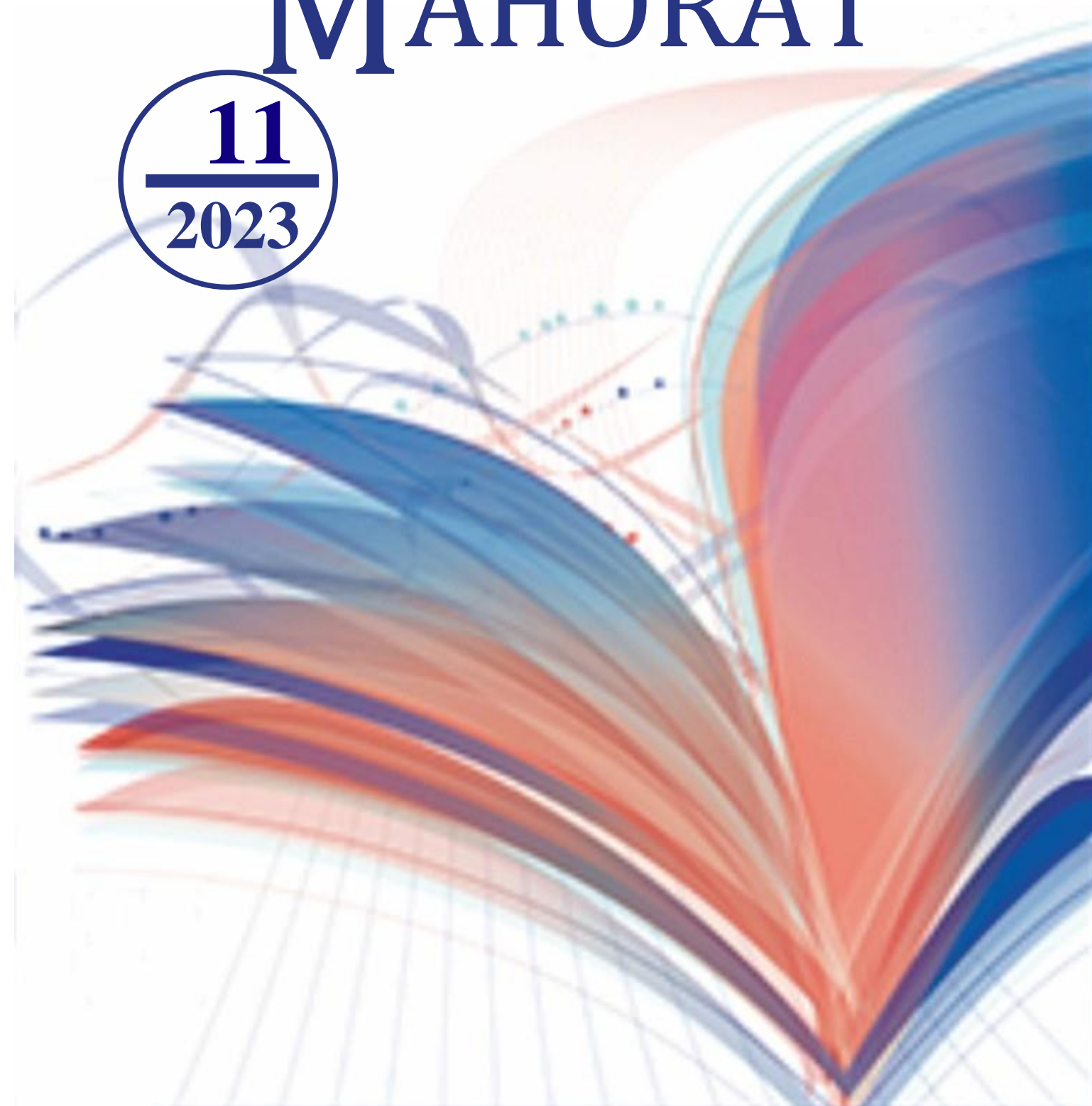


PEDAGOGIK MAHORAT

11
2023



PEDAGOGIK MAHORAT

Ilmiy-nazariy va metodik jurnal

2023, № 1

Jurnal O‘zbekiston Respublikasi Vazirlar Mahkamasi huzuridagi OAK Rayosatining 2016-yil 29-dekabrda qarori bilan **pedagogika** va **psixologiya** fanlari bo‘yicha dissertatsiya ishlari natijalari yuzasidan ilmiy maqolalar chop etilishi lozim bo‘lgan zaruriiy nashrlar ro‘yxatiga kiritilgan.

Jurnal 2001-yilda tashkil etilgan.

Jurnal 1 yilda 12 marta chiqadi.

Jurnal O‘zbekiston matbuot va axborot agentligi Buxoro viloyat matbuot va axborot boshqarmasi tomonidan 2016-yil 22-fevral № 05-072-sonli guvohnoma bilan ro‘yxatga olingan.

Muassis: Buxoro davlat universiteti

Tahririyat manzili: 200117, O‘zbekiston Respublikasi, Buxoro shahri Muhammad Iqbol ko‘chasi, 11-uy
Elektron manzil: nashriyot_buxdu@buxdu.uz

TAHRIR HAY‘ATI:

Bosh muharrir: Adizov Baxtiyor Rahmonovich – pedagogika fanlari doktori, professor

Mas’ul kotib: Sayfullayeva Nigora Zakiraliyevna – pedagogika fanlari bo‘yicha falsafa doktori (PhD)

Xamidov Obidjon Xafizovich, iqtisodiyot fanlari doktori, professor

Begimqulov Uzoqboy Shoyimqulovich, pedagogika fanlari doktori, professor

Navro‘z-zoda Baxtiyor Nigmatovich – iqtisodiyot fanlari doktori, professor

Mahmudov Mels Hasanovich, pedagogika fanlari doktori, professor

Ibragimov Xolboy Ibragimovich, pedagogika fanlari doktori, professor

Rasulov To‘lqin Husenovich, fizika-matematika fanlari doktori (DSc), professor

Yanakiyeva Yelka Kirilova, pedagogika fanlari doktori, professor (N. Rilski nomidagi Janubiy-G‘arbiy Universitet, Bolgariya)

Andriyenko Yelena Vasilyevna pedagogika fanlari doktori, professor (Novosibirsk davlat pedagogika universiteti Fizika, matematika, axborot va texnologiya ta‘limi instituti, Novosibirsk, Rossiya)

Romm Tatyana Aleksandrovna pedagogika fanlari doktori, professor (Novosibirsk davlat pedagogika universiteti Tarix, gumanitar va ijtimoiy ta‘lim instituti, Novosibirsk, Rossiya)

Chudakova Vera Petrovna, psixologiya fanlari nomzodi (Ukraina pedagogika fanlari milliy akademiyasi, Ukraina)

Hamroyev Alijon Ro‘ziqulovich – pedagogika fanlari doktori (DSc), dotsent

Qahhorov Siddiq Qahhorovich, pedagogika fanlari doktori, professor

Mahmudova Muyassar, pedagogika fanlari doktori, professor

Kozlov Vladimir Vasilyevich, psixologiya fanlari doktori, professor (Yaroslavl davlat universiteti, Rossiya)

Tadjixodjayev Zokirxo‘ja Abdusattorovich, texnika fanlari doktori, professor

Amonov Muxtor Raxmatovich, texnika fanlari doktori, professor

O‘rayeva Darmonoy Saidjonovna, filologiya fanlari doktori, professor

Durdiyev Durdimurod Qalandarovich, fizika-matematika fanlari doktori, professor

Mahmudov Nosir Mahmudovich, iqtisodiyot fanlari doktori, professor

Olimov Shirinboy Sharofovich, pedagogika fanlari doktori, professor

Chariyev Irgash To‘rayevich, pedagogika fanlari doktori, professor

Qiyamov Nishon Sodiqovich, pedagogika fanlari doktori (DSc), professor

Shomirzayev Maxmatmurod Xuramovich, pedagogika fanlari doktori, professor

Ro‘ziyeva Dilnoza Isomjonovna, pedagogika fanlari doktori, professor

Qurbonova Gulnoz Negmatovna, pedagogika fanlari doktori (DSc)

To‘xsanov Qahramon Rahimboyevich, filologiya fanlari doktori, dotsent

Nazarov Akmal Mardonovich, psixologiya fanlari bo‘yicha falsafa doktori (PhD), dotsent

Jumaev Rustam G‘aniyevich, siyosiy fanlari bo‘yicha falsafa doktori (PhD), dotsent

Nurulloyev Firuz No‘monjonovich, pedagogika fanlari bo‘yicha falsafa doktori (PhD)

Navruz-Zoda Layli Baxtiyorovna, iqtisodiyot fanlari bo‘yicha falsafa doktori (PhD)

MUNDARIJA

№	Familiya I.Sh.	Mavzu	Bet
FILOLOGIYA VA TILLARNI O‘QITISH			
1.	<i>NEMATOVA Zebo Tursunboyevna</i>	Boshlang‘ich sinflarda autentik materiallar orqali ingliz tili o‘qitish usullari	8
2.	<i>YADGAROVA Lola Jalalovna, ERGASHEVA Sarvinoz Bakhodirovna</i>	Effective use of innovative technologies in english lessons	14
3.	<i>КОМИЛОВА Дилдора Шавкатовна</i>	Нутқ одобида дўстлик концептининг берилиши	18
ANIQ VA TABIIY FANLARNI O‘QITISH			
4.	<i>НУРУЛЛОЕВ Феруз Нўмонжонович, ШАДМАНОВА Камола Умед қизи</i>	Методика обучения школьников современных программных средств	22
5.	<i>ABDULLAYEV Alibek Qodiraliyevich</i>	Pedagogika ta‘lim sohasi uchun mutaxassislar tayyorlashning yangi yo‘nalishlari	30
6.	<i>AVEZOV Alijon Xayrulloevich, TOSHPO‘LATOVA Shahzodabonu Voxid qizi</i>	Matematika fanini o‘qitishda noan‘anaviy ta‘lim yondashuvlari	34
7.	<i>BERDIYEVA Dinora Erkinovna</i>	Oliy ta‘lim muassasalarida “informatika va raqamli texnologiyalar” fanini o‘qitish muammolari	43
8.	<i>BOBOYEVA Muyassar Norboyevna, OCHILOVA Niginabonu Abduvoxid qizi</i>	Umumiy o‘rta ta‘lim maktablarida matnli masalalarni yechish usullari	48
9.	<i>ISROILOV Nurshohruh Sunnat o‘g‘li,</i>	Talabalarni virtual muhitda dasturlashga o‘rgatish usuli	54
10.	<i>KAMALOVA Nilufar Ilxomovna</i>	Semiotik yondashuv asosida python dasturlash tilini o‘qitish metodikasi	59
11.	<i>KHASANOVA Nilufar Khaqnazarovna, NIZAMOVA Saida Adilovna</i>	Finlandiya va O‘zbekistonda kimyo fanini o‘qitishning solishtirma tahlili	64
12.	<i>LUTFILLAEV Maxmud Xasanovich, MELIEVA Mohinur Baxromovna</i>	Kompyuter imitatsion modellar asosida o‘qitish texnologiyasi	69
13.	<i>MIXLIYEV Nurbek Avloyorovich</i>	Talabalarni kimyo fanidan amaliy tayyorgarliklarini biofalsafiy yondashuv asosida tashkil etishning pedagogik asoslari	74
14.	<i>NUROLLIYEV Novruz Shoymardon o‘g‘li</i>	Fizikadan talabalarni kasbga yo‘naltirib o‘qitishda ilmiy va o‘quv-bilish faolligini rivojlantirishning metodik modeli	78
15.	<i>NUROVA Oliya Salomovna</i>	Mashina detallari fanini o‘qitishda amaliy mashqulotlarini tashkil etish va o‘tkazish metodikasi	85
16.	<i>O‘KTAMOV Madadjon O‘ktam o‘g‘li</i>	Pedagogika oliy ta‘lim muassasalari talabalarining informatikadan axborot-texnologik kompetentligini rivojlantirish metodikasi	91
17.	<i>QODIROV Abbas</i>	Biologiya fanlarini masofaviy ta‘lim sharoitida o‘qitish	96

ANIQ VA TABIIY FANLARNI O‘QITISH

МЕТОДИКА ОБУЧЕНИЯ ШКОЛЬНИКОВ СОВРЕМЕННЫХ ПРОГРАММНЫХ СРЕДСТВ

*Нуруллоев Феруз Нўмонжонович,
Бухарский государственный университет кафедры
"Информационные системы и цифровые технологии" PhD, доцент
Шадманова Камола Умед қизи,
Учительница Каганской городской специализированной школы*

Практическое изучение современных программных средств, а также технологий, поддерживающих их разработку, имеет важное значение для развития технологий программирования. Однако по мере роста возможностей сред разработки растет и их сложность, что повышает их изучение и может привести к технологическому перекосу в обучении программированию.

Ключевые слова: информатика, технология, компьютер, программная средства.

MAKTAB O'QUVCHILARIGA ZAMONAVIY DASTURLARNI O'QITISH USULLARI

Dasturlash texnologiyalarini rivojlantirishda zamonaviy dasturiy vositalarni, shuningdek, ularning rivojlanishini ta'minlovchi texnologiyalarni amaliy o'rganish muhim ahamiyatga ega. Biroq, ishlab chiqish muhitlarining imkoniyatlari oshgani sayin, ularning murakkabligi ham oshadi, bu ularning o'rganish darajasini oshiradi va dasturlashni o'rgatishda texnologik tarafkashlikka olib kelishi mumkin.

Kalit so'zlar: informatika, texnologiya, kompyuter, dasturiy ta'minot.

METHODS OF TEACHING SCHOOLCHILDREN MODERN SOFTWARE

The practical study of modern software tools, as well as the technologies that support their development, is important for the development of programming technologies. However, as the capabilities of development environments increase, so does their complexity, which increases their learning and can lead to a technological bias in teaching programming.

Key words: computer science, technology, computer, software.

Десятки миллионов учеников по всему миру активно участвуют в различных мероприятиях с целью получения превосходных знаний, которые позволят им в конечном итоге занять хорошо оплачиваемую должность в IT-индустрии. Единственной целью их посещения кружков информатики, факультативов, других курсов, самостоятельного изучения и просмотра видеороликов на YouTube является расширение их знаний в области информационных технологий. Это стильно, утонченно, захватывающе, быстро развивается, и это позволит вам оставаться в курсе событий в мире высоких технологий будущего. учащиеся все равно учатся и входят в мир технологий, даже если они не намерены работать в IT-индустрии. Кроме того, программирование помогает учащимся улучшить свою память и навыки аналитического и критического мышления.

Практическое изучение современных программных средств и методов их разработки имеет важное значение для развития технологий программирования. К сожалению, сложность сред разработки возрастает вместе с их возможностями, а значит, и их изучение. Это может привести к технологическому перекосу в обучении программированию.

Область программирования пользуется большим спросом и хорошо оплачивается в современном информационном обществе. Однако, учитывая особенности этой области, саморазвитие имеет решающее значение, поскольку популярность программирования растет, а информационные технологии постоянно меняются - каждый день появляется что-то новое.

Одним из путей решения проблемы фундаментализации обучения программированию школьников является введение в курс информатики объектно ориентированного моделирования, с помощью специальных сред программирования с использованием готовых моделей. Вопросы обучения программирования с помощью специальных учебных сред посвящены работы многих

исследователей. В частности, в работах А. Г. Гейна, Н. В. Макаровой, А.П.Шестакова раскрываются методические аспекты обучения программированию с использованием визуальных моделей, в работах А. П. Одинцовой, Т. В. Софроновой, А. А. Тарасовой - геометро- графического моделирования; в работах А. В. Бобровской, Л. В. Кавурко, И. Г. Обойщиковой - математического моделирования в учебных средах программирования; в работах Д. Д. Бычковой, П. В. Кийко, И. В. Паболков, И. А. Теплицкого, Л.Н.Шенгелия показана интегративная роль учебного компьютерного моделирования.

Иностранцы специалисты приводят множество аргументов, почему детей необходимо обучать программированию. Важным аспектом является использование математического аппарата, то есть визуализация математических навыков, или, например, через выполнение задач программирования приложений Scratch, выработка навыков работы в команде [1].

С конца двадцатого века большинство ученых сошлось на том, что в обучении программированию школьников среднего и старшего звена следует придерживаться принципов конструктивизма и его разновидности – конструкционизма, что заметно повлияло на педагогический дизайн и воплотилось в следующих педагогических средах:

- Лого и его производные NetLogo и StarLogo;
- Squeak и производный от него Scratch;
- Alice и др.

Сделаем краткий обзор программных систем для изучения программирования.

Название Лого (LOGO) происходит от греческого «логос», что означает «слово», «смысл», «идея». В литературе термин «Лого» используется в двух значениях:

1) как язык программирования, настолько проста, что ее могут освоить дети, но настолько мощная и выразительная, что и опытный программист найдет в ней много интересного;

2) как философия обучения, система взглядов на процесс обучения, призванная, как считает С. Пейперт, а не частично улучшить, а коренным образом революционизировать традиционную организацию обучения. Язык Лого занимает в этой системе центральное место. Философия обучения Лого предполагает преобразование традиционной системы обучения и основывается на идее «использование компьютера как модели, которая может повлиять на наш образ мышления о самих себе» [2].

Объекты среды Лого — это компьютер, «математически говорит за существо» [3] и Черепаха, «кибернетическая животное», управляемая с помощью компьютера. Обучение в Лого происходит в процессе «бесед» ученика с Черепахой и компьютером в отличие от традиционной организации обучения, в Лого не компьютер управляет процессом обучения, а ученик «учит» компьютер, «сказал ему» на языке Лого.

Отправляя команды-сообщения объекту «Черепаха», ученики естественным образом усваивают принципы объектно-ориентированного программирования. «Работа в микромире Черепахи — это модель изучения идей в тот самый способ, которым мы познаем другого человека. Ученики, которые работают в этой среде, безусловно, открывают в нем интересные факты, приходят к обобщениям, усваивают навыки» [2].

Исследовательские возможности StarLogo получили дальнейшее развитие в NetLogo, в котором в последние годы были построены различные исследовательские модели, которые использовались в научных статьях и обсуждались в книгах по мультиагентного моделирования и социологии [4].

Scratch - еще одна среда программирования, созданная под руководством М. Резника [5]. Scratch позволяет детям создавать собственные анимированные и интерактивные истории, игры и другие творения. Основная задача проекта - стать частью образовательной программы для детей и подростков, развить в них творческие способности, логическое мышление и свободу в использовании информационных технологий. Все это предлагается развить путем привлечения учащихся к процессу конструирования интерактивных презентаций, мультимедиа, игр. Дети могут составлять свои программы из блоков команд («кирпичиков») так же, как они строили домики и машинки из деталей «Лего».

Squeak — язык программирования, диалект языка Smalltalk. Имеет кроссплатформенную реализацию (Windows, Linux, Macintosh). Первоначально Squeak был разработан группой программистов Apple Computer, в которую входили некоторые разработчики Smalltalk-80. Разработка была продолжена той же группой уже в Walt Disney Imagineering. На данный момент Squeak доступна абсолютно бесплатно для любого использования. Кроме того, Squeak полностью доступен в исходных кодах (в том числе и виртуальная машина). В Squeak реализовано несколько графических подсистем (в том числе MVC, унаследованная от оригинального Smalltalk-80, в текущей версии не

поддерживается, работает в версиях младше 3.8). Однако основной является собственная графическая подсистема Morphic (портированная из Self).

Поэтому при «исчерпании» возможностей Scratch по мере развития навыков программирования можно перейти к родительской мультимедийной среде объектно-ориентированного моделирования Squeak, используя мощные средства ООП языка Smalltalk. Так же, как и в NetLogo, вокруг Scratch существует сообщество. Scratch приучает собирать проект из кирпичиков и делиться результатами своих действий с другими людьми. Эти навыки важны не только внутри специальных сред программирования, но и в современных сетевых сообществах. Единство процессов создания, поиска и хранения информационных кирпичиков все чаще можно наблюдать на страницах современных сайтов, использующих концепцию Web 2.0. Метафора строительных блоков, из которых дети и взрослые могут собрать простые и очень сложные конструкции, присутствует не только в учебных проектах, но и в большинстве современных сетевых сервисов форматов Web 2.0, предназначенных для поддержки организаций и сетевых сообществ обмена знаниями:

1) обучающиеся представляют и представляют, что именно они хотят сделать и получить в результате;

2) обучающиеся создают проект, основанный на своих представлениях;

3) обучающиеся играют с результатами своей деятельности;

4) обучающиеся делятся результатами своей деятельности с другими людьми;

5) обучающиеся обдумывают и обсуждают свои результаты;

6) обсуждения и обдумывания приведет к новым представлениям и новым проектам.

Среда разработки Alice, так же, как и Scratch - относительно новый проект, разрабатываемый в университете Карнеги-Меллона. В отличие от своих предшественников, Alice - полностью трехмерная среда моделирования. Alice 2.2 позиционируется разработчиками как средство обучения объектноориентированного программирования, а Alice 3 - как средство объектноориентированного моделирования [6]. Alice является средой, в которой можно манипулировать 3D объектами (двигать, вращать, менять цвет и т.д.) и создавать программы, которые генерируют анимацию в виртуальных мирах.

Среда программирования Alice выпускается для Windows, Mac OS и Linux в двух редакциях: основной (для вузов) и упрощенной (для школ). В среде имеется большая библиотека объемных объектов из реального мира (природа, животные, инструменты быта и т.д.). Их можно двигать, вращать, менять цвет и размер при помощи мыши, а на основе полученного виртуального мира программно описывать анимацию и создавать игровые модели. Окна Alice очень похожи на те, что встречаются в профессиональных современных интегрированных средах разработки визуальных приложений типа Visual Studio.

Созданный проект запускается на исполнение в отдельном окне при помощи кнопки Play. В среде Alice используется собственный встроенный язык программирования, приближенный к синтаксису языков современных объектноориентированных языков программирования таких, как Java, C++ или Visual Basic. Поскольку программное обеспечение Alice позволяет создавать только синтаксически правильные команды, то программирование в данной среде сводится лишь к разработке и реализации соответствующих алгоритмов. [7]

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

Программирование в Alice отчасти напоминает работу в среде Scratch. Процесс написания игры напоминает игру «Пазлы», когда необходимо состыковать между собой программные элементы, из которых собирается вся программа. От учащегося лишь иногда требуется ввести с клавиатуры константы или название собственных идентификаторов. Для наглядности представленного кода блоки программных элементов выделяются определенным цветом в зависимости от их типа. Эти блоки можно легко перетаскивать, меняя порядок и вложенность [8].

Таким образом, Alice – игровая среда программирования, создающая игры и анимационные фильмы, обучающая основам объектно-ориентированного программирования, позволяет передать поведение реального мира, описывая взаимодействия объектов, скрывая детали реализации, разрабатывать программное обеспечение повышенной сложности за счет улучшения его технологичности (лучших механизмов разделения данных, увеличения повторяемости кодов, использования стандартизованных интерфейсов пользователя и т.д.).

Одной из учебных сред, используемых на уроках информатики, является «Scratch» - объектно-ориентированная визуальная среда программирования.

Можно начинать пользоваться языком с нуля, не обладая никакими предварительными знаниями в программировании. Как уже писалось ранее, в среде Scratch используется метафора кирпичиков Лего, из которых даже молодые дети могут собрать простые конструкции. Но, начав с малого, можно дальше развивать и расширять свое умение строить алгоритмы и программировать [9]. Scratch создавался специально для того, чтобы подростки 10 - 16 лет использовали его самостоятельно в сети внешкольного обучения. Важно отметить, что Scratch приходит в современный мир вместе с другими важными педагогическими новациями. Согласно идеологии этого движения, ребенок должен осваивать не программы, а различные способы деятельности: создавать свои собственные истории, придумывать игры, разрабатывать компьютерные и алгоритмические модели. Данная система проста для восприятия даже детям в младшей школе, ведь все операторы языка и другие его элементы представлены блоками, которые могут соединяться друг с другом, образуя скрипт (фрагмент кода) [7]. Программируя на Scratch ученики получают понятия алгоритмизации и программирования, создают игры, анимации или музыку.

Определим требования к применению и созданию анимации в программе Scratch:

- доступность, простота применения;
- соблюдение методических правил по количеству анимационных изображений, времени, выделенного на их применение;
- соответствие подобранных анимационных изображений теме и дидактическим целям, и задачам занятия;
- соответствие анимационных изображений возрастным особенностям детей;
- качество анимационных изображений, обоснованность и рациональность применения [10].

Анализ публикаций в иностранных изданиях свидетельствует о накоплении значительного опыта внедрения среды программирования Scratch в учебный процесс, а именно по этой тематике разработаны курсы в Гарвардском университете, Калифорнийском университете в Беркли, колледже Нью-Джерси других [1, 9].

Использование среды Scratch в начальных и средних классах, а также изучение элементов программирования со старшими школьниками на факультативных занятиях вызывает необходимость показать возможности работы в среде программирования, сочетая изучение алгоритмизации (алгоритмических структур: ветвления, циклы другие) и элементов графического интерфейса.

Проведен апробация с целью определения того, что программа курса в Scratch, будет являться хорошим введением в тему «Начало программирования».

Идея апробации заключается в том, чтобы показать позитивное влияние практических занятий в Scratch на развитие алгоритмических умений у обучающихся. Апробация направлена и на решение педагогических задач, в частности, на развитие алгоритмического мышления у обучающихся посредством практических работ. В связи с этим цель планируемой пробации: доказать, что программа курса является хорошим введением в тему «Начало программирования» и способствует развитию алгоритмических умений у обучающихся.

Для реализации цели апробации выдвинуты следующие задачи:

1. Продиагностировать у обучающихся уровень развития алгоритмических умений.
2. Организовать занятия с использованием программы курса по изучению Scratch.
3. Проанализировать результаты апробации.

Апробация планируется в три этапа:

- I. Подготовительный
- II. Рабочий
- III. Аналитический

В процессе изучения главы «Основы алгоритмизации» обучающиеся должны освоить следующие умения: разрабатывать план решения задачи, выдвигать и доказывать гипотезы, прогнозировать результаты решения, анализировать и находить рациональные способы. Именно эти мыслительные умения характеризуют уровень развития алгоритмического мышления.

Алгоритмическое мышление – совокупность мыслительных действий и приемов, нацеленных на решение задач, в результате которых создается алгоритм, являющийся специфическим продуктом человеческой деятельности.

Под способностью алгоритмически мыслить понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата.

Поэтому на подготовительном этапе деятельность была направлена не только на разработку программы курса, но и подборку диагностического инструментария для осуществления оценки сформированности алгоритмических умений.

Была разработана диагностическая работа, целью которой было определение первоначального уровня развития алгоритмических умений у обучающихся. Работа состояла из 5 заданий.

В качестве критериев для оценки выполнения заданий были приняты следующие умения, характеризующие уровень сформированности алгоритмического мышления: - умение упорядочивать действия в алгоритме; - умение выполнять алгоритм: линейный, разветвленный; - умение составлять план действий; - умение сопоставить задачу и готовый алгоритм – модель ее решения. Правильно выполненное задание оценивалось 1 баллом. Максимальное количество баллов – 5.

Классный коллектив стабильный, самостоятельный, в процессе наблюдения сплоченность коллектива проявлялась на низком уровне. Общий уровень дисциплины в классе - удовлетворительный.

Для успешной организации апробации необходимо соблюдение социально – педагогических и психолого-дидактических условий.

1) необходимые средства обучения – компьютер, проектор;

2) доверительность отношений между учащимися и учителем, основанных на взаимном уважении и интерактивности, что обеспечит благоприятную обстановку в «учебной» аудитории, способствующую успешному овладению учебным содержанием;

3) использование разнообразных методов обучения:

- словесные методы обучения (рассказ, объяснение, лекция, беседа, работа с учебником и книгой);

- наглядные методы (наблюдение, иллюстрация, демонстрация презентаций);

- практические методы (устные и письменные упражнения, практические компьютерные работы).

Социально – дидактические:

1) статус школы (зависит от того, какое оборудование есть в информационных классах, по какой программе работают учащиеся, компетенции учителя);

2) уровень обученности обучающихся, соответствующий программным требованиям, основанный на логичности, преемственности и последовательности содержания;

3) соблюдение дидактических принципов и правил организации учебного процесса, так как в условиях гуманизации образования, развития новых экономических механизмов вся совокупность требований к учебному процессу сводится, прежде всего, к соблюдению дидактических принципов обучения;

4) наличие у обучающихся возможности использовать компьютер в домашних условиях. На первом уроке была проведена диагностическая работа.

На основании этих результатов был проведен анализ выполнения заданий и получены следующие выводы.

С заданием № 1, в котором требовалось упорядочить действия в алгоритме, успешно справились 86% обучающихся – все действия поставлены в правильном порядке.

С заданием № 2, на выполнение линейного алгоритма, справились 90% обучающихся – они точно выполнили все предписания алгоритма, 3 обучающихся не справились, показав непонимание требуемых предписаний алгоритма.

С заданием № 3, на выполнение разветвленного алгоритма, справились 57 % обучающихся, у остальных возникли трудности при вычислении значения переменной x .

С заданием № 4, в котором проверялось умение составлять план действий, успешно справились 67 % обучающихся – все действия поставлены в правильном порядке.

Задание № 5, на сопоставление задачи и готового алгоритма, успешно справились 95% обучающихся. Не смог выполнить задание всего 1 обучающийся.

Результаты выполнения отдельных заданий представлены на рисунке 1.

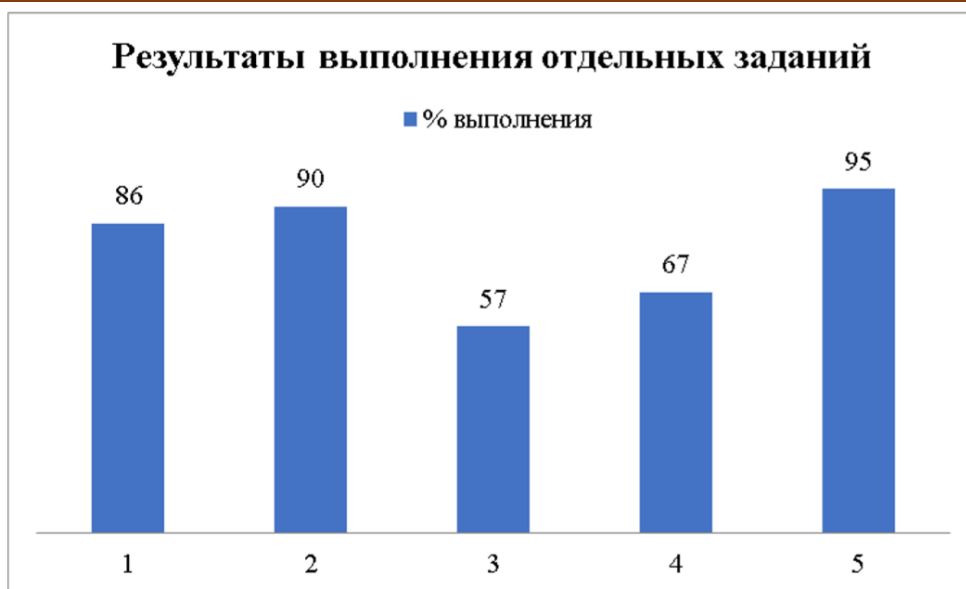


Рисунок 1. Результаты выполнения отдельных заданий

По результатам выполнения заданий были выделены следующие уровни: - 90-100% – высокий уровень; - 65-90 – средний уровень; - менее 65 – низкий уровень.

Распределение обучающихся по уровням сформированности алгоритмических умений (рисунок 2).



Рисунок 2. Распределение обучающихся по уровням сформированности алгоритмических умений

В ходе первичного замера было установлено, что в классе, в целом, наиболее сформированы такие показатели, как: упорядочивать действия в алгоритме; выполнять линейный алгоритм; сопоставить задачу и готовый алгоритм – модель ее решения.

Это обусловлено тем, что обучающиеся уже имели опыт выполнения подобных заданий. Таким образом, на рабочем этапе был выявлен уровень сформированности алгоритмических умений у обучающихся 6б класса Каганской городской специализированной школы. После оценки сформированности алгоритмических умений началась апробация курса. Было отведено 4 урока в рамках учебного процесса.

Были проведены следующие уроки: 1. Урок лекция. Интерфейс Scratch. 2. Практическое занятие: Путешествие в космосе. 3. Практическое занятие: Рисование собственного спрайта. 4. Практическое занятие: Проект «Часы».

На этапе знакомства с программой Scratch, работа была организована во фронтальном режиме (учитель показывал, знакомил с интерфейсом и основными инструментами). Уроки прошли успешно, обучающимся нравилось заниматься в среде программирования Scratch, материал был освоен хорошо. Дети с интересом выполняли практические работы. На первых уроках ученики не могли запомнить значения определенных блоков, терялись в интерфейсе. Но спустя 1 занятие, школьники освоили интерфейс программы. Так же, ученики начали придумывать свои собственные истории и воплощать их на экранах компьютера с помощью Scratch дома.

Целью аналитического этапа выступает анализ проделанной работы на рабочем этапе. В ходе первичного замера было установлено, что умения 3 и 4 находятся на низком уровне. Поэтому вторичный замер был ориентирован на данные умения.

В ходе вторичного замера мы получили следующие результаты. (Рисунок 3).



Рисунок 3. Результаты выполнения отдельных заданий

С умением № 3, на выполнение разветвленного алгоритма, справились 81 % обучающихся. С умением № 4, в котором проверялось умение составлять план действий, успешно справились 83 % обучающихся.

Анализируя результаты первичного и вторичного замеров, мы делаем вывод, что по данным показателям произошли существенные изменения.

Распределение обучающихся по уровням сформированности алгоритмических умений при использовании методики обучения школьников программированию в среде Scratch (Рисунок 4).

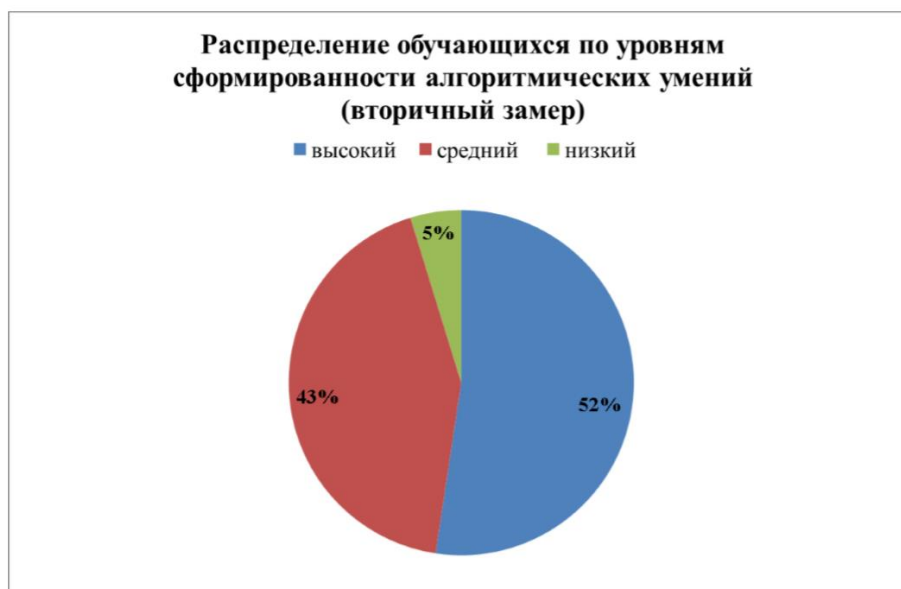


Рисунок 4. Распределение обучающихся по уровням сформированности алгоритмических умений

Положительный результат апробации обусловлен четко составленными практическими занятиями, понятными для данной возрастной группы, а также красочным и понятным интерфейсом программы Scratch, вызывающим интерес у школьников. Что указывает на достаточный уровень разработанности программы курса по изучению Scratch в 6 классе.

По результатам пробации можно определить перспективные направления деятельности в части разрабатываемой проблемы:

1. Совершенствование методических рекомендаций по организации курса, дополнения примерами работ обучающихся и алгоритмами выполнения некоторых типовых заданий.
2. Совершенствование содержания конспектов занятий за счет расширения спектра сюжетов занятий.

Таким образом, проведенная апробация позволила организовать занятия в 6 классе на основе разработанной методики и установить ее эффективность в части развития у обучающихся алгоритмических умений. Все это позволяет сделать вывод о том, что поставленные задачи и цель исследования достигнуты в полном объеме.

Выводы

В статье оопределенно основные достоинства и преимущества использования Scratch в процессе изучения основ и принципов программирования обучающимися общеобразовательной школы на уроках информатики. Проанализированы существующие программы, моделирующие процесс программирования в наглядно-графической форме. Разработано и обосновано систему занятий для обучения программированию школьников с использованием визуальной среды разработки Scratch. Проведена апробация разработанных материалов, которая показала заинтересованность в изучении программирования в среде Scratch.

Приобретенные школьниками приемы работы и умение использовать алгоритмические конструкции дают возможность более основательно и самостоятельно работать с программой. Использование среды Scratch в учебной деятельности позволяет: активизировать процесс их подготовки к учебным занятиям, созданию интерактивных анимированных материалов, коллективно работать над проектами и обмениваться результатами через Scratch сообщество, побуждать к самостоятельной деятельности.

ЛИТЕРАТУРЫ:

1. Денисова Л.В. Среда Scratch в практике учителя начальной школы / Л.В. Денисова, В.О.Дженжер // Начальная школа. 2012. - № 5. С. 31-35.
2. Зайцев В.В. Современные педагогические технологии: учебное пособие/ В.В. Зайцев//Челябинск, ЧГПУ, 2012-411с.
3. Сорокина Т.Е. Визуальная среда Scratch как средство мотивации учащихся основной школы к изучению программирования / Т.Е. Сорокина // Информатика и образование. 2015 - №5 (264). - С. 30 - 34.
4. Ломаковская Г. В. Ступеньки к информатике: учеб. для 2 кл. общеобразоват. учеб. заведений / Г. В. Ломаковская, Г. А. Проценко, И. Я. Ривкинд, Ф. М. Ривкинд. - М.: Издательский дом «Образование», 2012. - 160 с.
5. Ломаковская Г. В. Ступеньки к информатике: учеб. для 3 кл. общеобразоват. учеб. заведений / Г. В. Ломаковская, Г. А. Проценко, И. Я. Ривкинд, Ф. М. Ривкинд. - М.: Издательский дом «Образование», 2013. - 160 с.
6. Патаракин Е.Д. Учимся готовить в среде Скретч / Е.Д. Патарикин. - М.: 2007.
7. Лесневский А. С. Объектно-ориентированное программирование для начинающих / А. С. Лесневский. – М. : БИНОМ. Лаборатория базовых знаний, 2005. – 232 с.
8. Теплицкий А. И. Средства обучения объектно-ориентированного моделирования студентов естественных специальностей педагогических университетов / А. И. Теплицкий // Сб. наук. пр. Кам.-Подол. нац. ун-та. Серия педагогическая. - Каменец-Подольский: Кам.-Подол. нац. ун-т им.И.ОГИЕНКА, 2011. - Вып. 17. - С. 246-248.
9. Кончакова Р.Б. Анимационные программы в курсе информатики для студентов гуманитарных специальностей / Р.Б. Кончакова, М.Ю. Сидляр // Психологопедагогический журнал Гаудеамус. 2014 - № 2 (24). - С. 137-143.
10. Пейперт С. Переворот в сознании: дети, компьютеры и плодотворные идеи / Сеймур Пейперт. – М. : Педагогика, 1989. – 224 с. 10.