

Главное управление образования Гродненского облисполкома
Управление образования Лидского райисполкома
Государственное учреждение образования
«Лидский районный центр технического творчества»

А.В. Янковская

**Проблемы изучения нескольких языков
программирования в одном
объединении по интересам в УДОДиМ:
пандемия, дистанционное образование,
опыт работы и вектора развития**

Лида

2021

3

ББК 74.200.58

Я 62

Рекомендовано педагогическим советом государственного учреждения образования «Лидский районный центр технического творчества»

Автор: А.В. Янковская, методист государственного учреждения образования «Лидский районный центр технического творчества»

Рецензент: В.А. Грабовский, директор ГУО «Лидский районный центр технического творчества»

Я 62 Янковская, А.В. Проблемы изучения нескольких языков программирования в одном объединении по интересам в УДОДиМ: пандемия, дистанционное образование, опыт работы и вектора развития / А.В. Янковская – г. Лида: ГУО «Лидский районный центр технического творчества», 2021. - 59 с.

В статье рассматриваются проблемы изучения нескольких языков программирования в одном объединении по интересам в УДОДИМ, определяется место видеоконференцсвязи Zoom, Google Meet, Microsoft Teams в среде Интернет-обучения программированию, рассматриваются средства поддержки интерактивности процесса обучения, специфические для педагога УДОДиМ как тьютора дистанционного обучения компетенции. Обозначены проблемные области, возникающие в современном УДОДИМ при применении инновационных технологий, предложены этапы подготовки

Адресуется педагогам дополнительного образования

ББК 74.200.58

© А.В. Янковская, 2021

© Государственное учреждение образования «Лидский районный центр технического творчества», 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ПРОБЛЕМЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ.....	6
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15
Приложение 1.....	20
Приложение 2.....	21
Приложение 3.....	21
Приложение 4.....	38

ВВЕДЕНИЕ

Государственное учреждение образования «Лидский районный центр технического творчества» работает в области обучения детей и молодежи робототехнике и программированию с 2015 года, сравнительно недавно. Несмотря на небольшой опыт работы в данных направлениях, за истекший период было апробировано немало образовательных программ и накоплен достаточный опыт работы, чтобы конкретизировать возникающие проблемы и выделить важные векторы дальнейшей работы.

Современное дополнительное образование детей и молодежи достаточно популярная ветвь образовательного процесса. Общественная потребность, актуальная для общества в целом диктует дополнительному образованию новые современные требования в образовательных областях «Робототехника», «Информатика». С учётом идеологической направленности официальной политики белорусского государства, отраженной в Концепции информатизации системы образования Республики Беларусь на период до 2020 года, утвержденной министром образования Республики Беларусь, для дополнительного образования детей и молодежи задаются основные цели, задачи, направления информатизации, а также определяются базовые принципы, подходы и условия для успешной реализации процесса информатизации. [1]

Социальный заказ общества на выпускника современного учебного заведения проявляется в требованиях к его образованности. Поэтому вопрос повышения качества образования является определяющим в деятельности педагогического коллектива любого образовательного учреждения. Вместе с тем в условиях реализации

лично ориентированной парадигмы образования речь должна идти о развитии личности, учете ее интересов, потребностей и возможностей. Следовательно, качество образования имеет социально-личностное значение.

Категория «качество» в социально-личностном значении, ориентирует работу учреждения дополнительного образования на обучение робототехнике и программированию. В учреждении дополнительного образования детей и молодежи данный процесс в образовательных областях «Информатика», «Робототехника», соответствует стратегии «Наука и технологии: 2018-2040» [2], подготовленной во исполнение поручения президента Республики Беларусь А.Г.Лукашенко от 07.04.2017, соотносится с ключевыми компонентами модели «Беларусь интеллектуальная»: сквозная цифровизация экономики и создание IT-страны, развитый неиндустриальный комплекс, высокоинтеллектуальное общество и базируется на принципах Государственной программы инновационного развития Республики Беларусь на 2016-2020 г.г. [3]

ПРОБЛЕМЫ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Самый главный проблемный вопрос в обучении программированию - преемственность.

При обучении детей младшего школьного возраста программированию была выявлена проблема при обучении Scratch с 1 класса. Обучение велось по программам, предоставленным в рамках образовательного проекта Парка высоких технологий и Министерства образования Республики Беларусь [5]

Практика работы с детьми данного возраста была успешной и конструктивной только с теми учащимися первого класса, которые владели навыками чтения. С остальными учащимися работать было крайне сложно, снижалась скорость выполнения плана занятия, дети, умеющие читать и быстрее выполняющие работу, оставались без внимания. Кроме того, существует проблема резерва часов в учреждении, что так же накладывает существенные ограничения при построении образовательной траектории. [4]

Следующая проблема связана с обучением программированию старшей группы. При обучении детей начального уровня, используя программы Scratch, важно понимать, в каком направлении и на каком языке будет писать программы учащийся далее. Здесь есть несколько вариантов движения от Scratch: - к Python или Java, либо C++. Большинство детей после 2 и более лет (в основном 2-3 года) обучения Scratch имеют достаточно высокий уровень компетенций по программированию, позволяющий работать в любом направлении- на языке Python или Java, либо C++. Но, при переходе в периоде 7 класса и старше возникает конфликт обучения со школьной программой

программирования на языке Pascal. Следует отметить, что дети путаются в синтаксисе. Но проблема даже не в этом, а в том, что данный язык не находит применения в будущем при дальнейшем обучении, особенно в техническом ВУЗе. Так же существует проблема низкой осведомленности родителей. Некоторые стремятся сразу с начальной школы к высокому, практически вузовскому уровню программирования, что снижает мотивацию из-за сложностей проектов.

В государственном учреждении образования «Лидский районный центр технического творчества» мы работаем по нескольким схемам в обучении программированию, которые определяют основные векторы нашей работы.

2-4 класс (7-10 лет) — 1 год обучения программированию Scratch. Данный период является входным в компетенции программирования;

с 5 класса и старше (с 11 лет) — учащимся предлагаем переход к Python или Java, либо C++ при условии достаточного уровня владением компетенциями программирования на Scratch. Здесь следует отметить, что и в 7 классе программирование Scratch в качестве проектной деятельности даже на первом году обучения вызывает неподдельный интерес у учащихся, и в 8 классе вход в Lego Mindstorms EV3 также будет результативен и интересен.

Каждая программа со старта, с первого года обучения, предполагает работу до 3х лет. В своем большинстве учащиеся все же занимаются два года. Не составляет труда просчитать, что к 8-9 классу (с 14 лет) некоторые дети теряют интерес и мотивацию к робототехнике и программированию. Здесь мы предлагаем детям объединения по интересам САПР, 3D МАХ, Прототипирование. На этом этапе важно не упустить таких детей, сформировать банк их достижений, отобрать

одаренных высокомотивированных учащихся, построить для них индивидуальную образовательную траекторию, практикоориентированную, интересную, творческую и важную для самого учащегося. Здесь педагогическую проблему составляет построение для учреждения образования программы работы с одаренными учащимися.

Апробированные тематические планы программ объединений по интересам можно рассмотреть в Приложении 1 и 2.

Проанализируем представленный тематический план в Приложении 2. Большинство программистов знают больше одного языка, а наиболее успешные практикуют разработку на нескольких одновременно, даже если специализируются только на одном. Языки программирования мало чем отличаются от обычных разговорных языков: тот же набор ключевых слов, символов, а также правила порядка их употребления. Основная работа программиста — не написание кода, а разработка алгоритмов. Язык лишь средство выражения алгоритма в понятной для машины форме. Знание нескольких языков расширяет угол зрения для оценки задач.

Каждый язык программирования по-своему уникален, это позволяет со свойственной ему уникальностью решить задачу. Язык из сферы базового образования C++ можно назвать антагонистами. Их сравнение кардинально меняет восприятие разработки. Язык определяет предел способностей к оценке окружающего мира и степени влияния на него. Таким образом сложность задач, которые мы способны решать, зависит от того языка, которым мы думаем.

Когда программист решает задачу, то он мыслит инструментами, которые этот язык программирования способен дать. Американский

предприниматель Пол Грэм отметил, что специалист с низкой квалификацией не в состоянии осознать этого из-за своей низкой квалификации. Уже в отношении разработки он наблюдает зависимость программиста от языка, которым тот пользуется. Разработчики любят «свои» языки поскольку привыкли думать на них, а языки, в свою очередь, лексикой диктуют программистам, как они должны думать о программах. Это образует парадокс, который невозможно разорвать без изучения нового языка. [6]

Зачем нужно так много разных языков программирования&

Практически все языки программирования появились для решения специфических задач или в ответ на невозможность решить их существующими инструментами.

Python был создан на досуге — нужен был расширяемый скриптовый язык. Сегодня он стабильно возглавляет чарты популярности и является универсальным языком: от разработки обычных сайтов до ML.

C++ -опытный разработчик осмысленно выбирает каждый следующий язык программирования для изучения под конкретные задачи. Не стоит в качестве первого языка выбирать какой-либо из малораспространённых языков. Выбирайте наиболее популярные классические языки, например, C++, C#, Java, JavaScript, Python, которые научат использовать правильный стиль программирования и мыслить, как программист. Узкоспециализированные языки можно выучить позже, когда уже заложена основа. Это расширит ваш кругозор и дополнит сложившуюся картину. А там, кто знает, может быть, однажды вы сами создадите востребованный язык программирования. Естественно, не стоит тратить время на изучение языков, не

предназначенных под ваши задачи. Например, нет необходимости учить С для реализации веб-интерфейса.

JavaScript разработали для дизайнеров и программистов с низкой квалификацией, что «оживить» веб-сайты, сделав их динамическими. Сейчас это один из самых востребованных языков, без него невозможно представить современный интернет.

Кроме того, важно отметить роль профориентационной направленности при обучении программированию с использованием различных языков программирования. Не секрет, что лучшая мотивация для ребенка- это уметь делать так, как делает взрослый. В данном случае, владеть компетенциями программирования как входными для начинающего профессионала, причем владея основами нескольких языков. Проектно-ориентированное обучение программированию способствует важности профориентирования в жизни обучающегося.

При работе в современном режиме в условиях пандемии обучение программированию как никогда не утратило своей актуальности. Учащиеся при организации дистанционных занятий освоили платформы Zoom, Google Meet, Microsoft Teams

Определим место платформ видеоконференцсвязи в учебном Интернет-процессе. Формы Интернет-обучения и средства поддержки интерактивности в зависимости от режима общения и типа взаимодействия приведены в табл.1.

Место видеоконференцсвязи в учебном Интернет-процессе

Форма	Синхронная (on-line)	Асинхронная (off-line)
Лекция	Видеоконференцсвязь	Гипертекст

		Видеозапись
Семинар	Видеоконференцсвязь	Форум
Индивидуальная работа		Электронная почта Блог
Контрольная работа	Онлайн тестирование	Электронная почта
Групповой проект	Вики	
Обсуждение в виртуальной группе	Чат	Электронная рассылка
Консультация	Мессенджер Видеоконференцсвязь	Электронная почта
Контроль знаний	Видеоконференцсвязь Онлайн тестирование	Офлайн тестирование

Синхронные и асинхронные средства возможно комбинировать при реализации различных форм дистанционного обучения.

Определим несколько проблемных областей в ходе анализа ситуации, складывающейся в современном УДОДиМ при применении инновационных технологий в обучении программированию.

Во-первых, повышенное внимание к использованию инновационных технологий в образовательном процессе ведет к утрате смысла, важности самой цели обучения программированию. Например, применение тестовых методик приводит к завышению значения результата тестирования, превращая прохождение теста в самоцель.

Во-вторых, предъявляются повышенные требования к преподаванию в системе дистанционного обучения. Если преподаватель не владеет информационными средствами, он не может пользоваться ресурсами. Дистанционное обучение должно быть четко отлаженным,

максимально сбалансированным в отношениях «образовательная среда - личность - информационные технологии» механизмом по реализации целей обучения, направленным на раскрытие потенциала личности.

В-третьих, деятельность в условиях насыщенной информационно-коммуникационной образовательной среды приводит к значительной информационной перегрузке, вызванной обилием источников информации и невозможностью охватить даже сколь-нибудь значительную ее часть. В данном аспекте актуальными становятся проблемы формирования информационной здоровьесберегающей культуры и создание здоровьесберегающей педагогики, основанной на здоровьесберегающих технологиях [7].

В-четвертых, применение только асинхронного дистанционного обучения программированию может привести к психологическому дискомфорту. Психологический диссонанс возникает в том случае, если наряду с виртуальным общением, учащийся не общается с педагогом напрямую, очно. Также существует опасность утери креативности и сотрудничества, реализуемых непосредственно в совместной деятельности, некоего инсайта в работе.

Обращение к технологиям платформ Zoom, Google Meet, Microsoft Teams способствует преодолению вышеназванных проблем. Это тот путь, когда исключается ситуация «один на один, учащийся и компьютер». Это «живой язык общения», прочтение учебной аудитории. Zoom, Google Meet, Microsoft Teams должны быть грамотно построенным механизмом взаимодействия личностного потенциала и информационного ресурса.

Zoom, Google Meet, Microsoft Teams в настоящее время становятся основой для создания единого сообщества образовательных

учреждений, позволяющей обмениваться опытом, проводить мастер-классы и конференции независимо от территориального нахождения участников образовательного процесса.

Руководствуясь идеей, что цели обучения являются определяющими, можно установить следующую этапность подготовки и проведения занятия программированием в дистанционной форме.

1. Этап целеполагания. Необходимо ответить на вопросы, а что является целью образовательного проекта?

Какие знания и навыки приобретет учащийся? Какое место займет дистант в учебном процессе?

2. Подготовительный этап. На этом этапе необходимо определить уровень подготовки учащихся к обучению в дистанционном режиме, качество материально-технического обеспечения интерактивной связи, наличие телекоммуникационных сетей, соответствие выбранной технологии аудитории.

3. Этап технологического оснащения (проектирования). На данном этапе определяется техническое оснащение, технологические и методолого-дидактические умения преподавателя, выбирается куратор проекта, временные рамки, расписание контактов, проектирование инструментов синхронного и асинхронного взаимодействия.

4. Проектирование дидактического маршрута. Определение дидактической среды обучения онлайн, определение дидактического формата. На данном этапе проходит регистрация участников видеоконференции. Осуществляется загрузка дидактических материалов, конкретизируются вопросы для размышления, определяются предметные и дидактико-методические темы и вопросы,

проводятся предварительные консультации, форумы, где обсуждаются актуальные вопросы и затруднения.

5. Этап проведения и реализации целей проекта. Главным принципом на этом этапе является согласованность действий всех участников процесса. Zoom, Google Meet, Microsoft Teams как технологии включает не только традиционные для всех форм дистанционного обучения принципы. Мы можем выделить такие специфические, как общение «здесь и сейчас», персонификация высказываний, максимальное число деловых контактов и общения с различными людьми, активное участие в происходящем, уважение и доброжелательное отношение ко всем участникам конференции, постоянная обратная связь, диалогизация взаимодействия, то есть равноправное полноценное межличностное общение.

6. Этап оценки. На этом этапе нужно определить критерии для оценивания эффективности проведенной работы обучению программирования в дистанционной форме. Определение проблемных областей, значимых событий, достигнутых целей будет способствовать дальнейшему совершенствованию технологии видеоконференции.

При проведении Zoom, Google Meet, Microsoft Teams для распределенной аудитории необходимо учитывать ряд требований, основанных на технических, методических и психолого-педагогических особенностях данного вида коммуникации [8]. Таких, как требования к слайдам и печатным иллюстрациям (контрастность, компоновка и т.п.), к искусству общения, к навыкам демонстрации и передачи иллюстрационного материала, к временной проработке сценария занятия, к психологической подготовке педагога дополнительного образования. Есть общие «законы жанра», касающиеся особого

характера восприятия учебной информации с экрана. Необходим видеоряд, выстроенный методически грамотно и технически квалифицированно, формат «говорящей головы» педагогически малоэффективен. Лектор обязан выполнять функции помощника восприятия информации, а не поставщика. Принимающая сторона также должна иметь минимум знаний о средствах дистанционной связи и особенностях работы.

Таким образом, для успешного использования дистанционных сервисов в обучении программированию в учебном процессе необходимо активизировать соответствующую подготовку кадров, развивать новые компетенции и навыки тьюторов с целью реализации различных форм учебных занятий и преодоления проблемных областей в контексте современной рациональности - «Эффективность, оперативность, включенность». Этому способствует и практическая деятельность - непосредственное участие в видеоконференциях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Концепция информатизации системы образования Республики Беларусь на период до 2020 года. Электронный ресурс. Режим доступа <https://edu.gov.by/statistics/informatizatsiya-obrazovaniya/> Дата доступа 19.11.2020

2 Стратегия «Наука и технологии». Электронный ресурс. Режим доступа http://nasb.gov.by/congress2/strategy_2018-2040.pdf Дата доступа 19.11.2020

3 Государственная программа инновационного развития Республики Беларусь на 2016-2020 г.г. Электронный ресурс. Режим доступа <https://www.belta.by/president/view/lukashenko-v-belarusi-za-pjat-let-vypusk-nauchno-tehnicheskoy-i-innovatsionnoj-produktsii-vozros-bolee-280221-2017/> Дата доступа 19.01.2021

4 Министерство образования Республики Беларусь, дополнительное образование детей и молодежи. Электронный ресурс. Режим доступа <https://edu.gov.by/sistema-obrazovaniya/upravlenie-raboty/molodezhi/> Дата доступа 19.01.2021

5 Образовательный проект Парка высоких технологий и Министерства образования Республики Беларусь, электронный ресурс. Режим доступа <https://scratch.by/teachers/curriculum/> Дата доступа 19.01.2021

6. Зачем программисты изучают несколько языков. Электронный ресурс. Режим доступа <https://vc.ru/dev/159465-zachem-programmisty-izuchayut-neskolko-yazykov> Дата доступа 13.02.2021

7. Мухаметзянов И.Ш. Медицинские и психологические условия формирования и функционирования информационно-ком-

муникационной образовательной среды учебного заведения.//Казанский педагогический журнал. - 2009. - №4. - С.92-96.

8. Костиков ^//.Видеоконференц-связь: проблемы и пути их решения.//Высшее образование в России. - 2009. - №8. -С.104-108.

9. Усков В.Л. Информационно-коммуникационные технологии в образовании / Усков В.Л., Иванников А.Д., Усков А.В. // Телематика: XIV Всероссийская научно-методическая конференция. - 2007.

Приложение 1

Тематический план объединения по интересам «Scratch» 1-го года обучения (1-2 класс)

№ п/п	Название разделов, тем	Количество часов		
		Всего часов	В том числе	
			теорети ческих	практи ческих
1	Вводное занятие	2	2	0
2	Тема 1. Знакомство с компьютером	2	1	1
3	Тема 2. Работа с текстом	6	3	3
4	Тема 3. Первоначальное представление о глобальной сети Интернет. Правила безопасной работы в сети Интернет	4	2	2
5	Тема 4. Первичное знакомство со средой программирования Scratch. Основы анимации в Scratch	8	4	4
6	Тема 5. Интерактивная анимация и взаимодействие объектов	14	7	7
7	Тема 6. Сенсоры	18	9	9
8	Тема 7. Графический редактор Scratch. Создание авторских спрайтов и фонов	6	3	3
9	Тема 8. Звук и музыка в анимации	2	1	1
10	Тема 9. Создание творческого проекта по итогам курса	8	4	4
11	Заключительное занятие	2	1	1
	Всего:	72	37	35

Тематический план объединения по интересам «Code Club» с изучением нескольких языков программирования

№ п/п	Темы	Количество часов			
		Всего часов	В том числе		
			Теорети- ческих	Практи- ческих	
1.	Вводное занятие	2	1	1	
	Раздел 1. C++	16			
2	введение в C++	2	1	1	
3	Понятие алгоритма, виды алгоритмов.	2	1	1	
4	Основные характеристики языка C(C++). Алфавит языка.	2	1	1	
5	Структура простой программы.	2	1	1	
6	Операции отношения, логические, битовые операции.	2	1	1	
7	Программирование циклических структур алгоритмов.	2	1	1	
8	Структурированные типы.	2	1	1	
9	Указатели. Операции над указателями.	2	1	1	
	Раздел 2 Python.	44			
10	Введение в Python.	2	1	1	
11	Стандартная библиотека Python.	2	1	1	
12	Редакторы коды для Python.	2	1	1	
13	Инсталлируем дистрибутив Anaconda.	2	1	1	
14	Введение в Anaconda.	2	1	1	

15	Тестирование.	2	1	1
16	Основы Python.	6	3	3
17	Коллекции, циклы и логика в Python.	8	4	4
18	Функции и модули.	4	2	2
19	Ошибки и исключения. Автоматизированные тесты.	2	1	1
20	ООП в Python.	4	2	2
21	Модули и пакеты.	2	1	1
22	Дополнительные возможности.	6	3	3
	Раздел 3. Java.	8		
23	Введение в Java.	2	1	1
24	Синтаксис Java.	6	3	3
25	Заключительное занятие.	2	1	1
	Итого часов:	72	36	36

**Программа объединения по интересам
Программирование «Время кодить»
(базовый уровень изучения образовательной области
«Информатика»,
технический и естественно-математический профиль)**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа объединения по интересам ПРОГРАММИРОВАНИЕ «ВРЕМЯ КОДИТЬ» разработана на основе типовой программы дополнительного образования детей и молодежи (технический и естественно-математический профиль).

Требование к программе объединения по интересам ПРОГРАММИРОВАНИЕ «ВРЕМЯ КОДИТЬ» диктует век информационных технологий. Современные дети с ранних лет начинают осваивать различные компьютерные устройства и гаджеты. Для них это так же естественно, как учиться читать и писать. И компьютерная грамотность на сегодняшний день так же важна, как и владение навыками письма и чтения. Со временем эта тенденция будет лишь усиливаться. Лет через 20 неумение логически мыслить, понимать структуру кода и писать его будет восприниматься так же, как неграмотность и незнание математики сегодня.

Дети, растущие в современной информационной среде, все больше увлекаются компьютерными технологиями, интернетом и программированием. С одной стороны, это формирует ряд

преимуществ, поскольку учащиеся с юных лет легко учатся разбираться в компьютерах и электронных устройствах, программах и различных мобильных приложениях. С другой стороны, формирует противоречия, что порой учащиеся ограничиваются ролью потребителя, довольствуясь интернетом и социальными сетями, не редко в ущерб живому общению со сверстниками и взрослыми. Вместе с тем, сфера IT не ограничивается компьютерными играми, роликами на YouTube, фильмами и соцсетями, она очень обширна и многогранна.

Специалисты в этой области занимаются программированием искусственного интеллекта, разработкой «умной» техники, автоматизацией различных процессов и даже созданием компьютерных программ для полетов в космос. Практически в любой области, так или иначе, уже не обойтись без программирования и можно найти ему применение. Информационные технологии - это огромный мир, в котором детям есть что изучать, это мощный инструмент познания и развития.

Даже если учащийся не собирается посвятить свою жизнь программированию, есть целый ряд доводов, по которым это умение может оказаться полезным в будущем. Согласно последним исследованиям на рынке труда, спрос на IT специалистов во всем мире не просто велик, но и будет непременно расти. В одной только Америке, например, предлагается более 1,5 миллиона вакансий, связанных с программированием и лишь 400 000 выпускников кафедр информационных технологий. 60 % востребованных профессий связаны с компьютерными технологиями. По статистике 4 из 10 самых высокооплачиваемых работ требуют опыта программирования. Кроме того, среднестатистическая заработная плата специалистов IT сферы в 2

раза больше, чем у представителей других профессий. Поэтому учащимся уже сейчас неплохо бы задуматься об освоении профессии, которая либо напрямую, либо косвенно связана с информационными технологиями. И чем раньше учащиеся начнут обучаться этим навыкам, тем лучше.

Цель реализации программы – создание условий для изучения основ алгоритмизации и программирования, развития научно-технического и творческого потенциала учащихся путем организации их деятельности в процессе интеграции начального инженерно-технического конструирования программ.

Задачи:

ознакомление учащихся с комплексом базовых технологий, применяемых при создании программ;

реализация межпредметных связей с физикой, информатикой и математикой;

решение учащимися ряда кибернетических задач, результатом каждой из которых будет работающая программа

развитие у учащихся инженерного мышления, навыков конструирования, программирования и эффективного использования кибернетических систем;

повышение мотивации учащихся к изобретательству и созданию собственных программных комплексов.

Основной формой организации образовательного процесса при реализации предлагаемой образовательной программы дополнительного образования детей и молодежи технического профиля является занятие (теоретическое и практическое).

В образовательном процессе при реализации данной образовательной программы дополнительного образования детей и молодежи технического профиля используются, как правило, смешанные виды занятий: чередование теоретических и практических видов деятельности. Программа разработана с учетом возрастных и психологических особенностей учащихся, уровня их развития и кругозора, предназначена для организации и проведения занятий в объединениях по интересам общеобразовательных учреждений и учреждений дополнительного образования.

Занятие проводится 2 раза в неделю по 2 академических (45 минут) часа. Общее количество часов в год – 144, при этом неукоснительно должны соблюдаться требования к организации труда и отдыха детей, правила безопасной работы и охраны труда.

Для организации работы объединения по интересам необходимо благоустроенное помещение (класс для занятий объединения по интересам, лаборатория и т.д.), оборудованное персональными компьютерами/ноутбуками с установленным необходимым программным обеспечением. В процессе практических занятий рекомендуется проводить физкультминутки, направленные на активацию дыхания, кровообращения и активный отдых группы мышц, задействованных при основной деятельности

Срок реализации программы составляет 1 год.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Темы	Количество часов	
		О	В

			ТОМ ЧИСЛЕ	
			оретиче	рактиче
1.	Вводное занятие	2	1	1
	Раздел 1. C++	18		
2	Введение в C++	2	1	1
3	Понятие алгоритма, виды алгоритмов.	2	1	1
4	Основные характеристики языка C(C++). Алфавит языка.	2	1	1
5	Структура простой программы.	2	1	1
6	Операции отношения, логические, битовые операции.	2	1	1
7	Программирование циклических структур алгоритмов.	2	1	1
8	Структурированные типы.	2	1	1
9	Указатели. Операции над указателями.	2	1	1
10	Функции. Прототипы функций.	2	1	1
	Раздел 2 Python	88		
11	Введение в Python	2	1	1
12	Стандартная библиотека Python	2	1	1
13	Редакторы коды для Python	2	1	1
14	Инсталлируем дистрибутив Anaconda	2	1	1
15	Введение в Anaconda	2	1	1
16	Проблемы запуска Jupyter Notebook через Anaconda Navigator	2	1	1
17	Тестирование	2	1	1
18	Основы Python	14	7	7
19	Коллекции, циклы и логика в Python	16	8	8

20	Функции и модули	10	5	5
21	Ошибки и исключения. Автоматизированные тесты	4	2	2
22	ООП в Python	10	5	5
23	Модули и пакеты	4	2	2
24	Дополнительные возможности	16	8	8
	Раздел 3. Java	34		
25	Введение в Java	2	1	1
26	Синтаксис Java	12	6	6
27	Объектно-ориентированное программирование. ООП	14	7	7
28	Java продвинутый уровень	6	3	3
29	Заключительное занятие	2	1	1
	Итого часов:	144	72	72

СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНОЙ ОБЛАСТИ

Вводное занятие. Роль и значение естественно-математической образовательной области. Правила безопасного поведения учащихся на занятиях, во время проведения конференций, выставок, конкурсов, слетов. Правила работы на компьютере (2 часа)

Раздел 1. C++ (18 часов)

Понятие алгоритма, виды алгоритмов. Составление линейных, разветвляющихся, циклических и итерационных алгоритмов. Разработка блок-схем циклических и итерационных структур алгоритмов. Программирование циклических и итерационных алгоритмов. (2 часа)

Основные характеристики языка C(C++). Алфавит языка. Идентификаторы. Базовые типы данных. Целый тип. Вещественный тип. Представление данных целого и вещественного типа в памяти. Символьный тип, перечисления. Преобразование типов.

Интегрированная среда (Microsoft C++). Создание простой программы на C++, ее ввод и отладка. Программирование линейных и разветвляющихся алгоритмов(2 часа). Л.3, с. 15-20,38-45,129-135; Л.7, с. 42-53, с. 58-68; Л. 10, с. 20-25, с. 63-69; Л. 4, с. 43.

Структура простой программы. Организация ввода/вывода (printf(), scanf(), putchar(), getchar(), puts(), gets() и др.). Операции языка C (C++). Операция присваивания. Арифметические операции. Л.4, с. 147-155; Л.7, с.54-58. (2 часа).

Операции отношения, логические, битовые операции. Условная операция, операция сдвига. Приоритет операций. Операции приведения типов. Программирование разветвляющихся алгоритмов. Условный оператор. Оператор перехода. Пустой оператор. Составной оператор. Работа с примером: в какой десяток попадает число, сколько букв, цифр в тексте. (2 часа). Л.3, с. 42, с. 53-60; Л.7, с. 68-71; Л.10, с.42-63, с. 69-74; Л.4, с. 56-64.

Программирование циклических структур алгоритмов. Операторы цикла. Оператор цикла с параметром. Оператор цикла с постусловием, с предусловием. В заданном диапазоне нахождение простых чисел, совершенных чисел. Объединение рассортированных массивов. Оператор выбора, break, return, exit, continue. Вычисление рядов. (2 часа). Л.3, с. 50-52, с. 81-83; Л.7, с. 71-75, с. 172-182; Л.10, с. 74-78, с. 108-121, Л.4, с. 64-71.

Структурированные типы. Одномерные массивы, их инициализация. Работа с многомерными массивами. Определить частоту встречаемости букв латинского алфавита, цифр в тексте. Транспонирование, перемножение матриц, сортировка строк матрицы.

Обработка массивов арифметических данных. Указатели. (2 часа). Л.3, с. 84-90; Л.7, с. 75-85; Л.10, с. 28-31; Л.4 30-32.

Указатели. Операции над указателями. Связь между указателями и массивами. Трехмерный массив рассортировать, используя указатели типа: int, void. Сортировка массивов данных (2 часа). Л.3, с. 88-97; Л.7, с. 87-97; Л.10, с.121-132, Л.4, с. 94-104.

Функции. Прототипы функций. Передача параметров, стек. Структура сложной программы. Найти сумму элементов каждой строки матрицы. Передача массивов в функции. Принадлежность числа рассортированному массиву чисел. Функции с произвольным числом параметров. (2 часа) Л. 3, с. 64-69, Л.7, с. 97-110; Л.4, с. 71- 82.

Раздел 2 Python (88 часов)

Введение в Python (2 часа)

Стандартная библиотека Python (2 часа)

Редакторы коды для Python (2 часа)

Инсталлируем дистрибутив Anaconda (2 часа)

Введение в Anaconda (2 часа)

Проблемы запуска Jupyter Notebook через Anaconda Navigator (2 часа)

Тестирование (2 часа)

Основы Python (14 часов)

Обзор основных типов данных Числа и элементарная математика

Упражнение по написанию кода: Элементарная арифметика.

Упражнение по написанию кода: Объявление переменных bool и None.

Тип string Упражнение по написанию кода: Работа со строками.

Функции string. Форматирование строк. Операторы сравнения.

Операции над файлами Строки и байты: str, bytes, bytearray.
Тестирование. (4 часа.) Л.3, с 101-105; Л.4, с. 104-110; Л.7, с. 85- 87.

Коллекции, циклы и логика в Python (16 часов)

list – список. dict – словарь. OrderedDict vs dict. tuple – кортеж.
namedtuple - именованные кортежи. Логика с условиями. set – множество
Цикл for list comprehension. Цикл while, continue, break.
Генераторы. Игра «Угадай число». Игра в палочки. Тестирование. Л.3,
с. 69-80; Л.7, с. 113-120; Л.10, с. 98-107; Л.4, с. 77-87.

Функции и модули (10 часов)

Помощь по функциям Встроенные функции Основы функций
Лямбды Вложенные функции и область видимости переменных
Декораторы Декоратор @wraps Тестирование Парсинг римских чисел.
Л. 3, с. 88-90; Л. 10, с. 91-95.

Ошибки и исключения. Автоматизированные тесты (4 часа)

Основы обработки ошибок. Выброс исключений. Кастомные типы
исключений. Юнит-тестирования. Л.3, с.108-112, с. 116-117; Л.7, с. 150-
165; Л.10, с. 31-42; Л.4,125-135

ООП в Python (10 часов)

Основы классов. Атрибуты и методы. Константы. Защищённые и
приватные атрибуты. Свойства. Статические методы - @staticmethod,
@classmethod. Наследование и полиморфизм. Множественное
наследование. Миксины. Абстрактный класс и модуль ABC.
Магические методы. Игра «Крестики-нолики». Л.3, с. 123-128; Л.10, с.
34-42; Л.4, с. 143-145.

Модули и пакеты (4 часа)

PyPi и Pip. Модули и пакеты. __name__ и __main__. Тестирование.
Л.3, с. 112-116, с. 118-121; Л.7, с. 160-161; Л.4, с. 145-147.

Дополнительные возможности (16 часов)

Отладка. Реализуем Stack. datetime - даты и время. Singleton Design Pattern: __new__ and __init__.Pickle – консервирование. repr and str, eq and ne, eval. Deep copy vs Shallow copy. Enum – перечисления. Работаем с JSON. Модуль intertools. Интроспекция. Модуль requests. Управление памятью. Л.4, с.113-117; Л.10, с. 93-95; Л.7, с. 107-110.

Раздел 3. Java (34 часа)

Введение в Java (2 часа)

Установка среды разработки. Первая программа Hello World

Синтаксис Java (12 часов)

Переменные. Условный оператор. Типы данных. Целые числа. Типы данных. Примитивные типы. Массивы. Закрываем пробелы. String и Switch Case. Л.7, с. 110-113; Л.10, с. 87-95; Л.4, с. 117-120.

Объектно-ориентированное программирование. ООП (14 часов)

Введение в ООП. Введение в методы. Параметризованные методы. Конструкторы. Перегрузка методов и конструкторов. Объекты в качестве параметров и возврат объектов. Модификаторы доступа. Ключевые слова static и final и аргументы переменной длины. Создаем собственную коллекцию. Ключевое слово import, оболочки типов и продвинутая работа со строками. Встроенные коллекции. Наследование. Интерфейсы. Л. 4, с. 155-158, с. 169-170; Л. 10, с.187-191, с. 197-201. 2 Л.4, с. 160-161.

Java продвинутый уровень (6 часов)

Обработка исключений. Методы String.format(), toString() и случайные числа. Многопоточное программирование. Л.4, с. 164-168.

Итоговое занятие. Подведение итогов работы в соответствии со сроками реализации программы. Итоговые выставки работ, соревнования, конкурсы и др. Итоговая аттестация учащихся. (2 часа).

ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ

Учащиеся мотивированы, подготовлены и профессионально ориентированы для возможного продолжения учебы в ВУЗах и последующей работы на предприятиях по специальностям, связанным с программированием, созданы условия для работы с разными источниками информации, оценивания их и на этой основе формулирования собственного мнения, суждения, оценки.

В результате реализации программы обучения учащиеся должны знать:

теоретические основы алгоритмизации задач и проектирования программ;

создание программ на алгоритмических языках;

основы организации вычислительных процессов на ЭВМ;

правила безопасной работы в компьютерном классе;

применение проектов в различных сферах жизни и деятельности человека;

назначении основных устройств компьютера;

роли сети Интернет в жизни человека;

алгоритмах, о видах и способах записи алгоритмов;

функциональном устройстве программных сред и основных структурных элементах пользовательского интерфейса;

назначении и использовании основных блоков команд, состояний, программ;

правилах сохранения программного продукта и необходимости присвоения правильного имени.

В результате реализации программы обучения учащиеся должны уметь:

- программировать;
- отлаживать и выполнять на ЭВМ конкретные задачи с использованием современных методов программирования;
- запускать программы и корректно завершать их работу;
- создавать простейшие программные продукты;
- осуществлять поиск информации в сети Интернет;
- работать с библиотеками программной среды;
- создавать и редактировать свой проект;
- владеть специальными теоретическими знаниями по изучению содержания образовательных областей, тем;
- уметь публично предъявлять собственные результаты на конференциях, конкурсах, выставках и др.;
- владеть умениями и навыками технического творчества и исследовательской деятельности.

ФОРМЫ ПОДВЕДЕНИЯ ИТОГОВ РЕАЛИЗАЦИИ ПРОГРАММЫ

Формами подведения итогов реализации программы могут быть: «летопись» объединения по интересам (видео- и фотоматериалы); портфолио творческих достижений объединения по интересам (грамоты, дипломы, сертификаты и др.); отзывы учащихся о выставках, экскурсиях и других мероприятиях, в которых они принимали участие или которые посетили; открытые занятия; отчетные выставки работ

учащихся; презентации исследовательских работ; защита проектов; отчетные тематические вечера; заключительные занятия; мастер-классы; составление сборника лучших исследовательских работ учащихся и др.

Для подведения итогов реализации программы может использоваться также оценка результатов, полученных во время публичного предъявления учащимися собственных результатов деятельности на конференциях, конкурсах, выставках и др.

ФОРМЫ И МЕТОДЫ РЕАЛИЗАЦИИ ПРОГРАММЫ

Реализация данной программы требует традиционных и нетрадиционных форм и методов работы, направленных на создание оптимальных условий для достижения ожидаемых результатов в обучении, воспитании, развитии учащихся, удовлетворении их индивидуальных возможностей, потребностей, интересов, раскрытия личностного потенциала каждого.

Для реализации программы используются следующие формы обучения: групповые и индивидуальные.

При реализации данной программы предпочтительной является индивидуально-групповая форма обучения. Педагог дополнительного образования обучает учащихся разновозрастного состава, различного уровня подготовленности, поочередно работая с каждым и предлагая им практические задания. Рекомендуется также парная работа над выполнением конкретного задания.

На практических занятиях с применением компьютерного оборудования, технических устройств, приборов и механизмов, требующих повышенного внимания при их использовании, которые

невозможно эксплуатировать при групповой форме обучения, рекомендуется применять индивидуальные формы обучения.

Методы обучения:

объяснительно-иллюстративный метод обучения – метод, при котором обучающиеся получают знания на занятиях во время тренинга, диспута, дискуссии, семинара, консультации, инструктажа, обсуждения; из учебной, технической, справочной литературы через мультимедийные и экранные пособия, интернет;

репродуктивный метод обучения – метод, в котором применение изученного осуществляется на основе образца или правила. Здесь деятельность учащихся носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях;

метод проблемного обучения – это совокупность действий, приемов, направленных на усвоение знаний через активную мыслительную деятельность, содержащую постановку и решение продуктивно-познавательных вопросов и задач, имеющих противоречия (учебные или реальные), способствующих успешной реализации целей учебно-воспитательного процесса;

эвристический метод обучения применяется для организации активного поиска решения выдвинутых в обучении или самостоятельно сформулированных технических и познавательных задач; для стимулирования активного поиска решения поставленных задач используются элементы теории решения изобретательских задач (ТРИЗ), применяются методы: мозговой штурм, проб и ошибок, синектики, морфологический анализ, фокальных объектов, контрольных вопросов, аналогий, объединения, секционирования, модифицирования,

копирования прототипов, оптимального проектирования, унификации, агрегатирования, модификации, стандартизации, инверсии, конструирования «КАРУС» и др.;

исследовательский метод обучения – метод, в котором после анализа материала, постановки проблем и задач, краткого инструктажа учащиеся самостоятельно изучают литературу, источники, проводят эксперименты, наблюдения и измерения, моделируют и конструируют, разрабатывают программные продукты, решают прикладные задачи.

Формы воспитания: массовые, групповые, индивидуальные.

Воспитательные мероприятия при реализации программы проводятся в соответствии с Концепцией непрерывного воспитания детей и учащейся молодежи с учетом основных составляющих воспитания детей и учащейся молодежи: идеологическое воспитание; гражданское и патриотическое воспитание; духовно-нравственное воспитание; поликультурное воспитание, экономическое воспитание; воспитание культуры безопасности жизнедеятельности; эстетическое воспитание; воспитание психологической культуры; воспитание культуры здорового образа жизни; экологическое воспитание; семейное и гендерное воспитание; трудовое и профессиональное воспитание; воспитание культуры быта и досуга.

ЛИТЕРАТУРА И ИНФОРМАЦИОННЫЕ РЕСУРСЫ

Основная литература

1. Брябрин В.М. Программное обеспечение персональных ЭВМ. - М.: Наука, 1988.
2. Фигурнов В.Э. IBM PC для пользователя. - М.: Финансы и статистика, 1991.

3. Керниган Б., Ритчи Д.,Фьэр А. Язык программирования Си. Задачи по языку Си. - М.: Финансы и статистика, 1985.
4. Керниган Б., Ритчи Д. Язык программирования Си. - М.: Финансы и статистика, 1992.
5. Шилд Г. Программирование на Borland C++ для профессионалов. - Мн.:ООО «Попури», 1998.
6. Сэвитч У. C++ в примерах. -М.: ЭКОМ, 1997.
7. Касаткин А.И, Вальвачев А.Н. От TURBO C к Borland C++. - Мн.: Выш.шк.,1992.
8. Касаткин А.И. Управление ресурсами, - Мн.: Выш. шк., 1992.
9. Касаткин А.И. Системное, программирование. - Мн.: Выш.шк., 1993.
10. Романовская Л.М., Русс Т.В., Свитковский С.Г., Программирование в среде Си для ПЭВМ. - М.: Финансы и статистика, 1992.
11. Бочков С.О., Субботин Д.М. Язык программирования Си для персональных компьютеров. - М.: Радио и связь, 1990.
12. Луцик Ю.А., Ковальчук А.М. и др. Методическое пособие по курсу «Основы алгоритмизации и программирования» -Мн.: БГУИР, 1999.

Дополнительная литература

1. Демидович Е.М. и др. Практикум по курсу «Программирование» для студентов специальности «Вычислительные машины, системы и сети» . - Мн. : БГУИР,1995.
2. Демидович Е.М. Учебное пособие по курсу «Основы алгоритмизации и программирования» для студентов специальности «Вычислительные машины, системы и сети». - Мн.: БГУИР, 1999.

3. Демидович Е. М. и др. Методическое пособие по курсу «Основы алгоритмизации и программирования» для студентов специальности «Вычислительные машины, системы и сети». -Мн.: БГУИР, 1999. 4.4. Глухова Л.А. и др. Методические указания по вычислительной практике и самостоятельной работе по дисциплине «Программирование» и «Конструирование программ и языки программирования» для студентов специальности «Вычислительные машины, комплексы, системы и сети» «Программное обеспечение ЭВМ и автоматизированных систем» и слушателей спецфакультета переподготовки по направлению «Микропроцессорные системы», часть 1. -Мн.: МРТИ, 1989, 33с.
4. Фадеева Е.П. и др. Методические указания по вычислительной практике и самостоятельной работе по курсам «Программирование» и «Конструирование программ и языки программирования» для студентов специальности «Вычислительные машины, комплексы, системы и сети», «Программное обеспечение ЭВМ и автоматизированных систем» и слушателей спецфакультета переподготовки по направлению «Микропроцессорные системы», часть 2. - Мн.: МРТИ, 1990.
5. Кнут Д.Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. -720 с.: ил.
6. Кнут Д.Э. Искусство программирования, том 2. Получисленные алгоритмы, 3-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. - 832 с.: ил.
7. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. - 832 с.: ил.

ПРОГРАММА ОБЪЕДИНЕНИЯ ПО ИНТЕРЕСАМ

«ПРОГРАММИРОВАНИЕ C++»

(базовый уровень изучения образовательной области,
технический и естественно-математический профиль)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа объединения по интересам «ПРОГРАММИРОВАНИЕ C++» разработана на основе типовой программы дополнительного образования детей и молодежи (технический и естественно-математический профиль).

Требование к программе объединения по интересам «ПРОГРАММИРОВАНИЕ C++» диктует век информационных технологий. Современные дети с ранних лет начинают осваивать различные компьютерные устройства и гаджеты. Для них это так же естественно, как учиться читать и писать. И компьютерная грамотность на сегодняшний день так же важна, как и владение навыками письма и чтения. Со временем эта тенденция будет лишь усиливаться. Лет через 20 неумение логически мыслить, понимать структуру кода и писать его будет восприниматься так же, как неграмотность и незнание математики сегодня.

Дети, растущие в современной информационной среде, все больше увлекаются компьютерными технологиями, интернетом и программированием. С одной стороны, это формирует ряд преимуществ, поскольку учащиеся с юных лет легко учатся разбираться в компьютерах и электронных устройствах, программах и различных

мобильных приложениях. С другой стороны, формирует противоречия, что порой учащиеся ограничиваются ролью потребителя, довольствуясь интернетом и социальными сетями, не редко в ущерб живому общению со сверстниками и взрослыми. Вместе с тем, сфера IT не ограничивается компьютерными играми, роликами на YouTube, фильмами и соцсетями, она очень обширна и многогранна.

Специалисты в этой области занимаются программированием искусственного интеллекта, разработкой «умной» техники, автоматизацией различных процессов и даже созданием компьютерных программ для полетов в космос. Практически в любой области, так или иначе, уже не обойтись без программирования и можно найти ему применение. Информационные технологии - это огромный мир, в котором детям есть что изучать, это мощный инструмент познания и развития.

Даже если учащийся не собирается посвятить свою жизнь программированию, есть целый ряд доводов, по которым это умение может оказаться полезным в будущем. Согласно последним исследованиям на рынке труда, спрос на IT специалистов во всем мире не просто велик, но и будет непременно расти. В одной только Америке, например, предлагается более 1,5 миллиона вакансий, связанных с программированием и лишь 400 000 выпускников кафедр информационных технологий. 60 % востребованных профессий связаны с компьютерными технологиями. По статистике 4 из 10 самых высокооплачиваемых работ требуют опыта программирования. Кроме того, среднестатистическая заработная плата специалистов IT сферы в 2 раза больше, чем у представителей других профессий. Поэтому учащимся уже сейчас неплохо бы задуматься об освоении профессии,

которая либо напрямую, либо косвенно связана с информационными технологиями. И чем раньше учащиеся начнут обучаться этим навыкам, тем лучше.

Цель реализации программы – создание условий для изучения основ алгоритмизации и программирования, развития научно-технического и творческого потенциала учащихся путем организации их деятельности в процессе интеграции начального инженерно-технического конструирования программ.

Задачи:

ознакомление учащихся с комплексом базовых технологий, применяемых при создании программ;

реализация межпредметных связей с физикой, информатикой и математикой;

решение учащимися ряда кибернетических задач, результатом каждой из которых будет работающая программа

развитие у учащихся инженерного мышления, навыков конструирования, программирования и эффективного использования кибернетических систем;

повышение мотивации учащихся к изобретательству и созданию собственных программных комплексов.

Основной формой организации образовательного процесса при реализации предлагаемой образовательной программы дополнительного образования детей и молодежи технического профиля является занятие (теоретическое и практическое).

В образовательном процессе при реализации данной образовательной программы дополнительного образования детей и молодежи технического профиля используются, как правило,

смешанные виды занятий: чередование теоретических и практических видов деятельности. Программа разработана с учетом возрастных и психологических особенностей учащихся, уровня их развития и кругозора, предназначена для организации и проведения занятий в объединениях по интересам общеобразовательных учреждений и учреждений дополнительного образования.

Занятие проводится 2 раза в неделю по 2 академических (45 минут) часа. Общее количество часов в год – 144, при этом неукоснительно должны соблюдаться требования к организации труда и отдыха детей, правила безопасной работы и охраны труда.

Для организации работы объединения по интересам необходимо благоустроенное помещение (класс для занятий объединения по интересам, лаборатория и т.д.), оборудованное персональными компьютерами/ноутбуками с установленным необходимым программным обеспечением. В процессе практических занятий рекомендуется проводить физкультминутки, направленные на активацию дыхания, кровообращения и активный отдых группы мышц, задействованных при основной деятельности

Срок реализации программы составляет 1 год.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Темы	Количество часов		
		Всего	часов	В том числе

			Теоретических	Практических
1.	Вводное занятие	2	1	1
2.	Понятие алгоритма, виды алгоритмов. Составление линейных, разветвляющихся, циклических и итерационных алгоритмов. Разработка блок-схем циклических и итерационных структур алгоритмов Программирование циклических и итерационных алгоритмов	6	1	5
3.	Основные характеристики языка C(C++). Алфавит языка. Идентификаторы. Базовые типы данных. Целый тип. Вещественный тип. Представление данных целого и вещественного типа в памяти. Символьный тип, перечисления. Преобразование типов. Интегрированная среда (Microsoft C++). Создание простой программы на Си, ее ввод и отладка. Программирование линейных и разветвляющихся алгоритмов	4	1	3
4.	Структура простой программы. Организация ввода/вывода (printf(), scanf(), putchar(), getchar(), puts(), gets() и др.). Операции языка C (C++). Операция присваивания. Арифметические операции.	2	1	1
5.	Операции отношения, логические, битовые операции. Условная операция, операция сдвига. Приоритет операций. Операции приведения типов. Программирование разветвляющихся алгоритмов. Условный оператор. Оператор перехода. Пустой	2	1	1

	оператор. Составной оператор. Работа с примером: в какой десяток попадает число, сколько букв, цифр в тексте.			
6.	Программирование циклических структур алгоритмов. Операторы цикла. Оператор цикла с параметром. Оператор цикла с постусловием. Оператор цикла с предусловием. В заданном диапазоне найти простые числа, совершенные числа. Объединение рассортированных массивов. Оператор выбора, break, return, exit, continue. Вычисление рядов.	4	1	3
7.	Структурированные типы. Одномерные массивы, их инициализация. Работа с многомерными массивами. Определить частоту встречаемости букв латинского алфавита, цифр в тексте. Транспонирование, перемножение матриц, сортировка строк матрицы. Обработка массивов арифметических данных. Указатели	4	1	3
8.	Указатели. Операции над указателями. Связь между указателями и массивами. Трехмерный массив рассортировать, используя указатели типа: int, void. Сортировка массивов данных	4	1	3
9.	Функции. Прототипы функций. Передача параметров, стек. Структура сложной программы. Найти сумму элементов каждой строки матрицы. Передача массивов в функции. Принадлежит ли число рассортированному массиву чисел. Функции с произвольным числом параметров	6	1	5
10.	Строковые данные. Реализация функций для работы со строками: reverse, comp, copy, length, вхождение строки s1 в s2, cat, puts, gets. Сравнение и сортировка строк в	4	1	3

	алфавитном порядке. Использование указателей при работе со строками			
11.	Создание, корректировка, копирование текстовых файлов Локальные и глобальные переменные. Классы памяти. Автоматические, внешние, статистические и регистровые переменные.	4	2	2
12.	Рекурсии: $n!$, числа Фибоначчи, перевод чисел из любой системы счисления (<36) в 10 с/с. Не объявляя массива, ввести группу данных и вывести их в обратном порядке. Реверс строки, используя рекурсию. Рекурсивное использование функций	6	1	5
13.	Структуры, их объявление, инициализация, способы обращения к элементам. Размещение в памяти. Обработка результатов сдачи сессии, определение среднего балла. Рассортировка списка по среднему баллу	2	1	1
14.	Объединения. Объявление, инициализация, обращение к элементам, размещение в памяти. Вычисление площади геометрических фигур	2	1	1
15.	Поля битов в структурах. Размещение в памяти. Инициализация полей бит. Вывести битовое представление символов. Используя поля битов, нахождение остатка от деления на 2, 4, 8, 16	2	1	1
16.	Методы сортировки данных в оперативной памяти: Шелла, Хоора, сравнения соседних элементов, пузырьковая сортировка	4	2	2
17.	Задание информации в командной строке. Вывод содержимого командной строки, выполнение заданной функции с заданным списком параметров. Использование в функциях информации командной	6	1	5

	строки			
18.	Указатели на функции. Методом деления отрезка пополам нахождение корня уравнения. Вычисление значения интеграла заданным в командной строке методом (трапеций, прямоугольников, Симпсона). Массивы указателей на функции. Указатели типа <code>near</code> и <code>far</code> .	4	2	2
19.	Файлы. Бинарные и текстовые файлы. Библиотечные функции для открытия и закрытия файлов. Стандартные файлы. Файлы прямого и последовательного доступа. Создание файла об учащиххся и копирование его в другой файл. Библиотечные функции организации прямого доступа к файлу Библиотечные функции для работы с текстовыми файлами. Печать файлов в прямом и обратном направлении.	4	2	2
20.	Неформатированный ввод/вывод с использованием буферов, работа с бинарными файлами, обработка блоками Бинарные файлы. Прямой доступ к файлам	6	1	5
21.	Корректировка информации в файлах, заданных в командной строке. Создание файлов, рассортированных в указанном порядке, и сливание их в упорядоченный файл	2	1	1
22.	Функции с переменным числом параметров. Использование переменных типа <code>va_list</code> , указателей на тип <code>void</code> и стандартные типы. Нахождение максимального элемента из списка переменной длины. Обработка списка переменной длины из данных различного типа, из строк, массивов.	4	2	2
23.	Динамическое использование памяти, модели памяти, библиотечные функции. Стеки: создание, просмотр,	6	2	4

	удаление верхнего, удаление заданного, сортировка (перемещая элементы, перемещая указатели). Обработка списковых структур данных. Стек.			
24.	Очереди, кольца: создание, просмотр, удаление, добавление элементов. Двухнаправленные списки Обработка списковых структур данных (очередь, кольцо)	8	1	7
25.	Бинарные деревья: создание, обход (используя рекурсии, без них), добавление (используя рекурсии, без них), удаление элементов. Построение симметричных деревьев Обработка списков в виде бинарных деревьев	8	2	6
26.	Проверка правильности расстановки скобок в арифметическом выражении. Построение обратной польской записи. Потоки Создание и ведение справочника наличия деталей на складе. Функции хеширования. Создание телефонного справочника. Сложные объекты, их объявление. Нахождение кратчайшего пути на графе. Сжатие информации	6	3	3
27.	Модели памяти. Препроцессор. Многоуровневые ссылки. Директивы препроцессора. Условная компиляция	2	1	1
28.	Понятие объекта и фундаментальные характеристики ООП (инкапсуляция, наследование, полиморфизм). Определение классов (class, struct, union). Элементы данных и методы класса. Защита элементов класса и атрибуты доступа. Объявление и определения методов класса. Вызов членов класса. Конструкторы и деструкторы	4	2	2
29.	Понятие конструктора и деструктора и их свойства, explicit конструктор Скрытый указатель this. Массив объектов. Классы, объекты, конструктор и деструктор,	8	2	6

	this-указатель			
30.	Инициализация объектов. Автоматические, динамические и статические объекты. Операторы new и delete. Распределение памяти. Указатель this. Конструкторы и деструкторы Организация ввода/вывода в C++, манипуляторы, операторы new и delete Базовые и производные классы Организация ввода/вывода в C++ с использованием объектов cin и cout. Манипуляторы	12	2	10
31.	Полиморфизм. Перегрузка функций. Особенности использования перегруженных функций. Перегрузка конструкторов	4	1	3
32.	Итоговое занятие	2	0	2
	Всего часов	144	44	100

СОДЕРЖАНИЕ ОБРАЗОВАТЕЛЬНЫХ ОБЛАСТЕЙ

Вводное занятие. Роль и значение естественно-математической образовательной области. Правила безопасного поведения учащихся на занятиях, во время проведения конференций, выставок, конкурсов, слетов. Правила работы на компьютере (2 часа)

Понятие алгоритма, виды алгоритмов. Составление линейных, разветвляющихся, циклических и итерационных алгоритмов. Разработка блок-схем циклических и итерационных структур алгоритмов. Программирование циклических и итерационных алгоритмов. (6 часов)

Основные характеристики языка C(C++). Алфавит языка. Идентификаторы. Базовые типы данных. Целый тип. Вещественный тип. Представление данных целого и вещественного типа в памяти. Символьный тип, перечисления. Преобразование типов. Интегрированная среда (Microsoft C++). Создание простой программы

на C++, ее ввод и отладка. Программирование линейных и разветвляющихся алгоритмов. (4 часа). Л.3, с. 15-20,38-45,129-135; Л.7, с. 42-53, с. 58-68; Л. 10, с. 20-25, с. 63-69; Л. 4, с. 43.

Структура простой программы. Организация ввода/вывода (printf(), scanf(), putchar(), getchar(), puts(), gets() и др.). Операции языка C (C++). Операция присваивания. Арифметические операции. Л.4, с. 147-155; Л.7, с.54-58. (2 часа).

Операции отношения, логические, битовые операции. Условная операция, операция сдвига. Приоритет операций. Операции приведения типов. Программирование разветвляющихся алгоритмов. Условный оператор. Оператор перехода. Пустой оператор. Составной оператор. Работа с примером: в какой десяток попадает число, сколько букв, цифр в тексте. (2 часа). Л.3, с. 42, с. 53-60; Л.7, с. 68-71; Л.10, с.42-63, с. 69-74; Л.4, с. 56-64.

Программирование циклических структур алгоритмов. Операторы цикла. Оператор цикла с параметром. Оператор цикла с постусловием, с предусловием. В заданном диапазоне нахождение простых чисел, совершенных чисел. Объединение рассортированных массивов. Оператор выбора, break, return, exit, continue. Вычисление рядов. (4 часа). Л.3, с. 50-52, с. 81-83; Л.7, с. 71-75, с. 172-182; Л.10, с. 74-78, с. 108-121, Л.4, с. 64-71.

Структурированные типы. Одномерные массивы, их инициализация. Работа с многомерными массивами. Определить частоту встречаемости букв латинского алфавита, цифр в тексте. Транспонирование, перемножение матриц, сортировка строк матрицы. Обработка массивов арифметических данных. Указатели. (4 часа). Л.3, с. 84-90; Л.7, с. 75-85; Л.10, с. 28-31; Л.4 30-32.

Указатели. Операции над указателями. Связь между указателями и массивами. Трехмерный массив рассортировать, используя указатели типа: int, void. Сортировка массивов данных. (4 часа). Л.3, с. 88-97; Л.7, с. 87-97; Л.10, с.121-132, Л.4, с. 94-104.

Функции. Прототипы функций. Передача параметров, стек. Структура сложной программы. Найти сумму элементов каждой строки матрицы. Передача массивов в функции. Принадлежность числа рассортированному массиву чисел. Функции с произвольным числом параметров. (6 часов.) Л. 3, с. 64-69, Л.7, с. 97-110; Л.4, с. 71- 82.

Строковые данные. Реализация функций для работы со строками: reverse, comp, copy, length, вхождение строки s1 в s2, cat, puts, gets. Сравнение и сортировка строк в алфавитном порядке. Использование указателей при работе со строками. (4 часа.) Л.3, с 101-105; Л.4, с. 104-110; Л.7, с. 85- 87.

Создание, корректировка, копирование текстовых файлов. Локальные и глобальные переменные. Классы памяти. Автоматические, внешние, статистические и регистровые переменные. (4 часа). Л.3, с. 69-80; Л.7, с. 113-120; Л.10, с. 98-107; Л.4, с. 77-87.

Рекурсии: n!, числа Фибоначчи, перевод чисел из любой системы счисления (<36) в 10 с/с. Не объявляя массива, ввод группы данных и выводи их в обратном порядке. Реверс строки, используя рекурсию. Рекурсивное использование функций. (6 часов.) Л. 3, с. 88-90; Л. 10, с. 91-95.

Структуры, их объявление, инициализация, способы обращения к элементам. Размещение в памяти. Обработка результатов сдачи сессии, определение среднего балла. Рассортировка списка по среднему баллу.

(2 часа.) Л.3, с.108-112, с. 116-117; Л.7, с. 150-165; Л.10, с. 31-42; Л.4,125-135.

Объединения. Объявление, инициализация, обращение к элементам, размещение в памяти. Вычисление площади геометрических фигур. (2 часа). Л.3, с. 123-128; Л.10, с. 34-42; Л.4, с. 143-145.

Поля битов в структурах. Размещение в памяти. Инициализация полей бит. Вывод битового представления символов. Используя поля битов, нахождение остатка от деления на 2, 4,8,16. (2 часа). Л.3, с. 112-116, с. 118-121; Л.7, с. 160-161; Л.4, с. 145-147.

Методы сортировки данных в оперативной памяти: Шелла, Хоора, сравнения соседних элементов, пузырьковая сортировка. (4 часа).

Задание информации в командной строке. Вывод содержимого командной строки, выполнение заданной функции с заданным списком параметров. Использование в функциях информации командной строки. (6 часов). Л.4, с.113-117; Л.10, с. 93-95; Л.7, с. 107-110.

Указатели на функции. Методом деления отрезка пополам нахождение корня уравнения. Вычисление значения интеграла заданным в командной строке (методом трапеций, прямоугольников, Симпсона). Массивы указателей на функции. Указатели типа near и far. (4 часа). Л.7, с. 110-113; Л.10, с. 87-95; Л.4, с. 117-120.

Файлы. Бинарные и текстовые файлы. Библиотечные функции для открытия и закрытия файлов. Стандартные файлы. Файлы прямого и последовательного доступа. Создание файла об учащихся и копирование его в другой файл. Библиотечные функции организации прямого доступа к файлу, для работы с текстовыми файлами. Печать файлов в прямом и обратном направлении. (4 часа). Л. 4, с. 155-158, с. 169-170; Л. 10, с.187-191, с. 197-201. 2 Л.4, с. 160-161.

Неформатированный ввод/вывод с использованием буферов, работа с бинарными файлами, обработка блоками Бинарные файлы. Прямой доступ к файлам. (6 часов). Л.4, с. 164-168.

Корректировка информации в файлах, заданных в командной строке. Создание файлов, рассортированных в указанном порядке, и сливание их в упорядоченный файл. (4 часа).

Функции с переменным числом параметров. Использование переменных типа `valist`, указателей на тип `void` и стандартные типы. Нахождение максимального элемента из списка переменной длины. Обработка списка переменной длины из данных различного типа, из строк, массивов. (4 часа). Л.10, с. 91-93; Л.7, с. 99-107.

Динамическое использование памяти, модели памяти, библиотечные функции. Стеки: создание, просмотр, удаление верхнего, удаление заданного, сортировка (перемещая элементы, перемещая указатели). Обработка списковых структур данных. Стек. (6 часов). Л.7, с.211-226; Л.8, с. 184-213; Л.4, с. 135-139.

Очереди, кольца: создание, просмотр, удаление, добавление элементов. Двухнаправленные списки. Обработка списковых структур данных. Очередь. Кольцо. (8 часов). Л.7, с. 216-223.

Бинарные деревья: создание, обход и добавление (используя рекурсии, без них), удаление элементов. Построение симметричных деревьев. Обработка списков в виде бинарных деревьев. (8 часов). Л.4, с. 139-142.

Проверка правильности расстановки скобок в арифметическом выражении. Построение обратной польской записи. Потоки. Создание и ведение справочника наличия деталей на складе. Функции хеширования. Создание телефонного справочника. Сложные объекты,

их объявление. Нахождение кратчайшего пути на графе. Сжатие информации. (6 часов). Л.7, с. 211-226; Л.8,184-201.

Модели памяти. Препроцессор. Многоуровневые ссылки. Директивы препроцессора. Условная компиляция. (2 часа).

Понятие объекта и фундаментальные характеристики ООП (инкапсуляция, наследование, полиморфизм). Определение классов (class, struct, union). Элементы данных и методы класса. Защита элементов класса и атрибуты доступа. Объявление и определения методов класса. Вызов членов класса. Конструкторы и деструкторы. (4 часа). Л.5, с. 450-488; Л.6, с. 270-311.

Понятие конструктора и деструктора и их свойства, explicit конструктор. Скрытый указатель this. Массив объектов. Классы, объекты, конструктор и деструктор, this-указатель. (8 часов). Л.5, С. 516-527 2 Л.5, с. 587-595; Л.6, с. 599-625.

Инициализация объектов. Автоматические, динамические и статические объекты. Операторы new и delete. Распределение памяти. Указатель this. Конструкторы и деструкторы. Организация ввода/вывода в С++ с использованием объектов cin и cout. Манипуляторы. Базовые и производные классы. Организация ввода/вывода в С++, манипуляторы, операторы new и delete. (12 часов). Л.5, с. 587-595; Л.6, с. 599-625.

Полиморфизм. Перегрузка функций. Особенности использования перегруженных функций. Перегрузка конструкторов. Перегрузка функций. (4 часа). Л.5, С. 488-495.

Итоговое занятие. Подведение итогов работы в соответствии со сроками реализации программы. Итоговые выставки работ, соревнования, конкурсы и др. Итоговая аттестация учащихся. (2 часа).

ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ

Учащиеся мотивированы, подготовлены и профессионально ориентированы для возможного продолжения учебы в ВУЗах и последующей работы на предприятиях по специальностям, связанным с программированием, созданы условия для работы с разными источниками информации, оценивания их и на этой основе формулирования собственного мнения, суждения, оценки.

В результате реализации программы обучения учащиеся должны знать:

теоретические основы алгоритмизации задач и проектирования программ;

создание программ на алгоритмических языках;

основы организации вычислительных процессов на ЭВМ;

правила безопасной работы в компьютерном классе;

применение С++-проектов в различных сферах жизни и деятельности человека;

назначении основных устройств компьютера;

роли сети Интернет в жизни человека;

алгоритмах, о видах и способах записи алгоритмов;

функциональном устройстве программной среды С++ и основных структурных элементах пользовательского интерфейса;

назначении и использовании основных блоков команд, состояний, программ;

правилах сохранения программного продукта и необходимости присвоения правильного имени.

В результате реализации программы обучения учащиеся должны уметь:

программировать;

отлаживать и выполнять на ЭВМ конкретные задачи с использованием современных методов программирования;

запускать программы и корректно завершать их работу;

создавать простейшие программные продукты;

осуществлять поиск информации в сети Интернет;

работать с библиотеками программной среды;

создавать и редактировать свой проект;

владеть специальными теоретическими знаниями по изучению содержания образовательных областей, тем

уметь публично предъявлять собственные результаты на конференциях, конкурсах, выставках и др.

владеть умениями и навыками технического творчества и исследовательской деятельности.

ФОРМЫ ПОДВЕДЕНИЯ ИТОГОВ РЕАЛИЗАЦИИ ПРОГРАММЫ

Формами подведения итогов реализации программы могут быть: «летопись» объединения по интересам (видео- и фотоматериалы); портфолио творческих достижений объединения по интересам (грамоты, дипломы, сертификаты и др.); отзывы учащихся о выставках, экскурсиях и других мероприятиях, в которых они принимали участие или которые посетили; открытые занятия; отчетные выставки работ учащихся; презентации исследовательских работ; защита проектов; отчетные тематические вечера; заключительные занятия; мастер-классы; составление сборника лучших исследовательских работ учащихся и др.

Для подведения итогов реализации программы может использоваться также оценка результатов, полученных во время публичного предъявления учащимися собственных результатов деятельности на конференциях, конкурсах, выставках и др.

ФОРМЫ И МЕТОДЫ РЕАЛИЗАЦИИ ПРОГРАММЫ

Реализация данной программы требует традиционных и нетрадиционных форм и методов работы, направленных на создание оптимальных условий для достижения ожидаемых результатов в обучении, воспитании, развитии учащихся, удовлетворении их индивидуальных возможностей, потребностей, интересов, раскрытия личностного потенциала каждого.

Для реализации программы используются следующие формы обучения: групповые и индивидуальные.

При реализации данной программы предпочтительной является индивидуально-групповая форма обучения. Педагог дополнительного образования обучает учащихся разновозрастного состава, различного уровня подготовленности, поочередно работая с каждым и предлагая им практические задания. Рекомендуется также парная работа над выполнением конкретного задания.

На практических занятиях с применением компьютерного оборудования, технических устройств, приборов и механизмов, требующих повышенного внимания при их использовании, которые невозможно эксплуатировать при групповой форме обучения, рекомендуется применять индивидуальные формы обучения.

Методы обучения:

объяснительно-иллюстративный метод обучения – метод, при котором обучающиеся получают знания на занятиях во время тренинга, диспута, дискуссии, семинара, консультации, инструктажа, обсуждения; из учебной, технической, справочной литературы через мультимедийные и экранные пособия, интернет;

репродуктивный метод обучения – метод, в котором применение изученного осуществляется на основе образца или правила. Здесь деятельность учащихся носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях;

метод проблемного обучения – это совокупность действий, приемов, направленных на усвоение знаний через активную мыслительную деятельность, содержащую постановку и решение продуктивно-познавательных вопросов и задач, имеющих противоречия (учебные или реальные), способствующих успешной реализации целей учебно-воспитательного процесса;

эвристический метод обучения применяется для организации активного поиска решения выдвинутых в обучении или самостоятельно сформулированных технических и познавательных задач; для стимулирования активного поиска решения поставленных задач используются элементы теории решения изобретательских задач (ТРИЗ), применяются методы: мозговой штурм, проб и ошибок, синектики, морфологический анализ, фокальных объектов, контрольных вопросов, аналогий, объединения, секционирования, модифицирования, копирования прототипов, оптимального проектирования, унификации, агрегатирования, модификации, стандартизации, инверсии, конструирования «КАРУС» и др.;

исследовательский метод обучения – метод, в котором после анализа материала, постановки проблем и задач, краткого инструктажа учащиеся самостоятельно изучают литературу, источники, проводят эксперименты, наблюдения и измерения, моделируют и конструируют, разрабатывают программные продукты, решают прикладные задачи.

Формы воспитания: массовые, групповые, индивидуальные.

Воспитательные мероприятия при реализации программы проводятся в соответствии с Концепцией непрерывного воспитания детей и учащейся молодежи с учетом основных составляющих воспитания детей и учащейся молодежи: идеологическое воспитание; гражданское и патриотическое воспитание; духовно-нравственное воспитание; поликультурное воспитание, экономическое воспитание; воспитание культуры безопасности жизнедеятельности; эстетическое воспитание; воспитание психологической культуры; воспитание культуры здорового образа жизни; экологическое воспитание; семейное и гендерное воспитание; трудовое и профессиональное воспитание; воспитание культуры быта и досуга.

ЛИТЕРАТУРА И ИНФОРМАЦИОННЫЕ РЕСУРСЫ

Основная литература

13. Брябрин В.М. Программное обеспечение персональных ЭВМ. - М.: Наука, 1988.
14. Фигурнов В.Э. IBM PC для пользователя. - М.: Финансы и статистика, 1991.
15. Керниган Б., Ритчи Д., Фьэр А. Язык программирования Си. Задачи по языку Си. - М.: Финансы и статистика, 1985.

16. Керниган Б., Ритчи Д. Язык программирования Си. - М.: Финансы и статистика, 1992.
17. Шилд Г. Программирование на Borland C++ для профессионалов. - Мн.:ООО «Попури», 1998.
18. Сэвитч У. C++ в примерах. -М.: ЭКОМ, 1997.
19. Касаткин А.И, Вальвачев А.Н. От TURBO C к Borland C++. - Мн.: Выш.шк.,1992.
20. Касаткин А.И. Управление ресурсами, - Мн.: Выш. шк., 1992.
21. Касаткин А.И. Системное, программирование. - Мн.: Выш.шк., 1993.
22. Романовская Л.М., Русс Т.В., Свитковский С.Г., Программирование в среде Си для ПЭВМ. - М.: Финансы и статистика, 1992.
23. Бочков С.О., Субботин Д.М. Язык программирования Си для персональных компьютеров. - М.: Радио и связь, 1990.
24. Луцик Ю.А., Ковальчук А.М. и др. Методическое пособие по курсу «Основы алгоритмизации и программирования» -Мн.: БГУИР, 1999.

Дополнительная литература

8. Демидович Е.М. и др. Практикум по курсу «Программирование» для студентов специальности «Вычислительные машины, системы и сети» . - Мн. : БГУИР,1995.
9. Демидович Е.М. Учебное пособие по курсу «Основы алгоритмизации и программирования» для студентов специальности «Вычислительные машины, системы и сети». - Мн.: БГУИР, 1999.
10. Демидович Е. М. и др. Методическое пособие по курсу «Основы алгоритмизации и программирования» для студентов специальности

- «Вычислительные машины, системы и сети». -Мн.: БГУИР, 1999. 4.4.
- Глухова Л.А. и др. Методические указания по вычислительной практике и самостоятельной работе по дисциплине «Программирование» и «Конструирование программ и языки программирования» для студентов специальности «Вычислительные машины, комплексы, системы и сети» «Программное обеспечение ЭВМ и автоматизированных систем» и слушателей спецфакультета переподготовки по направлению «Микропроцессорные системы», часть 1. -Мн.: МРТИ, 1989, 33с.
11. Фадеева Е.П. и др. Методические указания по вычислительной практике и
12. самостоятельной работе по курсам «Программирование» и «Конструирование программ и языки программирования» для студентов специальности «Вычислительные машины, комплексы, системы и сети», «Программное обеспечение ЭВМ и автоматизированных систем» и слушателей спецфакультета переподготовки по направлению «Микропроцессорные системы», часть 2. - Мн.: МРТИ, 1990.
13. Кнут Д.Э. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. -720 с.: ил.
14. Кнут Д.Э. Искусство программирования, том 2. Получисленные алгоритмы, 3-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. - 832 с.: ил.
15. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд.: Пер. с англ.: Уч. Пос. - М.: Издательский дом «Вильямс», 2000. - 832 с.: ил.