждали - вывести на экран несколько раз слово «привет». Фактически нам нужно организовать цикл, в котором блок операторов выполнится заданное число. Для этого можно применить ещё один вид цикла - цикл по переменной (или цикл с параметром).

На языке Python он записывается так:



Здесь слово `for` означает «для», переменная `i` (её называют переменной цикла) изменяется в диапазоне (`in range`) от 0 до 10, не включая 10 (то есть от 0 до 9 включительно).

Таким образом, цикл выполняется для `i = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9` - ровно 10 раз.





Переменная i - это счётчик выполненных итераций цикла.

Можно было записать этот цикл и по-другому:

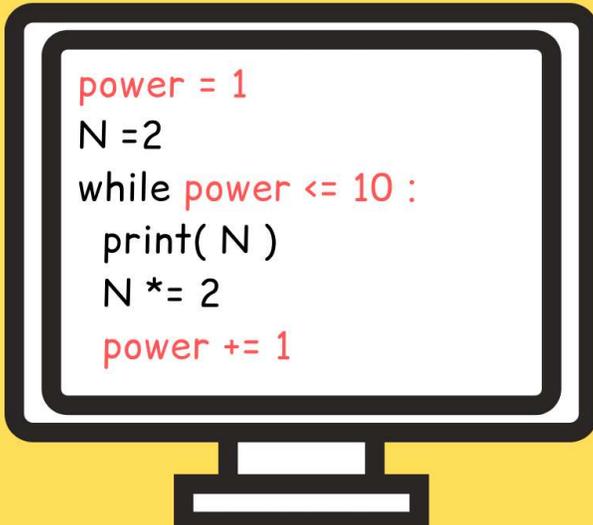
```
for i in [0,1,2,3,4,5,6,7,8,9]:  
    print( "привет" )
```

В квадратных скобках через запятую перечислены все значения переменной, при которых выполняется цикл. Если их много, такой способ неудобен, лучше использовать встроенную функцию `range`.

Обратите внимание, что последовательность, которую строит функция `range`, не бесконечна, то есть цикл с переменной всегда заканчивается, программа не может зациклиться.

Рассмотрим ещё один пример.

В информатике важную роль играют степени числа 2 (2, 4, 8, 16 и т.д.). Давайте выведем на экран все степени двойки от 21 до 210. Для решения этой задачи мы можем написать программу, использующую цикл с условием:



```
power = 1  
N = 2  
while power <= 10 :  
    print( N )  
    N *= 2  
    power += 1
```

Сколько раз используется переменная? (см. блоки, выделенные красным цветом): в операторе присваивания начального значения, в условии выполнения цикла и в теле цикла (увеличение на 1).



Чтобы собрать все действия с переменной `power` в один оператор, применим цикл по переменной. Нам нужно выполнить тело цикла при `power = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10`. Чтобы получить такой набор значений, нужно вызвать функцию `range` с двумя аргументами:

1. первый - это начальное значение (1),
2. второй - ограничитель, не входящий в последовательность (11):

```
N = 2
for power in range(1,11):
    print( N )
    N *= 2
```

Запись цикла получилась проще, и поэтому меньше шансов сделать ошибку.

- **Однако** не любой цикл с условием может быть переписан как цикл с переменной. Если количество повторений цикла неизвестно и не может быть найдено заранее (как в задаче с вычислением суммы цифр числа), цикл по переменной использовать не удаётся.
- **С другой стороны**, любой цикл по переменной можно заменить на равносильный цикл с условием: вместо вызова функции `range` придётся задать отдельно начальное значение переменной цикла, условие продолжения цикла и правило изменения переменной цикла.



Рассмотрим ещё одну задачу - найдём сумму всех натуральных чисел от 1 до 1000. Для накопления суммы будем использовать переменную, которую назовём `summa`. В цикле другая переменная (скажем, `i`) изменяется от 1 до 1000, и на каждом шаге этого цикла к сумме добавляется очередное значение `i`:

```
summa = 0
for i in
range(1,1001):
    summa += i
```

Запишите циклы, с помощью которых можно вычислить:

- сумму целых чисел от `a` до `b` ($a \leq b$);
- сумму квадратов целых чисел от `a` до `b` ($a \leq b$).

Что такое шаг изменения переменной цикла?

По умолчанию функция `range` строит последовательность, в которой каждое следующее число на 1 больше предыдущего. Но это правило можно изменить, если при вызове функции `range` указать третий аргумент - шаг изменения переменной цикла.

Следующая программа печатает квадраты натуральных чисел от 10 до 1 в порядке убывания:

В этом примере шаг равен -1 , то есть каждое следующее число на 1 меньше предыдущего.

Заметим, что конечное значение 0 не входит в последовательность.

```
for k in range(10,0,-1):  
    print( k*k )
```

```
for i in range(0, 101, 5):  
    ... # что-то делать с i
```

Пусть, например, нам нужно перебрать в цикле все значения переменной i от 0 до 100, кратные пяти: 0, 5, 10, ..., 100. Для этого нужно взять шаг изменения переменной 5:

Второй аргумент функции `range` равен 101 для того, чтобы последнее значение переменной i было равно 100. Значение-ограничитель должно быть больше, чем 100 (чтобы число 100 появилось в последовательности), но меньше, чем 106 (чтобы следующее число, 105, не появилось).

Урок 8. Как хранятся сложные данные?

Как хранить группу данных?

Данные можно хранить в неупорядоченном виде (например, множество натуральных чисел от 1 до 100, или русский алфавит).

Множество каких данных иллюстрировано на картинке?

Январь февраль март
апрель май июнь
Июль август сентябрь
октябрь ноябрь декабрь

Данные можно хранить в упорядоченном виде

Данные	a	b	c	d	e	f
Номер данных в группе	0	1	2	3	4	5

Упорядоченное хранение группы данных в рамках одной переменной в Питоне называются списками.

a=list() - резервирование места a под список



Как создать список?

Чтобы использовать списки, их нужно сначала создать. Сделать это можно несколькими способами.

Способ 1

Обработать любой итерируемый объект (например, строку) - объект, к элементу которого также можно обратиться по индексу.

```
#1 способ
...
>>> a=list('список')
>>> a
['с', 'п', 'и', 'с', 'о', 'к']
>>>
```

! Попробуйте создать список, содержащий буквы Ш К О Л А

Способ 2

Явно задать список

```
#2 способ
>>> a=[1,2,3]
>>> a
[1, 2, 3]
>>>
```

! Попробуйте создать список, содержащий названия времен года (подсказка, не забывайте про формат хранения текстового типа данных)

Способ 3

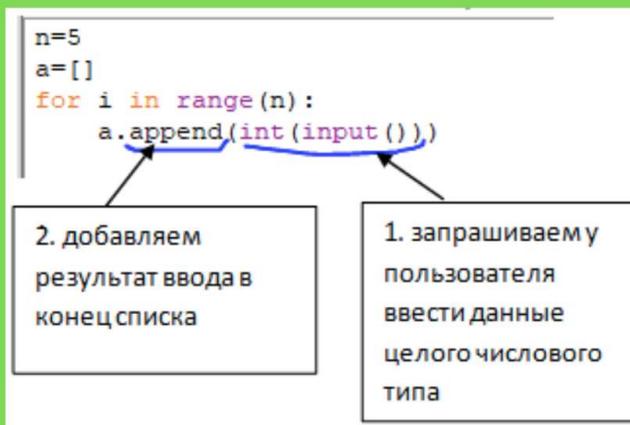
Формирование списка поэтапно в программе

```
#3 способ
>>> #задаем пустой список
>>> a=[]
>>> #добавляем в конец списка элемент
>>> a.append(2)
>>> #добавляем еще один элемент в конец списка
>>> a.append(4)
>>> a
[2, 4]
>>>
```

append - это метод добавления данных в конец списка. Данные можно добавлять только в конец списка.



Чтобы организовать ввод данных пользователем с клавиатуры, необходимо использовать метод `append` в цикле. Цикл будет повторяться столько раз, сколько вы хотите спросить данных у пользователя, т.е. на каждом новом витке цикла в конец списка будет добавляться новый элемент.



Попробуйте запустить фрагмент кода. Проверьте, что хранится в переменной `a`.

Задание: напишите программу, которая формирует список букв, введенных пользователем

Как обратиться к элементу списка?

A	[0]	=	10
Имя списка	Номер элемента в списке		Значение элемента - данные

```
>>> a=[2,4,6,8]
>>> a[0]
2
>>> a[1]
4
>>> a[2]
6
>>> a[3]
8
```

Так как список - это упорядоченное множество элементов, то у каждого элемента есть свой индекс. Индексация в любом списке начинается с 0. Индекс указывается в квадратных скобках к переменной, в которой хранится весь список, например, `a[0]` - обращение к первому элементу.

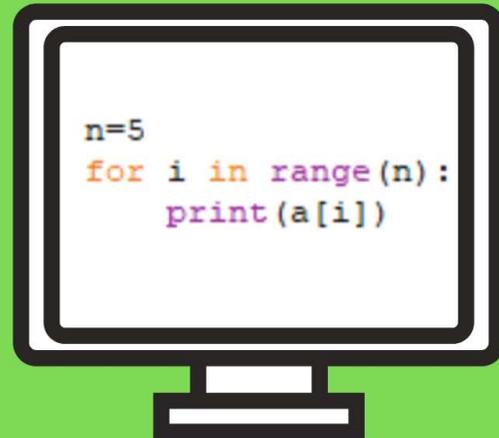
Какое будет последнее значение индекса в списке размером 10?



Как осуществить перебор элементов списка?

Список можно перебрать поэлементно в цикле for

Соедините данный фрагмент программы с фрагментом ввода данных в список с клавиатуры. Проверьте, что получится для серии данных: 1, 2, 3, 4, 5.



Как работают генераторы списков?

Еще один способ задать список - это генераторы списков - способ построить новый список, применяя выражение к каждому элементу последовательности. Генераторы очень похожи на цикл for.

```
>>> c=[i for i in 'список']
>>> c
['с', 'п', 'и', 'с', 'о', 'к']
```

Генерация списка с через i , где i лежит в итерируемом объекте 'список'

Над генерируемыми элементами можно производить операции

```
>>> c=[i*3 for i in 'список']
>>> c
['ссс', 'ппп', 'иии', 'ссс', 'ооо', 'ккк']
>>>

>>> a=[i*i-5 for i in range(5)]
>>> a
[-5, -4, -1, 4, 11]
>>>
```

Сгенерируйте список значений выражения $5*x*x+2*x-9$ в диапазоне от -5 до 5.

Как заполнить список случайными числами?

Чтобы сгенерировать случайное целое число, надо вызвать команду `randint()`, где в скобках указывается в пределах какого диапазона происходит генерация числа. Например, чтобы получить случайное число от 0 до 10, необходимо вызвать команду: `randint(0,10)`

Но

Команда `randint()` хранится в модуле `random`

```
import random
print(random.randint(0,10))
```

или так:

```
from random import randint
print(randint(0,10))
```

Сравните два способа. Чем они отличаются?

Для того, чтобы сгенерировать список случайным образом, необходимо также, как при вводе данных через клавиатуру, организовать цикл, а вместо команды `input()` использовать команду `randint()`.

```
from random import randint
n=5
a=[]
for i in range(n):
    a.append(randint(0,100))
```



Добавим к программе еще фрагмент кода:
For I in range(n):
Print(a[i])

Проверьте, что получится.

ПРОВЕРЬ СЕБЯ

- Мы забыли ответить на вопрос, как осуществить вывод списка на экран, но в процессе изучения списков, мы постоянно это делали. Напиши фрагмент кода, как вывести список на экран.
- Что выведет программа на экран. Проверь свой ответ, запустив предложенный код:

```
a=[1,2]
for i in range(3):
    a.append(i)
for i in range(5):
    print(a[i])
```



Урок 9. Спутник М постоянно шлет данные. Что с ними делать?

Спутник М каждый день присылает данные об объеме данных, обработанных за сутки (целое число в Терабайтах).

Как найти общее количество данных, обработанных за неделю?

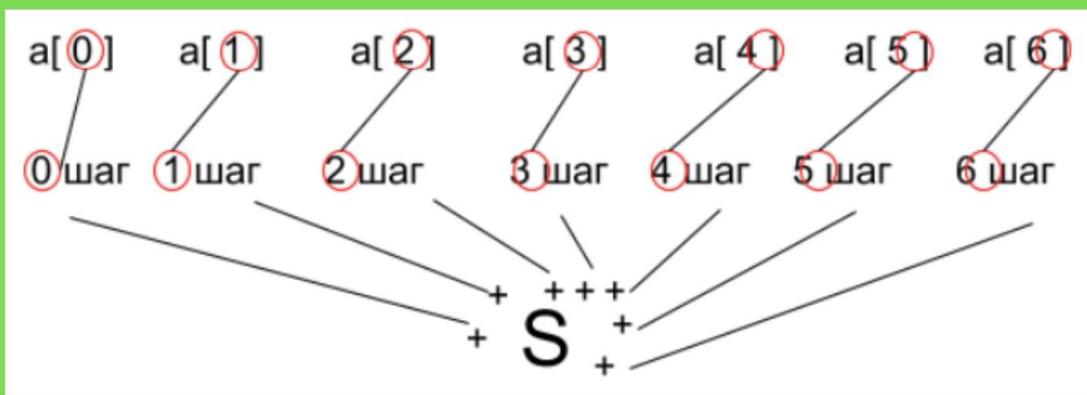
Будем считать, что данные каждый день записываются в список, т.е. у нас есть список, в котором записаны 7 целых чисел.

a	0	1	2	3	4	5	6
	20	10	30	40	10	20	25

Запишем эти данные в список:

```
n = 7 #количество дней
a = [ ]
for i in range(n):
    a.append(int(input()))
```

Для нахождения суммы необходимо завести «корзинку», в которую поочередно мы будем класть элементы списка.

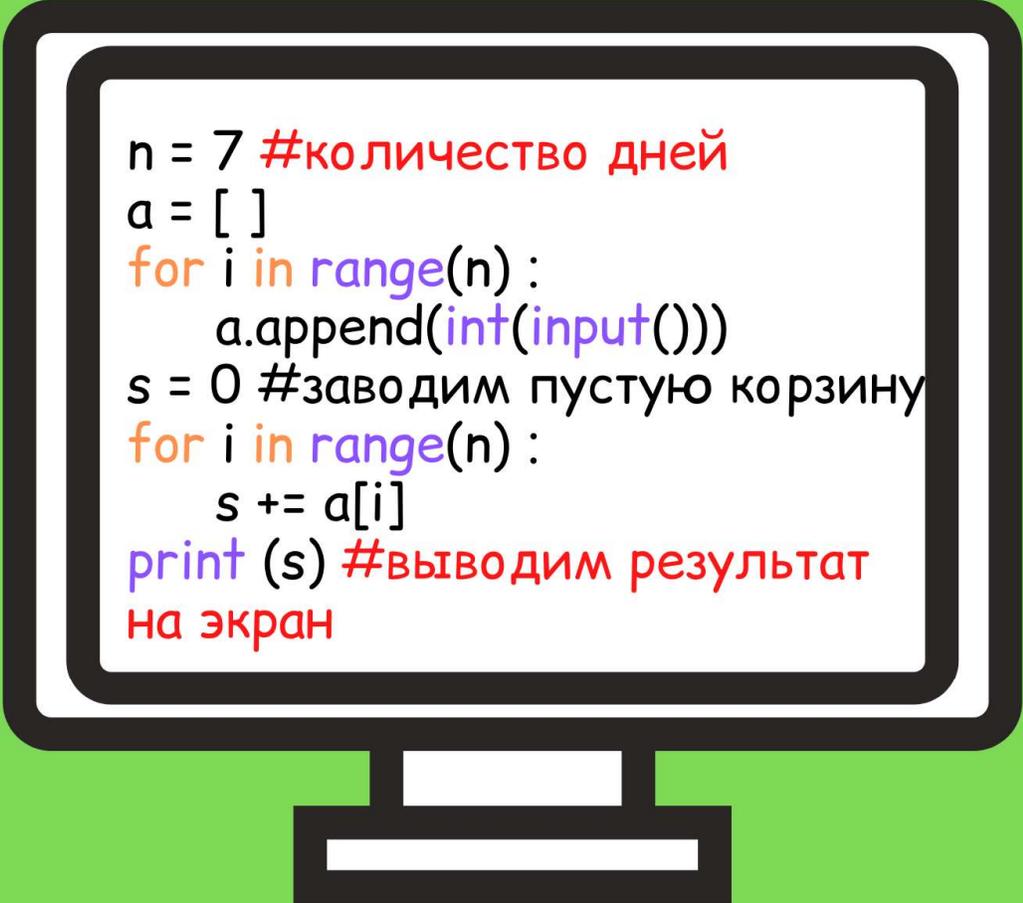


Обратите внимание, что шаг совпадает с номером элемента в списке.

Тогда в общем можно сказать, что на каком-то i -ом шаге мы кладем в «корзинку» i -ый элемент. Шаги можно отсчитывать в цикле `for`:

```
n = 7 #количество дней
for i in range(n):
    s += a[i]
```

Не забывайте, что до начала подсчета суммы «корзина» должна быть пуста. Тогда полный текст программы будет следующий:



```
n = 7 #количество дней
a = [ ]
for i in range(n) :
    a.append(int(input()))
s = 0 #заводим пустую корзину
for i in range(n) :
    s += a[i]
print (s) #выводим результат
на экран
```

Задача

Измените программу на нахождение среднего количества данных, обработанных за месяц (30 дней). Данные занесите случайным образом в диапазоне от 10 до 50 Тб.

Спутник М высылает данные, но не все корректны. Как отобрать корректные данные по условию, присланные за неделю?

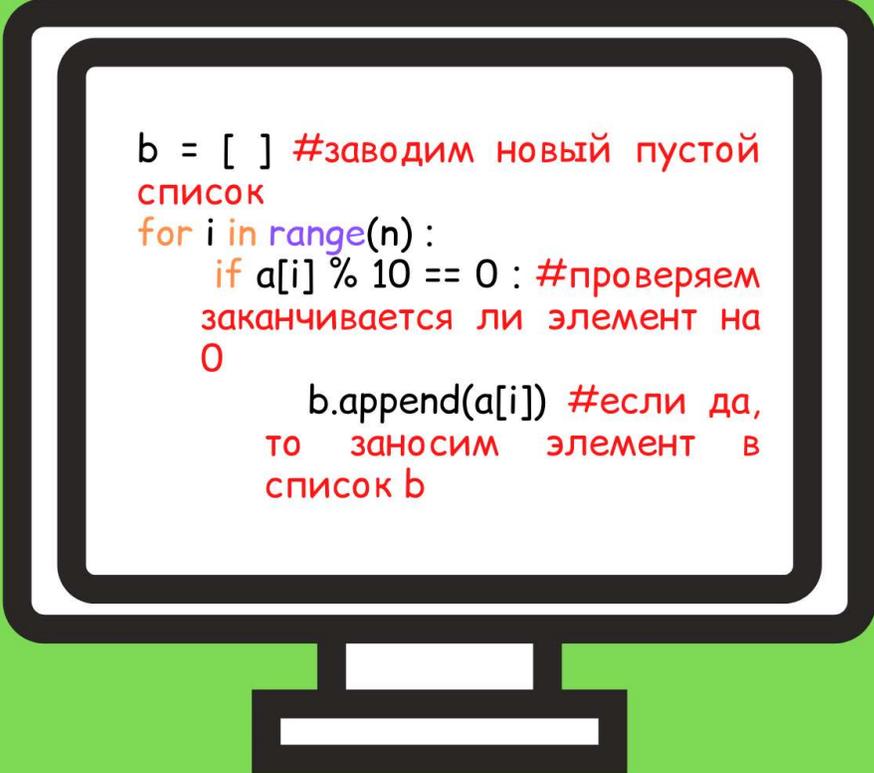
Будем считать, что данные каждый день записываются в список, т.е. у нас есть список, в котором записаны 7 целых чисел.

a	0	1	2	3	4	5	6
	20	10	30	40	10	20	25

Запишем эти данные в список:

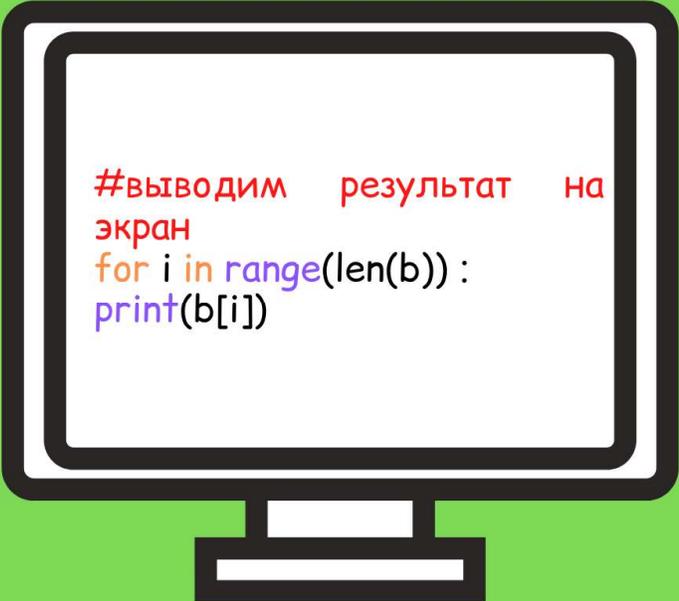
```
n = 7 #количество дней
a = []
for i in range(n):
    a.append(int(input()))
```

Будем считать корректными данными те, которые заканчиваются на 0, которые также надо занести в новый список.



```
b = [ ] #заводим новый пустой
СПИСОК
for i in range(n):
    if a[i] % 10 == 0 : #проверяем
        заканчивается ли элемент на
        0
        b.append(a[i]) #если да,
        то заносим элемент в
        список b
```

Чтобы вывести новый список, будем использовать также цикл `for`, но мы не знаем длину нового списка `b`. Чтобы узнать длину списка, можно использовать функцию `len()`:



```
#выводим результат на
экран
for i in range(len(b)):
    print(b[i])
```

Задача
Измените программу на нахождение нечетных данных, обработанных за месяц (30 дней). Данные занесите случайным образом в диапазоне от 10 до 50 Тб.

Спутник М каждый день присылает данные об объеме данных, обработанных за сутки (целое число в Терабайтах).

Как найти день, в который было обработано максимальное количество данных, присланных за неделю?

Будем считать, что данные каждый день записываются в список, т.е. у нас есть список, в котором записаны 7 целых чисел.

a	0	1	2	3	4	5	6
	20	10	30	40	10	20	25

Запишем эти данные в список:

```
n = 7 #количество дней
a = [ ]
for i in range(n):
    a.append(int(input()))
```

Чтобы найти максимальный элемент, необходимо поочередно сравнивать каждый элемент с эталонным максимумом. Например, говорим, пусть первый элемент - максимальный, и далее сравниваем каждый элемент списка с этим эталоном. Если текущий элемент оказался больше эталона, то заменяем это эталон на новое значение текущего.

```
m = a[0] #заводим эталонный
максимум, равный первому элементу
for i in range(1,n) : #нет
необходимости проверять 1ый
элемент на max, поэтому начинаем
работать со 2ым
    if a[i] > m : #проверяем текущий
элемент, больше ли он m
        m = a[i] #если да, то
заменяем m на a[i]
```

Но кроме поиска максимального количества данных, в задаче просят вывести день недели, в который было обработано это количество, поэтому в полученный цикл необходимо внести запоминание номера текущего максимума. А до цикла указать, что за эталон мы взяли первый день.



```
day = 1
m = a[0] #заводим эталонный
максимум, равный первому элементу
списка
for i in range(1,n) : #нет необходимости
проверять 1ый элемент на max, поэтому
начинаем работать со 2ым
    if a[i] > m : #проверяем текущий
элемент, больше ли он m
        m = a[i] #если да, то заменяем
m на a[i]
    day = i + 1
```

Т.к. дни обычно считают натуральными числами, а индексация списка начинается с 0, поэтому при обновлении переменной day, мы добавляем к ней 1. Т.е. a[0] отвечает за первый день, a[1] - за второй и т.д.



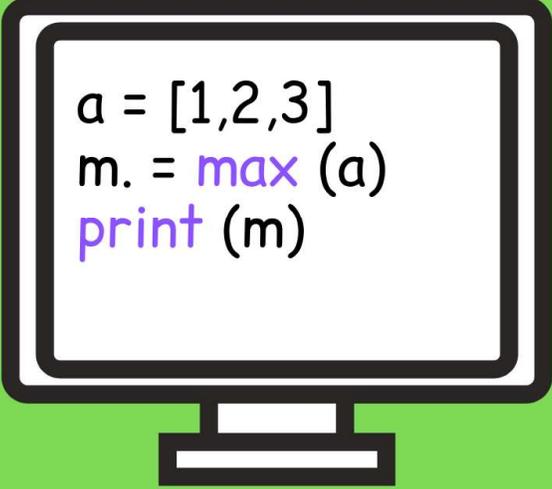


Полный текст программы выглядит следующим образом:

```
n = 7 #количество дней
a = [ ]
for i in range(n) :
    a.append(int(input()))
day = 1
m = a[0] #заводим эталонный максимум, равный
первому элементу списка
for i in range(1,n) : #нет необходимости проверять
1ый элемент на max, поэтому начинаем работать со
2ым
    if a[i] > m : #проверяем текущий элемент,
    больше ли он m
        m = a[i] #если да, то заменяем m на a[i]
        day = i + 1
print (day,m) #заводим номер дня и кол-во данных
(max), обработанных в этот день
```

Если бы в задаче не было условия указать день, а найти только максимальное количество данных, обработанных за день в течение недели, то можно было использовать функцию max().

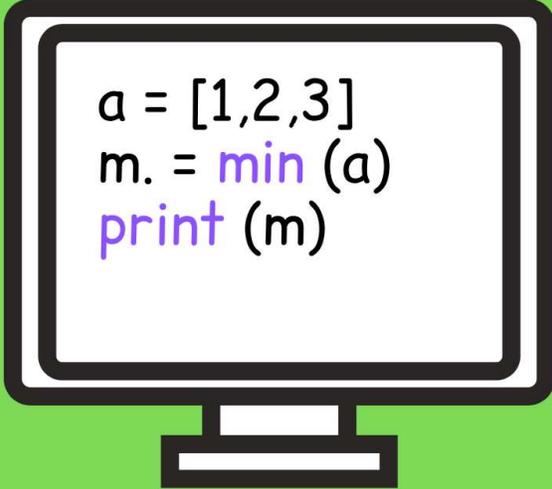




```
a = [1,2,3]
m. = max (a)
print (m)
```

Проверьте, как
работает эта
программа.

Аналогично для
поиска
минимума
можно
использовать
функцию min().
Проверьте, как
работает эта
программа.



```
a = [1,2,3]
m. = min (a)
print (m)
```

Задача

Спутник М в течение недели присылает данные
об объеме данных, обработанных за сутки.

Найдите сколько дней из них спутник
обработывал минимальное количество данных?



Список публикаций и наград, полученные автором за период выполнения исследовательской работы по теме выпускной квалификационной работы

1) «Обучение школьников программированию на основе вопросно-задачного подхода» // VII междунар. науч.-практ. конференция «АПАК» (в печати)

2) «Достоинства перевёрнутых учебников в дистанционном обучении школьников» // Всероссийская с междунар. участием науч.-практ. конференция студентов, аспирантов и молодых ученых



КРАСНОЯРСКИЙ
ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ
ИМ. В. П. АСТАФЬЕВА

XXII Международный
научно-практический
форум студентов, аспирантов
и молодых учёных
Молодёжь и наука XXI века

СЕРТИФИКАТ

удостоверяет, что

Никитина Лидия Викторовна

принимал(а) участие в работе секции «Актуальные проблемы информатики и информационных технологий в образовании» конференции форума «Образование и наука в XXI веке: физика, информатика и технология в смарт-мире»

РУКОВОДИТЕЛЬ СЕКЦИИ

КРАСНОЯРСК, 2021



П.С. Ломаско

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ



КРАСНОЯРСКИЙ
ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ
УНИВЕРСИТЕТ
ИМ. В. П. АСТАФЬЕВА

XXII Международный
научно-практический
форум студентов, аспирантов
и молодых учёных
Молодёжь и наука XXI века

ДИПЛОМ

победителя в номинации

«Самая оригинальная идея»

присуждается

Никитиной Лидии Викторовне

по итогам конкурса исследовательских работ секции
«Актуальные проблемы информатики и информационных
технологий в образовании»,

проходившего в рамках Всероссийской с международным участием
научно-практической конференции студентов, аспирантов
и молодых ученых «Образование и наука в XXI веке: физика,
информатика и технология в смарт-мире»

ПРОРЕКТОР ПО НАУЧНОЙ РАБОТЕ
И ВНЕШНЕМУ ВЗАИМОДЕЙСТВИЮ



С.В. БУТАКОВ

КРАСНОЯРСК, 2021

Министерство науки и высшего образования РФ



Сибирский государственный
университет науки и технологий
имени академика М.Ф. Решетнева

ДИПЛОМ

Награждается

Никитина Лидия Викторовна

студентка группы DO-Б16Б-01

Института математики, физики и информатики
ФГБОУ ВО «Красноярский государственный педагогический
университет имени В.П. Астафьева»

за II место

по секции «Инновационные технологии в современном образовании»
VII Международной научно-практической конференции
«Актуальные проблемы авиации и космонавтики»,
посвященной Дню космонавтики и 60-летию полета Ю.А. Гагарина

Ректор
СибГУ им. М.Ф. Решетнева



Э. Ш. Акбулатов

г. Красноярск

12-16 апреля 2021 г.