

**Государственное бюджетное учреждение дополнительного образования
Центр детского (юношеского) технического творчества
Кировского района Санкт – Петербурга
198095, Санкт-Петербург, ул. Маршала Говорова, д. 34, литер 3**

**Принята на заседании
педагогического совета
от « 31 » 08 2020 г.
Протокол № 1**

**УТВЕРЖДЕНА
Приказом № 40-ОД от « 31 » 08 2020 г.
Директор ГБУ ДО ЦДЮТТ
Ясинская Е.С**

**Дополнительная общеобразовательная общеразвивающая программа
«ВВЕДЕНИЕ В ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ»**

Возраст учащихся: 11-14 лет
Срок реализации: 1 год

Разработчик:
*Егорова А. Н.,
педагог дополнительного образования*

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В наши дни программирование является одной из наиболее актуальных сфер человеческой деятельности. На рынке труда программисты являются крайне востребованными специалистами. Профессия значительно помолодела, и сейчас на выходе из школы необходимо владеть несколькими языками программирования на уровне создания базовых конструкций для успешного совершенствования в данной отрасли.

Современное программное обеспечение меняется и совершенствуется на глазах, поэтому недостаточно знать одну версию языка, базирующуюся на конкретной операционной системе. Кроме того, школьный курс программирования не предоставляет учащимся актуального уровня знаний и навыков в этой области.

Вместе с этим существует ряд смежных и взаимодействующих с программированием специальностей. Программное обеспечение внедрено во многие сферы деятельности, уже в подавляющее число отраслей экономики, производства, коммуникаций. Для максимально эффективного использования приложений и технических систем желательно более глубокое знакомство с предметом программирования, чем просто запоминание способов использования предоставляемых готовых интерфейсов. Также в инженерной, студенческой среде перестала быть редкостью разработка несложных программных продуктов непрофессионалами. Специалистам прикладных областей в некоторых случаях бывает проще написать программный код, чем составлять техническое задание, искать специалиста или подрядчика, нести затраты.

Необходимость разработки предоставляемой дополнительной общеобразовательной общеразвивающей программы *«Введение в процедурное программирование»* обусловлена современной потребностью сопровождения технического курса изучения программирования элементами гибких компетенций (soft skills), обзорами направлений применения технических знаний и навыков. Программа служит инструментом самоориентации учащегося перед изучением классического курса программирования, устранения неверно сформированных через СМИ представлений о лёгкости работы программиста или нацеленности на визуальные эффекты. Программа сглаживает возможные предварительные пробелы в общей компьютерной грамотности учащихся, связанные с различным качеством персонального рабочего места, опытом работы, облегчая «срабатывание» коллектива при начале изучения классического курса программирования.

Программа разработана согласно требованиям следующих документов:

- Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 №273-ФЗ.
- Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 № 1726-р).
- СанПин к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41)
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (утвержден приказом Министерства просвещения РФ от 09.11.2018 № 196).
- Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена Распоряжением Правительства Российской Федерации от 29 мая 2015 г. N 996- р).
- Методические рекомендации по проектированию дополнительных общеразвивающих программ в государственных образовательных организациях Санкт-

Петербурга, находящихся в ведении Комитета по образованию (утверждено распоряжением Комитета по образованию от 01.03.2017 №617-р).

Направленность программы – техническая.

Уровень освоения программы – базовый.

Актуальность программы

Программа составлена в соответствии с запросом учащихся на развитие их компетентности в области программирования, а также на предварительную профориентацию в области информационных технологий.

Программа должна служить популяризации работы в отрасли информационных технологий или связях с нею вместе с формированием более осознанного правильного отношения к ценности труда разработчика программных продуктов. Программа должна показать возможности применения компьютера помимо создания и использования игр и сайтов, необходимость которых при правильном подходе не отрицается.

Программа представляет собой курс программирования на классическом процедурном языке, актуальном в учреждении на момент применения, с сокращённой теоретической частью по программированию за счёт усиления обзорных ориентационных занятий, элементов информатики и общей компьютерной грамотности, творческих практических занятий.

Программа в полной мере соответствует положениям Концепции развития дополнительного образования, т.к. направлена на «создание необходимых условий для личностного развития учащихся, позитивной социализации и профессионального самоопределения», «удовлетворение индивидуальных потребностей учащихся в интеллектуальном развитии», имеет «тесную связь с практикой, ориентацию на создание конкретного персонального продукта и его презентацию».

Та же Концепция провозглашает «принцип содействия государственно-частному партнерству в сфере игровой индустрии, производящей безопасные игры (в том числе компьютерные игры общеразвивающего и обучающего характера), игрушки, имитационные модели, способствующие расширению условий реализации дополнительных общеобразовательных программ, психолого-педагогическому проектированию образовательных сред, стимулированию детей к познанию, творчеству и конструктивной деятельности». Данная образовательная программа призвана мотивировать учащихся не бездумно потреблять игровой продукт, а конструировать свои проекты, тематика которых может регулироваться педагогом. Проводимые занятия должны иметь своей целью, в том числе, и стимулирование учащихся к конструктивной творческой деятельности по тематике объединения.

Отличительные особенности программы

Решение о создании программы, не содержащей указания на конкретный язык программирования, принято в связи с тем, что многие курсы обучения программированию для начинающих имеют в основе общие фундаментальные понятия информатики, неизменные уже в течение десятилетий (байт, массив данных, пиксельные элементы экрана, ввод с клавиатуры и других устройств), при том, что конкретные инструменты разработчика непрерывно перерабатываются, изменяются наименования команд (например, команды вывода текста на экран в разных языках и версиях библиотек именуются `print`, `write`, `cout` и т. д., оформляются разными видами скобок и дополнительных параметров, но в целом выполняют сходные действия). В связи с распространением дистанционных форм образования стали появляться случаи, когда учащийся успешно использует дома иные версии сред разработки и даже версии языка, чем в классе на занятиях - при этом формулировка темы занятия без привязки к языку облегчает педагогу соблюдение тематического плана.

Содержание программы является самостоятельным блоком материала по фундаментальным теоретическим основам программирования и алгоритмизации. Одной из главных задач является формирование понимания учащимися целей программирования

устройств, выделение алгоритма как объекта действия, выделение устройства как объекта со специфическими свойствами, и собственной роли при манипуляциях с устройством. Отличительная особенность программы состоит в сочетании академического подхода к изучению процедурного программирования и современных технологий, как информационных, так и педагогических. Подача материала дублируется, по возможности, разными доступными способами: лекция, практика, косвенное восприятие через многократное повторение в разных формах с переплетением этих форм (массив как абстрактная структура - массив пикселей экрана, линейный массив в памяти и т. д.).

Программа может быть противопоставлена курсам профессиональной и объектно-ориентированной разработки программного обеспечения, так как ставит освоение конкретного технического продукта или технологии второстепенной задачей, отдавая приоритет пониманию принципов подхода к технике и психологической подготовке к восприятию больших объёмов информации на классических курсах программирования. Важно, что работа должна проводиться в зоне ближайшего развития целевой группы учащихся, обращаясь к образному правополушарному восприятию, преобладающему в младшем возрасте.

Для обучения по программе целесообразно использовать доступные среды разработки с возможностью создания полноценных текстовых и графических оконных приложений. Среда разработки должна быть не адаптированной специально для обучения и обладать функционалом отладки приложений. Наличие русскоязычного интерфейса не обязательно. Среди прочих может быть выбран инструмент с довольно лаконичным интерфейсом и доступный для установки учащимися самостоятельно в домашних условиях.

Важной составляющей программы является наличие индивидуального проекта, который может быть продемонстрирован педагогу и родителям, всей группе в рамках занятия, при наличии конкурентных достоинств - на внешнем публичном мероприятии. Самостоятельный выбор темы или элементов проекта развивает гибкие навыки, способствует рефлексии и профессиональному общению. Демонстрация проекта внутри группы, внутри учреждения с присутствием нескольких групп, вне учреждения или на базе учреждения с присутствием приглашённой аудитории, наличие или отсутствие формальных критериев оценки способствует тренировке навыков самопрезентации, академического общения. Получение публичной обратной связи помогает самоопределению, корректировке творческих целей, выявлению склонностей к исследовательской работе. Приглашение к творческой деятельности требует учёта индивидуальных психологических особенностей учащихся, возможной вариативности в тематике или готовности мягко обосновать недостижимость предлагаемых детьми целей в рамках учебного года.

Адресат программы

Программа рассчитана на учащихся 11-14 лет, заинтересованных в обучении программированию, не имеющих подтверждённого опыта в этой дисциплине или чёткого понимания целесообразности изучения классического программирования. Программа может заинтересовать учащихся с опытом программирования, ищущих возможностей повторения и закрепления материала.

Пол значения не имеет. Медицинские противопоказания отсутствуют.

Цель образовательной программы:

Повышение уровня информационной культуры, развитие абстрактного и логического мышления учащихся через приобщение их к созданию проектов по программированию посредством теоретических и практических занятий.

Задачи программы

Обучающие:

- дать обзорные представления об отрасли информационных технологий, её участниках;

- дать первоначальное теоретическое представление о компьютерной технике;
- формировать восприятие программируемого устройства как статического объекта деятельности программиста и инженера;
- формировать понимание минимальной схемы вычислительных устройств на основе архитектуры фон Неймана: процессор и линейная память;
- дать первоначальное теоретическое представление об операционной системе;
- сформировать представления о целях и способах создания программного обеспечения;
- сформировать или актуализировать минимально необходимые пользовательские навыки;
- ознакомить с понятием алгоритмов, их видами и реализациями;
- обучить читать структурные схемы алгоритмов;
- ознакомить с операторами и стандартными функциями процедурного языка;
- обучить базовым конструкциям в программировании: циклы, ветвления, массивы и их сортировка;
- обучить созданию графического оконного приложения на самостоятельно выбранную прикладную тему;
- ознакомить с возможностями проектирования пользовательского интерфейса;
- обучить принципам построения программного кода, приучить к комментированию и разметке кода;
 - научить создавать программы для имеющейся операционной системы;
 - создать условия для дальнейшего совершенствования навыков учащихся в области программирования.

Воспитательные:

- развивать способность и стремление к самообразованию;
- создавать условия для профессионального самоопределения учащихся по направлению деятельности объединения;
- воспитывать способность и готовность к общению и сотрудничеству со сверстниками и взрослыми в процессе учебной, творческой деятельности;
- воспитывать навыки академического общения;
- воспитывать стремление к соблюдению принципов академической честности при работе с информацией;
- формировать осознанное правильное отношение к ценности труда разработчика программных продуктов;
- воспитывать убежденность в преимуществах общечеловеческих ценностей через участие в учебно-воспитательных мероприятиях ЦДЮТТ, района, города, в процессе бесед в рамках объединения.

Развивающие:

- развивать память, внимание, техническое, абстрактное, логическое и образное мышление, эстетический вкус учащихся;
- формировать умение ставить и формулировать проблемы;
- развивать навыки самоорганизации собственной познавательной деятельности;
- формировать умение применять методы информационного поиска;
- развивать навыки структурирования информации;
- повышать уровень информационной грамотности учащихся;
- формировать творческий подход к решению задач, творческую самостоятельность;
- развивать способность к рефлексии;
- расширять кругозор учащихся;
- формировать умение представлять результаты своего труда публично (в форме защиты).

Объём и сроки реализации программы

Программа рассчитана на 1 год обучения, реализуется в объёме 72 часа.

Условия реализации программы

Прием в коллектив осуществляется на основании заявления родителей. Принимаются все желающие; не предъявляется никаких специальных требований к знаниям, уровню подготовки, способностям подростков.

Состав группы разновозрастной – 11-14 лет.

Наполняемость группы – 11 человек.

(Наполняемость определяется количеством рабочих посадочных мест за компьютерами в компьютерном классе).

Программа может ежегодно корректироваться в зависимости от нагрузки педагога (на основании локального акта Учреждения) и особенностей набранного контингента учащихся.

Форма проведения занятий

Основные формы проведения занятий - комбинированное, практическое компьютерное занятие. Проводятся занятия-исследования. В конце года используется такая форма проведения занятий как защита творческих работ.

Выбор данных форм проведения занятий обусловлен предметной областью, к которой относится программа, а также при выборе учитывалась необходимость поддерживать постоянный интерес учащихся на всем протяжении обучения.

Форма организации деятельности учащихся на занятии

- фронтальная (беседа, демонстрация приёмов программирования и компьютерной работы);
- индивидуальная в рамках фронтальной (работа по образцу, выполнение практических и творческих заданий);
- групповая (обсуждение проектов, выработка оригинальных алгоритмических решений).

Материально-техническое оснащение

Для успешной реализации представляемой дополнительной общеобразовательной общеразвивающей программы необходим отдельный компьютерный класс с количеством компьютеров по количеству учащихся + отдельное рабочее место педагога. Помещение должно соответствовать санитарно-гигиеническим нормам и технике безопасности.

На компьютерах должна использоваться операционная система ОС Windows или Linux. В качестве среды разработки могут быть использованы такие IDE как Microsoft Visual Studio, Qt, Lazarus, линейка JetBrains - по текущим дидактическим возможностям педагога или коллектива, желательно доступные для бесплатной самостоятельной установки учащимися дома. Среда разработки должна позволять создавать графическое оконное приложение и иметь базовый набор рисования графических фигур на двумерной «канве» или непосредственно в графическом окне.

Примеры подборок программных инструментов:

Microsoft Visual Studio, язык си и другие, графика WinApi или SFML или OpenGL;

Microsoft Visual Studio, язык си/C#, библиотека Windows Forms;

Среда разработки Qt (можно под Linux/Ubuntu), графика OpenGL;

Среда разработки Lazarus, язык Pascal, графика SFML или OpenGL.

Среда разработки Borland Pascal или C при поддержке симулятором графики.

Кадровое обеспечение программы

Программа реализуется педагогом дополнительного образования, имеющим достаточный уровень квалификации и профильное образование.

Планируемые результаты освоения программы

Предметные:

- сформированность представления об отрасли информационных технологий, её участниках;

- сформированность представления об операционной системе;
- сформированность представлений о целях и способах создания программного обеспечения;
- начальное понимание аппаратной схемы компьютера или другого программируемого устройства;
- сформированность восприятия программируемого устройства как статического объекта деятельности программиста и инженера;
- знание базовых алгоритмических конструкций;
- умение читать структурные схемы алгоритмов;
- знание основ и синтаксиса процедурного языка;
- сформированность представления о базовых конструкциях в программировании: циклы, ветвления, массивы и их сортировка;
- сформированность необходимых пользовательских навыков;
- умение создавать масштабируемый код, сопровождаемый достаточным количеством комментариев;
- умение создавать графическое оконное приложение на самостоятельно выбранную прикладную тему;
- сформированность представления о возможностях проектирования пользовательского интерфейса;
- умение создавать программы для имеющейся операционной системы

Метапредметные:

- умение ставить и формулировать проблемы;
- умение применять методы информационного поиска;
- умение структурировать и визуализировать информацию;
- освоение способов решения проблем творческого и поискового характера;
- получение опыта планирования и организации собственной познавательной деятельности на основе сформированных регулятивных учебных действий;
- сформированность технического, образного, абстрактно-логического мышления, эстетического вкуса учащихся;
- сформированность умения представлять результаты своего труда публично (в форме защиты);
- расширение кругозора.

Личностные:

- готовность к соблюдению принципов академической честности при работе с информацией;
- приобретение навыков сотрудничества, содержательного и бесконфликтного участия в совместной учебной работе;
- формирование навыков академического общения;
- формирование навыков рефлексии;
- понимание роли информационных процессов в современном мире;
- готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ;
- предварительная профориентация в области информационных технологий;
- сформированность осознанного правильного отношения к ценности труда разработчика программных продуктов;
- ориентация на общечеловеческие ценности.

Учебный план

Тема занятий	Часы			Формы контроля
	Теория	Практика	Всего	
Введение	1	1	2	Опрос, практическая работа
Тема 1. Компьютер и пользователь	2	2	4	Практическая работа. Коллективное обсуждение
Тема 2. Компьютер и разработчик	2	2	4	Коллективное обсуждение. Подготовка доклада или презентации
Тема 3. Основы языка программирования: данные, условия, циклы	4	4	8	Анализ выполненных работ, опрос
Тема 4. Графический проект	4	4	8	Анализ выполненных работ, коллективное обсуждение
Тема 5. Массивы и их применение	4	4	8	Анализ выполненных работ, опрос
Тема 6. Структуры, файлы, алгоритмы	8	10	18	Анализ выполненных работ, опрос
Выполнение итоговых творческих работ	2	8	10	Анализ выполненных работ, коллективное обсуждение
Итоговое занятие	0	2	2	Смотр творческих работ
Учебно-массовые мероприятия	0	8	8	Конкурс, конференция, коллективное обсуждение
Всего часов	27	45	72	

Календарный учебный график

Год обучения	Дата начала обучения	Дата окончания обучения	Всего учебных недель	Кол-во учебных часов	Режим занятий
1 год	14.09.	31.08	36	72	2 раза в неделю по 1 часу

Рабочая программа

Содержание программы I года обучения

Введение

Цели и задачи образовательной программы, объединения, режим посещения.

Рекомендуемая литература, интернет-источники.

Охрана труда при работе с компьютером.

Практическая работа: Проведение практической работы по охране труда при работе с компьютером.

Выполнение пробного проекта по инструкции: минимальная программа в основной среде программирования. Наблюдение, индивидуальная и групповая беседа. Предварительная оценка знаний и навыков учащихся - восприятие печатного материала, скорость набора текста на клавиатуре и выполнения действий в операционной системе, наличие предметных знаний, мотивации к самостоятельному поиску, наличие предметного кругозора и источники его пополнения.

Тема 1. Компьютер и пользователь

Современное состояние отрасли информационных технологий в мире и в РФ. Экономическое значение. Мир без информационных технологий (фантазия). Виды производств с использованием цифровых технологий: промышленное производство (цифровизация цехов, встроенные бортовые компьютеры, космос и добыча полезных ископаемых), системы "B2B", потребительский рынок (бытовая техника с электроникой, "умный дом"), государственные услуги онлайн. Развитие отрасли в последние десятилетия, даты изобретений XX века и ранее - физика, электротехника, первые компьютеры, эволюция элементной базы.

Понятие программного обеспечения как экономического продукта.

Программируемые устройства: от дешёвого контроллера до мощного компьютера.

Примеры, рекорды производительности.

Виды профессиональной деятельности при производстве программного обеспечения: программист, проектировщик, разработчик интерфейса, понятия front-end и back-end.

Изучение конкретного рабочего компьютера, за которым планируется дальнейшего практическая работа учащегося. Обзор всех (желательно) ярлыков, максимального количества видимых учащемуся программ (пояснение без демонстрации). Правила пользования совместным компьютером, политика запретов и разрешения на перемещение ярлыков, смену обоев рабочего стола и т. п. Личная папка учащегося. Сравнение с другими рабочими местами, которыми учащиеся пользовались ранее.

Английский как язык межнационального профессионального общения в цифровой отрасли и академической научной среде. Факт наличия публикаций и книг, издаваемых на языке авторов.

Переводы и правильное написание часто употребляемых слов и понятий (error, keyboard, language, script, style...), надписей на клавишах и системном блоке (Enter, Escape, Reset, Space...). Происхождение сокращений или аббревиатур (OS - operating system, FAQ - frequently asked questions, Ru - Russia, etc), имён доменов (как com, org)

Способы ввода: клавиатура, мышь, джойстик, "тачпад", микрофон, любой внешний датчик. Группировка клавиш. Различия клавиатур - отдельных, встроенных в ноутбук, интересные фото.

Кодировки (1251, UTF, Unicode). Таблица ASCII как изначальный стандарт. Кодирование некоторых неотображаемых символов (Enter как 10 + 13, обозначение \n). События, которые могут приходить от мыши.

Практическая работа:

Выделение вокруг себя продуктов, полностью цифровых, содержащих элементы информационных технологий, разработанных при помощи информационных технологий.

Выделение информационных цепочек и алгоритмов, заметных в повседневной жизни (софт оператора - сетевая передача - информационные текстовые табло на КАД/ЗСД; касса магазина - база данных товара - связь с банком; сообщения об ошибке при сбоях).

Путешествие по своему рабочему месту. Блокнот, офис, проводник и пользование им, флеш-карта, браузер, графический редактор, калькулятор, командная строка, справка, переключение регистра и языка клавиатуры, настройки рабочего стола. Обсуждение программ, которые учащиеся знают, но не нашли на своём конкретном рабочем месте.

Набор текста в английской раскладке. Знакомство с кратким словариком.

Поиск терминов в первом пробном проекте и в среде разработки

Изучение всех клавиш на клавиатуре - назначение, необходимость в данный момент. Ввод текста в разных кодировках с помощью стандартной офисной программы и основной среды разработки (как видоизменение пробного проекта).

Тема 2. Компьютер и разработчик

Минимальная схема компьютера (процессор - шина данных - ОЗУ и контроллеры основных устройств).

Параметры процессора, ОЗУ - мировая история, реальные примеры, современные рекорды производительности, габаритов, стоимости.

Потребность в многопроцессорности. Шейдерная графика как способ оптимизации.

Минимальная схема контроллера (на примере Arduino или подобного популярного), требующего программирования через подачу сигнала на конкретные выходы.

Операционная система как программа управления аппаратурой.

История развития, наименования, даты.

Минимальная схема ОС (ядро + сервисы и расширения). Варианты архитектур. ОС Windows - причины критики.

Двоичный элемент как основа всей компьютерной логики. Троичный элемент как альтернатива. Минимальные основы физики (электроток, напряжение). Математические системы счисления.

Байт как единица информации. Причина избрания величины байта. Наглядность выполнения операций со степенью двойки на схеме байта (битовые сдвиги). Максимальные и минимальные числа, которые можно разместить в байте, двух, четырёх, восьми байтах.

Проблема порядка размещения в памяти многобайтовых чисел (старший слева или старший справа)

Виды ОЗУ: процессорная память, кэш, съёмная плата, соотношение быстродействия и стоимости изготовления.

Разметка диска. FAT как наглядный способ представления.

Битность операционной системы, адресация (с ростом битности растёт возможность адресовать больший объём, но снижается компактность хранения малых чисел).

Практическая работа:

Беседа об известных устройствах, операционных системах, внедрении электроники в производство и изделия повседневного пользования.

Перевод между системами счисления.

Подготовка доклада или презентации на свободную тему, связанную с деятельностью отрасли информационных технологий, производств, программистов.

Тема 3. Основы языка программирования: данные, условия, циклы

Переменная как область памяти для хранения. Простые типы данных.

Ввод и вывод числовой переменной. Вывод текстовых запросов при вводе и наименования величины при выводе данных.

Арифметические действия: сложение, вычитание, умножение, деление, нахождение остатка. Разница между целочисленным и вещественным делением.

Переполнение данных.

Примеры программ для решения арифметических задач (заголовок программы, запрос данных, ввод данных, выполнение действий, вывод результата и пояснения к результату).

Выбор типа данных в зависимости от целей задачи (диапазон, точность, вероятность появления вещественных чисел после деления).

Условный оператор с одним условием. Встраивание одного условия в арифметические задачи. Составной оператор, блок кода. Проверка валидности входных данных

Формальная схема алгоритма, все основные блоки. Алгоритм с условием, с циклом, с подпрограммой. Простая подпрограмма для оформления одинаковых участков кода.

Основные логические функции (отрицание, конъюнкция, дизъюнкция, импликация, таблица вариантов результата, битовая схема в случае применения к целочисленному типу). Составление сложных условий, учёт скобок.

Вложенные блоки условий. Вложенные операторы.

Цикл с целочисленным счётчиком. Граничные условия, строгие и нестрогие проверки равенства в условии. Возрастающий и нисходящий счётчик. Инкремент и декремент с префиксной и постфиксной записью (при наличии в языке программирования). Потенциально опасное действие - изменение итератора внутри тела цикла.

Числовые ряды, диапазоны (чётные трёхзначные числа, квадраты чисел заданного диапазона в нисходящем порядке, числа Фибоначчи и пр.).

Числовые таблицы в текстовом режиме. Итераторы вложенных циклов.

Циклы без счётчика. Возможность полной замены цикла со счётчиком на цикл с предусловием.

Ввод и проверка нажатого символа. Таблица ASCII.

Практическая работа:

Арифметические задачи - реализация программ. Составление собственных условий задач, реализация программ (число параметров по выбору учащихся, вычисления в одно или несколько действий).

Составление блок-схемы алгоритма. Написание программы по блок-схеме с условием.

Составление и решение арифметических задач с составными и вложенными условиями.
Проверка валидности входных данных.

Вывод числовых рядов в текстовом режиме. Определение алгоритма по готовому ряду.
Вывод числовых таблиц (таблица умножения и т. д.).

Использование цикла без счётчика.

Ввод и проверка нажатого символа.

Программа "угадай число" - продолжение попыток до появления правильного ответа.

Защипливание кода решения задач (запросы "Продолжить? Y/N", "Для выхода нажмите '1' " и т.п.).

Тема 4. Графический проект

Физика цвета RGB (оптика трёх лучей). История многоцветного аппаратного вывода, битность на пиксел (двухцветный с яркостью, 16-цветный, 256-цветный, 16 bit, 24bit). История техники, типы мониторов (ЭЛТ, ЖК).

Окно в терминах операционной системы. Поток. Приложение без окна.

Стандартные части окна (заголовок и кнопки управления, область вывода).

Минимальное окно. Окно нестандартной формы (круглое и т.п.). Полноэкранный вывод (fullscreen).

Возможность и способ хранения массива пикселей в линейной памяти: адрес = $y * \text{ширина} + x$ (независимо от того, используется ли эта модель в графической системе применяемой среды разработки).

Вложенный цикл в графике. Рисование множества фигур - одинаковых или с постепенно изменяющимися параметрами.

График функции - математическая беседа. Переменная и зависимость. Двумерное отображение, оси координат. Простейшие функции.

Изменение параметров содержимого графического окна при нажатии клавиши или кнопки мыши. Общепринятые клавиши управления (стрелки, asdw), особенности считывания клавиш (регистр, языковая раскладка, двухбайтовые коды). Считывание функциональных клавиш при наличии возможности.

Время в компьютере. Форматы хранения времени. Таймер, доступный приложению.

Алгоритмическое тестирование (отображение на фигурах программных орнаментов).

Загрузка внешних файлов изображений при возможности.

Оформление окна приложения. Цветовая гамма.

Практическая работа:

Работа с шаблоном оконного проекта. Знакомство с двумерной графикой, минимальными фигурами (точка, отрезок, окружность). Окрашивание.

Работа с рядом минимальных фигур. Работа с таблицей минимальных фигур. Создание собственного орнамента.

Построение графика функции. Оформление осей координат - масштаб, подписи при возможности.

Управление простым объектом с клавиатуры на двумерном экранном поле.

Динамическое изменение графика функции при нажатии клавиши.

Вставка текстуры или программное окрашивание. Оценка ресурсоёмкости. Обсуждение удачных цветовых и дизайнерских решений приложений из личного опыта учащихся.

Изображение занимательных кривых. Спираль круглая и квадратная. Снежинки как функции вида $x(a) = b1*\sin(a) + b2*\cos(a*b3)$; $y(a) = b4*\cos(a) + b5*\sin(a*b6)$;
Новогодняя открытка из статических фигур (снеговик, ель и т. д.)

Тема 5. Массивы и их применение

Одномерный (линейный) числовой массив. Ввод и вывод в текстовом режиме.

Сумма элементов. Сумма элементов по критерию.

Поиск значения в массиве (неотсортированном).

Разворот массива in-place (с одной вспомогательной переменной). Разворот с записью во второй массив.

Слияние, разбиение массивов.

Комбинированные манипуляции (слияние совместно с разворотом или вычислением накопительной суммы, разбиение по условию и пр.)

Движение группы объектов. "Падающий снег" на массиве координат: массив высот каждой точки (пробегаем каждый пиксел по ширине), массив пар координат (x, y) - два отдельных массива или массив структур. "Переход" снежинок снизу вверх - достигнув земли, появляется в верхней точке экрана.

"Ветер": смещение горизонтальных координат снежинок на фиксированную или случайную величину. "Переход" снежинок через левый или правый край экрана.

Накопление падающего снега (дополнительный массив для хранения максимальной высоты снежного столбика для каждого горизонтального пиксела).

Сортировка массива "пузырьком", "обменом", по возрастанию или убыванию. Действия в случае равенства элементов. "Быстрая сортировка" (тема повышенной сложности).

Двумерные массивы. Ввод и вывод в текстовом режиме, оформление приглашения ко вводу (формата "x[i][j] = " с подстановкой текущих индексов в строку запроса).

Игровое поле на основе двумерного массива. Варианты игр, подходящие по сложности для реализации на первом году обучения (лабиринт, змейка, сапёр и др.).

Практическая работа

Реализация упражнений с одномерными массивами в текстовом режиме.

Реализация графической программы "Новогодняя открытка" (или на другую тему, по согласованию с педагогом для определения уровня сложности).

Реализация сортировки массива выбранным способом или несколькими способами.

Реализация упражнений с двумерными массивами в текстовом режиме.

Начало реализации игровой программы на двумерном поле (как один из наглядных вариантов для начала разработки).

Тема 6. Структуры, файлы, алгоритмы

Алгоритмы игр для двумерного поля - массива либо координатного пространства без массива.

Свободное перемещение объекта по координатному пространству (шаг вперёд/назад и поворот влево/вправо) - тригонометрическая формула. Отражение объектов от стен.

Способы реализации "бота" в лабиринте и координатном пространстве: движущийся "туда-обратно" по прямой в заданном диапазоне клеток, движущийся по прописанной в статическом массиве траектории, делающий шаг в сторону персонажа без обхода препятствия, делающий шаг в сторону персонажа с попыткой обхода препятствия в случайном направлении или по правилу левой/правой руки.

Тестирование последовательности псевдослучайных чисел на повтор. Генерация псевдослучайного числа.

Алгоритм точного поиска минимального пути в лабиринте через расстановку чисел (начиная от объекта назначения, ставим во все свободные соседние клетки число, на единицу большее предыдущего).

Персональные таймеры для нескольких перемещающихся объектов. Анализ производительности при перемещении нескольких объектов.

Соприкосновение двух кругов/окружностей - вычисление расстояния между центрами. При невозможности определения цвета пиксела экрана, подходы к определению касания персонажей через их упрощенную геометрическую модель.

Полёт тела по параболе (баллистика) - основы для игр типа Scorch, Mario. Учёт текущей координаты и скорости перемещения (подбор детализации отрисовки при расчёте параболы, так как размер отрезков-скачков может оказаться разным). Изображение катящегося колеса.

Комбинаторика - математическая теория (сочетание, перебор). Алгоритм перебора. Код Грея.

Использование массива массивов (например, хранение нескольких двумерных игровых уровней). Правильное обращение с индексами массивов (горизонталь, вертикаль, номер массива - не путаем), представление многомерного массива в линейной памяти.

Преимущества и сложности динамического использования памяти. Необходимость освобождения памяти вручную, кроме случаев автоматической очистки в конце работы приложения.

Абстрактные динамические структуры (список, дерево, хэш-таблица, граф), их представление в линейной памяти. Организация списка вручную через указатель (тема повышенной сложности). Обзор основных алгоритмов (односторонний, двусторонний списки, обход дерева, сортировка с помощью дерева, поиск пути в графе, веса в графе, словарь на основе хэш-таблицы).

Файл и файловая система (частично повторение пройденного). Анализ собственной файловой системы. Структура диска. Атрибуты файлов.

Запись в файл, чтение из файла минимальной единицы информации.

Текстовый и бинарный форматы файла - отличия, выбор между этими вариантами в зависимости от цели создания и назначения файла.

Запись в файл многострочной или структурированной информации.

Ведение таблицы рекордов на основе массива структур. Отсутствие необходимости сортировки при добавлении новой записи в отсортированную таблицу (использование поиска нужной позиции и сдвига остальных элементов).

Сортировка таблицы рекордов по критерию.

Обзор максимального числа известных форматов файла. Кодирование типа файла в первых его байтах. Условность расширения файла (при замене расширения вручную реальный тип файла не меняется, и он не будет прочтён другими приложениями). Примеры специализированных, но текстовых форматов без сжатия (dxf, vscproj и т. д.). Открытые и закрытые форматы.

Простые схемы сжатия информации, применимость алгоритмов к разным исходным данным (значение байта и количество аналогичных ему, поиск повторяющихся последовательностей и замена их, упаковка в биты LZW). Увеличение размера информации после сжатия при неподходящем алгоритме.

Сжатие с потерей и без потери. Замена градиента на граничные значения и количество шагов.

Шифрование. Простой сдвиг кода каждого символа или применение к каждому символу логической функции. Сдвиг на разное количество единиц (по известной разработчику функции). Шифрование с открытым и закрытым ключом.

Профессиональная база данных - обзор. Postgres, SQL. Обязательность уникального ключа. Пример запроса (независимо от наличия установленного программного обеспечения).

База данных товаров по артикулу, пользовательская корзина товаров.

Очередь, стек, дек как абстрактные структуры. Их реализация на массиве или на списке.

Практическая работа:

Реализация одного или нескольких выбранных алгоритмов для графических игр, или другой проект аналогичной сложности по согласованию с педагогом.

Реализация отдельных алгоритмических упражнений.

Работа с файлами. Чтение, запись минимальной единицы информации. Встраивание работы с файлами в начатый ранее игровой или иной проект (сохранение данных, загрузка прежней позиции при старте).

Реализация таблицы рекордов отдельно в текстовом режиме.

Встраивание таблицы рекордов в графический проект (или отдельная реализация в графическом режиме).

Реализация упражнений по форматам файлов (например, чтение/запись bmp по описанию заголовка).

Визуализация структур: столбиковая диаграмма по значениям введенного массива, пошаговое отображение сортировки, ручная реализация структур с указателями (повышенная сложность).

Реализация алгоритмических упражнений по согласованию с педагогом или разработка индивидуального проекта для подготовки к конференции (не исключая прослушивание теоретического материала).

Выполнение итоговых творческих работ

Обсуждение желаемых вариантов, реализуемых вариантов. Анализ проектов, выполняемых ранее в течение года.

Реализация главного меню: добавление меню в основной алгоритм (при наличии) или встраивание основного алгоритма во вновь созданный проект с главным меню.

Проектирование работы с файлами, необходимой для проекта. Реализация управления профилями, аккаунтами, файлами сохранения или другими необходимыми данными.

Реализация или доработка основного алгоритма проекта.

Практическая работа:

Уточнение, доработка основного алгоритма проекта. Коррекция или расширение возможностей (добавление уровней, алгоритмов). Дизайн основной части (цветовая гамма, внешний вид персонажей, надписи, орфография). Проверка и уточнение интерфейсов управления (мышь, клавиатура, графические кнопки управления). Внесение полной информации в раздел справки, создание или проверка статистики использования персональной или общей.

Общее тестирование, поиск уязвимостей.

Составление, проработка, чтение текстовых докладов.

Итоговое занятие

Практическая работа:

Смотр творческих работ

Персональные консультации, рекомендации по продолжению обучения по итогам года и с учётом уровня творческой работы.

Учебно-массовые мероприятия

Учебно-массовые мероприятия в рамках объединения, ЦДЮТТ, района и города: конкурсы внутри объединения, районный конкурс компьютерных работ, городской

конкурс школьников по программированию и компьютерным работам (номинация «Программирование»), при наличии выдающихся работ - всероссийские конкурсы («Будущее сильной России в высоких технологиях», «Балтийский научно-инженерный конкурс», «Школьная информатика и проблемы устойчивого развития», «Юный техник») и другие.

Общая беседа о профессии программиста (презентации).

Фестиваль новогодних компьютерных работ.

Новостная лента технологий (обсуждение отраслевой информации из интернета).

Весенний проект - открытка, игра, краткое поздравление в выбранной форме.

Городская конференция по решению задач.

Городская конференция по презентации проектов.

К Дню Победы (в выбранной форме - доклады, проекты, изделия).

Итоговый праздник (в выбранной форме).

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

Группа № ____

I год обучения

Дата занятия		Название раздела, темы	Всего (час)	Примечание
Планируемая	Фактическая			
Введение			2	
1.		Введение в программу. Правила техники безопасности при работе на компьютере, правила сетевой этики.	1	
2.		Пробный проект	1	
Тема 1. Компьютер и пользователь			4	
3.		История отрасли ИТ, виды и жанры программ	1	
4.		Наш компьютер: стандартные программы	1	
5.		Английский язык в отрасли ИТ	1	
6.		Клавиатура, кодировки, другие устройства ввода	1	
Тема 2. Компьютер и разработчик			4	
7.		Аппаратная схема программируемых устройств	1	
8.		Схема, логика, назначение операционной системы	1	
9.		Двоичный элемент, системы счисления, бит	1	
10.		Современная оперативная и дисковая память, байт	1	
11.		УММ1 Презентации о профессии		
Тема 3. Основы языка программирования: данные, условия, циклы			8	
12.		Переменная, ввод и вывод. Случайное число	1	
13.		Арифметические задачи	1	
14.		Простые условия	1	
15.		Схема алгоритма, подпрограмма	1	
16.		Составные условия, вложенные операторы	1	
17.		Цикл со счётчиком, числовые ряды	1	
18.		Вложенный цикл. Цикл без счётчика.	1	
19.		Циклы при решении задач	1	
Тема 4. Графический проект			8	
20.		Графическое приложение. Пиксел, цвет, координаты	1	
21.		Графический орнамент	1	
22.		График функции	1	
23.		Управление объектом с клавиатуры	1	
24.		Самостоятельное движение объекта, таймер	1	
25.		Текстурирование и способы украшения	1	
26.		Снежинка как параметрическая функция	1	
27.		Дизайн новогодней открытки	1	
Тема 5. Массивы и их применение			8	
28.		Одномерный массив: ввод и вывод	1	
29.		Алгоритмы на массиве: поиск, разворот	1	
30.		Движение группы объектов. «Падающий снег»	1	
31.		Работа с индивидуальными проектами	1	
32.		УММ 2 Фестиваль новогодних компьютерных работ		
33.		Алгоритмы на массиве: сортировка	1	
34.		Двумерный массив: ввод и вывод	1	
35.		Игровое поле на двумерном массиве	1	
36.		Работа с индивидуальными проектами	1	
Тема 6. Структуры, файлы, алгоритмы			18	
37.		Разбор алгоритмов игр для двумерного поля	1	
38.		Алгоритмы для лабиринта: поиск пути, бот		
39.		УММ3 Новостная лента технологий	1	
40.		Геометрия соприкосновения фигур	1	
41.		Физика перемещения персонажа	1	

42.		Комбинаторика. Перебор	1	
43.		Многомерный массив или динамическая память	1	
44.		Абстрактные структуры: списки, деревья, графы	1	
45.		УММ4 Весенний проект	1	
46.		Файл и файловые системы		
47.		Чтение, запись переменной в файл	1	
48.		Чтение, запись структур в файл	1	
49.		Таблица рекордов с сортировкой	1	
50.		Интерфейс для таблицы рекордов	1	
51.		Обзор, классификация известных форматов файлов	1	
52.		Сжатие (gif)	1	
53.		Шифрование (XOR и сложнее)	1	
54.		База данных. Записная книжка	1	
55.		Очередь, стек, дек с визуализацией	1	
56.		УММ5 Городская конференция по программированию (задачи)		
57.		Работа с индивидуальными проектами	1	
58.		УММ6 Городская конференция по программированию (проекты)		
Выполнение итоговых творческих работ			10	
59.		Спектр тематик проекта	1	
60.		Реализация главного меню	1	
61.		Ведение профилей с авторизацией	1	
62.		Основной алгоритм	1	
63.		УММ7 К Дню Победы		
64.		Развитие основного алгоритма	1	
65.		Оформление основной части	1	
66.		Оформление интерфейса	1	
67.		Коррекция справки, статистики	1	
68.		Финальное тестирование	1	
69.		Репетиция докладов	1	
Итоговое занятие			2	
70.		Смотр выпускных работ	1	
71.		УММ8 Итоговый праздник		
72.		Итоги года, консультации	1	
Учебно-массовые мероприятия			8	

ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Оценочные материалы

Формы и средства выявления, фиксации и предъявления результатов обучения в рамках реализации программы

При реализации программы используются следующие оценочные материалы:

1. Бланк оценки программного продукта по 5 критериям: Алгоритм, Пользовательский интерфейс, Тестирование, Практическое применение, Качество кода. Данные критерии соответствуют критериям городского конкурса школьников по программированию и компьютерным работам.
2. Валидация кода программы с помощью онлайн-ресурса <https://ideone.com/>;
3. Автоматическое тестирование функционала приложения с помощью онлайн-ресурса emma.sourceforge.net/;
4. Карта оценки дизайн-макета пользовательского интерфейса;

5. Тематические задания по созданию скриптов определённого функционала и бланки их оценки;
6. Тематические задания по созданию элементов компьютерной игры и бланки их оценки (на 2 году обучения).

Система контроля результативности обучения

Вид контроля	Срок	Форма выявления	Форма фиксации	Форма предъявления результатов
ПРЕДМЕТНЫЕ ДОСТИЖЕНИЯ				
<i>Входной</i>	Сентябрь	Педагогическое наблюдение, собеседование	Карта наблюдений педагога. Универсальная диагностическая карта ЦДЮТТ	Карта наблюдений педагога. Универсальная диагностическая карта ЦДЮТТ
<i>Текущий</i>	В течение учебного года	Анализ творческих работ учащихся, наблюдение, беседа, опрос.	Карта наблюдений педагога. Фотоматериалы (скриншоты). Практические работы учащихся.	Творческие работы
<i>Промежуточный</i>	По окончании изучения каждой темы	Анализ качества практических работ, опрос. Участие в конкурсах различного уровня.	Фотоматериалы (скриншоты). Грамоты и дипломы. Практические работы учащихся. Карта наблюдений педагога.	Творческие работы. Грамоты, дипломы, полученные на конкурсах.
	Декабрь	Анализ качества практических работ, опрос. Участие в конкурсах различного уровня.	Фотоматериалы (скриншоты). Грамоты и дипломы. Практические работы учащихся. Карта наблюдений педагога. Универсальная диагностическая карта ЦДЮТТ	Творческие работы. Грамоты, дипломы, полученные на конкурсах.
	Май	Защита творческих работ. Участие в конкурсах различного уровня.	Творческие работы учащихся Универсальная диагностическая карта ЦДЮТТ Бланк оценки программного продукта по 5 критериям.	Защита творческих работ. Открытые занятия. Грамоты, дипломы, полученные на конкурсах.

<i>Итоговый</i>	Май	Защита творческих работ, участие в конкурсах различного уровня.	Творческие работы учащихся Универсальная диагностическая карта ЦДЮТТ Бланк оценки программного продукта по 5 критериям.	Защита творческих работ. Открытые занятия. Грамоты, дипломы, полученные на конкурсах.
ЛИЧНОСТНЫЕ КАЧЕСТВА УЧАЩИХСЯ				
<i>Входной</i>	Сентябрь	Педагогическое наблюдение	Информационная карта Универсальная диагностическая карта ЦДЮТТ	Информационная карта Универсальная диагностическая карта ЦДЮТТ
<i>Текущий</i>	В течение учебного года	Педагогическое наблюдение	Тетрадь наблюдений педагога.	Тетрадь наблюдений педагога.
<i>Промежуточный</i>	Декабрь	Педагогическое наблюдение	Информационная карта Универсальная диагностическая карта ЦДЮТТ	Аналитическая справка
<i>Итоговый</i>	Май	Педагогическое наблюдение	Информационная карта Универсальная диагностическая карта ЦДЮТТ	Защита творческих работ. Открытое занятие. Аналитическая справка
ВЗАИМООТНОШЕНИЯ В КОЛЛЕКТИВЕ				
<i>Входной</i>	Октябрь	Педагогическое наблюдение	Карта наблюдений педагога.	Аналитическая справка
<i>Текущий</i>	В течение учебного года	Педагогическое наблюдение	Карта наблюдений педагога.	Аналитическая справка
<i>Промежуточный</i>	Декабрь	Педагогическое наблюдение	Карта наблюдений педагога.	Аналитическая справка
<i>Итоговый</i>	Май	Педагогическое наблюдение	Карта наблюдений педагога.	Аналитическая справка. Защита творческих работ. Открытое занятие.

Универсальная диагностическая карта, разработанная для всех педагогов ЦДЮТТ, включает в себя образовательный и воспитательный компонент и содержит 6 параметров: самостоятельность при выполнении заданий, сложность выполненных заданий, качество выполнения заданий, культура поведения, творческие способности, активность на

занятиях в коллективе. Каждый из параметров оценивается по 4-ём уровням: 2 балла - самый низкий уровень, 5 баллов – наивысший уровень.

Диагностика уровня личностного развития учащихся производится три раза в год по следующим 8 параметрам: культура поведения; творческие способности; активность на занятиях в коллективе; коммуникативные навыки и умение работать в коллективе; стремление к самообразованию; мыслительные навыки; способность к самоорганизации деятельности; уровень информационной культуры.

Итоги диагностики педагог заносит в информационную карту, специально разработанную для данной программы, используя следующую шкалу:

<i>Оценка параметров</i>	<i>Уровень</i>
Начальный уровень - 1 балл	8- 11 баллов – начальный уровень
Средний уровень – 2 балла	12 – 20 баллов – средний уровень
Высокий уровень – 3 балла	21-24 балла – высокий уровень

Методические материалы

Используемые методы, приемы, технологии

Методы:

- словесный (устное изложение, беседа, рассказ, объяснение и т.д.)
- наглядный (наблюдение, показ (выполнение) педагогом, работа по образцу и др.)
- практический (выполнение работ самостоятельно)
- объяснительно-иллюстративный – учащиеся воспринимают и усваивают (запоминают) готовую информацию
- репродуктивный – учащиеся воспроизводят полученные знания и освоенные способы деятельности, работая по шаблону;
- частично-поисковый – создание творческих работ по собственному замыслу.

Методы стимулирования и мотивации учебно-познавательной деятельности: творческие задания, комфортная среда занятия и др.

Методы воспитания: беседы, метод примера, педагогическое требование, побуждение, создание воспитательных ситуаций, соревнование, поощрение.

Методы контроля - контрольные задания в виде творческих работ, участие в конкурсах и др.

Основные приемы – рассказ, беседа, пояснение, практическая работа, самостоятельная работа, показ образцов, демонстрация практических приемов работы на компьютере, творческая работа, проблемная постановка вопроса, устный обучающий контроль.

Используются следующие **современные педагогические технологии:**

- организация круглого стола по обсуждению тем программных продуктов;
- демонстрация процесса разработки приложения под различные операционные системы (в формате видеопрезентации и в режиме реального времени);
- личноориентированные технологии (подбор индивидуальных заданий по разработке программ с учётом уровня знаний и навыков учащихся и тематической направленности их проектов);
- проблемное обучение в формате частично-поисковой деятельности при выборе темы выпускной работы на всех годах обучения: обзор аналогов, критический анализ

предполагаемого тематического наполнения на базе списка разделов;

- исследовательская деятельность при поиске и подборе содержания для выпускной работы, а также по внедрению передовых технологий в области программирования;
- здоровьесберегающие технологии (проведение физкультминуток для предотвращения переутомления при работе за компьютером).

Дидактические средства:

Раздаточный материал по теории и практическим приёмам в области программирования:

1. Листы по языку

Демонстрационный материал:

1. Презентации

Информационные источники

Список литературы

Для педагогов:

Андреева А.Д., Воронова А.Д. Практическая психология образования: М., изд. ТЦ "Сфера", 1998.

Ауэр К., Миллер Р. Экстремальное программирование: постановка процесса. С первых шагов и до победного конца – СПб.: Питер, 2004-368 с.

Глушаков А.В., Коваль С. В., Черепнин С.А. Программирование на Visual C++, – Харьков: Фолио, 2003, 726 с.

Денисова А.Г. Учет половозрастных и индивидуальных особенностей детей в учебно-воспитательном процессе. Методические рекомендации для педагогов ДО - СПб, ЦСТТ Кировского района, 2005.

Документация DirectX для C++.

Документация к Visual C++ 6.0.

Карпов Б., Баранова Т. C++: специальный справочник – СПб: Питер, 2003 – 480 с.

Кетков А., Кетков Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель.– СПб.: БХВ-Петербург, 2001-480 с.

Крупник А.Б. Изучаем Ассемблер – СПб.: Питер, 2004- 249 с.

Николенко Д.В. Самоучитель по Visual C++ 6 – Наука и техника, 2001- 368 с.

Практическая психология образования: Учебник для студентов высших и средних специальных учебных заведений. / Под редакцией И.В. Дубровиной – М.: ТЦ «Сфера», 1998 – 528 с.

Холзнер С. Microsoft Visual C++5 с самого начала – СПб: Питер, 1998. – 480 с.

Для учащихся:

Березин Б.И., Березин С.Б. Начальный курс С и C++ - М: Диалог – Мифи.

Документация DirectX для C++. Документация к Visual C++ 6.0.

Кетков А., Кетков Ю. Практика программирования: Бейсик, Си, Паскаль. Самоучитель– СПб: БХВ-Петербург, 2001-480с.

Николенко Д.В. Самоучитель по Visual C++ 6 – Наука и техника, 2001- 368 с.

C++: специальный справочник / Карпов Б., Баранова Т. – СПб: Питер, 2003 – 480 с.

Интернет-источники:

[www.microsoft.com\com](http://www.microsoft.com/com)

www.msdn.com