

Введение

В современном мире профессия программиста становится все более популярной, хорошо оплачиваемой и востребованной на рынке труда. Высококвалифицированные специалисты, способные заниматься разработкой, эксплуатацией и сопровождением прикладного и системного программного обеспечения, требуются в самых различных сферах жизни и деятельности современного информационного общества.

Изменение взглядов общества на предметную область «Информатика» как науки, её место в системе научных знаний требует постоянных изменений в содержании преподавания учебного предмета «Информатика и ИКТ» и его раздела «Программирование» в общеобразовательных учреждениях России. Требования к предметным результатам освоения профильного курса информатики по разделу «Алгоритмизация и программирование» согласно Федеральному государственному образовательному стандарту общего образования второго поколения (далее ФГОС) включают на базовом и профильном уровнях:

- владение навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- владение умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкции программирования; умением анализировать алгоритмы с использованием таблиц;
- владение стандартными приемами написания на алгоритмическом языке программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- использование готовых прикладных компьютерных программ по выбранной специализации;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ.

К сожалению, учебный план общеобразовательного учреждения не содержит необходимого количества часов для изучения школьниками предмета «Информатика и ИКТ» на уровне, достаточном для получения высокого балла при сдаче единого государственного экзамена по информатике, который позволил бы будущему выпускнику быть конкурентоспособным при поступлении на факультет информационных технологий в ВУЗы России.

Поэтому особую актуальность приобретают программы дополнительного образования, способные не только углубить их знания в теоретической информатике, но и сформировать практические навыки в области информационных технологий и практического программирования.

Дополнительная образовательная программа «Основы программирования» предназначена для обучения школьников 9-11 классов..

Программа «Основы программирования» рассчитана на 1 год в количестве 90 часов.

Пояснительная записка

Дополнительная общеобразовательная программа «Основы программирования» является программой научно-технической направленности.

Содержание обучения курса «Основы программирования» позволяет обучить школьников 9-11 классов основам процедурного программирования, чтобы в дальнейшем обучающийся мог выбрать индивидуальную образовательную траекторию:

- выбрать для изучения другой Си-подобный язык программирования (C#, Си++);
- выбрать для изучения другой язык программирования (Pascal, Delphi, Assembler);
- выбрать другой профиль обучения;
- объединить свои знания нескольких профилей для проектной деятельности (например, «Программирование» + «Разработка мобильных приложений»);
- поступить в ВУЗ по профилю.

Это даст возможность обучающимся более полно выразить свой интеллектуальный и творческий потенциал, реализовать собственные идеи, применить освоенные навыки программирования в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий в области информационных технологий.

Цель общеобразовательной программы «Основы программирования» - обеспечить прочное и осознанное овладение обучающимися основами знаний процедурного программирования на языке Паскаль.

Основные задачи курса:

1. Знакомство с принципами структурирования, формализации информации и выработка умений строить и читать блок-схемы решения задач.
2. Развитие алгоритмического и логического стилей мышления.
3. Формирование навыков обработки информации различного типа, используя средства языка программирования Паскаль.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если обучающиеся осознают необходимость решения тех или иных задач. Если у них появляется острая необходимость к преодолению интеллектуальных трудностей, связанных с познанием, если они видят смысл в сотрудничестве с одноклассниками и учителем.

Содержание обучения, представленное в программе «Основы программирования», позволяет вести обучение школьников в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, исполняемых файлов — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики.

Проявления трудолюбия, целеустремленности и одухотворённости, возникающие при воплощении замыслов учащихся в рамках курса «Основы программирования», стимулируют развитие индивидуально-личностных качеств школьников.

Активизация познавательного процесса позволяет обучающимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Исключительно велика роль программирования для формирования мышления обучающихся, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления обучающихся.

Содержание программы «Основы программирования» не предполагает, что обучающиеся ранее были знакомы с базовыми понятиями алгоритмики. Однако, если некоторые из обучающихся имеют опыт программирования в других средах (Scratch, КуМир, ЛогоМиры и др.), процесс изучения языка Паскаль для них будет проходить более комфортно.

Обучение программированию предполагает дифференцированный подход. В частности, для одаренных обучающихся могут быть предложены задачи олимпиадного уровня, а также свой творческий потенциал отдельные обучающиеся могут реализовать через проектную деятельность (работа с графикой и анимацией).

Программа ориентирована на большой объем практических работ с использованием компьютера по всем изучаемым темам программирования.

Учебная работа может проводиться в трех формах:

1. Демонстрационная – работу на компьютере выполняет педагог, а обучающиеся воспроизводят действия на рабочих местах.

2. Фронтальная – синхронная работа обучающихся по освоению или закреплению материала под руководством педагога.

3. Самостоятельная – выполнение самостоятельной работы на компьютере с последующим контролем со стороны педагога.

При обучении программированию предполагается применение следующих педагогических технологий:

- обучение в сотрудничестве;
- погружение;
- обучение по индивидуальным образовательным траекториям;
- проектная деятельность.

Для оценивания качества освоения дополнительной общеобразовательной программы предполагается использовать зачетную систему.

В качестве измерителей учебных достижений предполагается использование таких форм, как решение индивидуальной задачи, тестирование, выполнение проектных работ.

Измерителями выше перечисленных ключевых образовательных компетенций являются следующие показатели:

- активное использование сервисов Всероссийской школьной образовательной сети (электронный журнал, виртуальное общение со всеми участниками образовательного процесса, участие в форумах, в дистанционных группах и событиях);
- участие в конкурсах и олимпиадах разных форм и уровней.

Предметом контроля являются знания, умения и навыки обучающихся, в некоторых случаях, созданные ими образовательные продукты (программы, модули), а также их внутренние личностные результаты (освоенные способы деятельности, знания, умения, готовность к саморазвитию и самоопределению), обозначенные целеполаганием курса.

Формой подведения итогов реализации дополнительной общеобразовательной программы в конце изучения профильного курса является зачет.

Рекомендации к реализации программы.

При обучении программированию рекомендуется использовать различные уровни сложности задач, в зависимости от способностей обучающихся:

1 уровень — задачи базового уровня сложности, где проверяется умение применить полученные знания и рассмотренные алгоритмы при решении стандартных задач.

2 уровень — задачи повышенного уровня сложности, где проверяется прочность усвоения знаний всех пройденных тем, а также умение устанавливать и использовать связи нового материала с пройденным при решении нестандартных задач.

3 уровень — задачи высокого уровня сложности, при решении которых обучающийся должен показать творческий подход к использованию нового материала и эффективность написания алгоритма решения сложной оригинальной задачи.

Учебно-тематический план
(90 часов)

№	Название раздела	Всего часов	Теория	Практика	Форма контроля
1.	Простейшие программы	2	1	1	
2.	Переменные	2	1	1	тест
3.	Ввод и вывод	2	1	1	
4.	Ветвление	2	1	1	тест
5.	Сложные условия	2	1	1	
6.	Циклы	4	2	2	тест
7.	Оператор выбора	2	1	1	
8.	Методы отладки программ	2	1	1	
9.	Работа в графическом режиме	4	2	2	защита проекта
10.	Процедуры	4	2	2	
11.	Анимация	4	2	2	защита проекта
13.	Функции	4	2	2	
14.	Случайные и псевдослучайные числа	4	2	2	
15.	Массивы	8	3	5	тест
16.	Моделирование	8	4	4	защита проекта
17.	Символьные строки	8	3	5	тест
18.	Рекурсивный перебор	6	3	3	
19.	Матрицы (двухмерные массивы)	8	4	4	тест
20.	Файлы	8	4	4	тест
21.	Подготовка к зачету	3	1	2	
22.	Зачет за учебный курс	3	1	2	зачет
	Итого	90	42	48	

Содержание учебного курса (90 часов)

1. Простейшие программы (2 часа)

Основные понятия. Строка и опция меню, компиляция, алгоритм, программа, алфавит.

Рассматриваемые вопросы. Инструктаж по технике безопасности и правилам поведения в компьютерном классе. Основные этапы решения задач на компьютере. Алгоритм. Типы алгоритмов. Свойства алгоритма. Способы записи алгоритмов. Язык программирования Паскаль. Знакомство с интегрированной средой Pascal ABC.Net. Структура экрана. Команды меню.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ историю появления языка Си и причины его популярности,
- ↑ основные файлы среды,
- ↑ назначение основных горячих клавиш,
- ↑ структуру программы.

Обучающийся должен уметь:

- ↑ запускать программу,
- ↑ сохранять файлы,
- ↑ пользоваться опциями меню среды Pascal ABC.Net

2. Переменные (2 часа)

Основные понятия. Тип переменной (integer, real, char), имя переменной, объявление переменных, оператор присваивания.

Рассматриваемые вопросы. Основные определения. Классификация типов данных. Структура программы. Первая программа. Целочисленные типы данных. Стандартные функции и процедуры для данных целого типа. Оператор присваивания. Вещественные типы данных в Паскаль. Особенности деления в Паскаль. Сокращенная запись операций, порядок выполнения операций.

Тест по теме «Переменные».

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ описание простых типов данных,
- ↑ действия над переменными числового типа данных,
- ↑ правила линейной записи арифметических выражений,
- ↑ порядок выполнения операций,
- ↑ требования к форматированию текста программы.

Обучающийся должен уметь:

- ↑ выполнять форматированный вывод чисел,
- ↑ преобразовывать сложные арифметические выражения в линейную форму.

3. Ввод и вывод(2 часа)

Основные понятия. Функции ввода-вывода

Спецификации формата. Оператор присваивания. Блок-схема линейного алгоритма

Рассматриваемые вопросы. Ввод чисел с клавиатуры. Вывод целых чисел. Вывод вещественных чисел. Блок-схема линейного алгоритма.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ функции ввода-вывода
- ↑ спецификации формата ввода-вывода,
- ↑ комбинированные операции.

Обучающийся должен уметь:

- ↑ осуществлять ввод и вывод данных в программе,
- ↑ применять оператор присваивания и комбинированные операции,
- ↑ составлять программы линейной структуры для решения задач.

4. Ветвление(2 часа)

Основные понятия. Условный оператор if (полная и неполная формы). Вложенность условных операторов.

Рассматриваемые вопросы. Ветвление. Условный оператор if. Решение задач с использованием разветвляющихся алгоритмов. Блок-схема разветвляющегося алгоритма. Вложенность условий. Решение задач с использованием условного оператора.

Тест по теме«Ветвление».

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ логические операции,
- ↑ назначение условного оператора if, его запись на языке Си.

Обучающийся должен уметь:

- ↑ использовать полную и неполную формы условного оператора,
- ↑ решать задачи на применение условного оператора,
- ↑ строить блок-схемы алгоритма решения задач.

5. Сложные условия(2 часа)

Основные понятия. Условие, составное условие, логические связки (or, and, not), логические выражения.

Рассматриваемые вопросы. Блок-схема сложного разветвляющегося алгоритма. Логический тип данных. Логические операции. Нахождение значений логических выражений. Составные условия. Решение задач с использованием вложенных операторов и составных условий.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ логические операции,
- ↑ назначение условного оператора if, его запись на языке Паскаль,
- ↑ правила построения и вычисления логических выражений.

Обучающийся должен уметь:

- ↑ использовать полную и неполную формы условного оператора,
- ↑ решать задачи на применение условного оператора с использованием логических выражений,
- ↑ строить блок-схемы алгоритма решения задач.

6. Циклы (4 часа)

Основные понятия. Цикл, циклический алгоритм. Оператор цикла с параметром (for). Оператор цикла с предусловием (while). Оператор цикла с постусловием (do while), переменная цикла, тело цикла, заикливание, условие выполнения (окончания) цикла, вложенные циклы.

Рассматриваемые вопросы. Циклы. Оператор цикла с предусловием while. Решение задач с использованием цикла while. Решение занимательных задач. Оператор цикла с постусловием (do while). Алгоритм Евклида. Блок-схема циклического алгоритма. Цикл с параметром. Решение задач с использованием цикла For.

Тест по теме"Циклический алгоритм".

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ синтаксис операторов цикла,
- ↑ отличия между циклами for, while, do while.

Обучающийся должен уметь:

- ↑ заменять один вид цикла на другой,
- ↑ решать задачи с использованием вложенных циклов,

↑ строить блок-схемы решения задач.

7. Оператор выбора (2 часа)

Основные понятия. Оператор выбора switch. Функция Break

Рассматриваемые вопросы. Оператор выбора switch. Выход из оператора выбора. Решение задач разветвляющейся структуры.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ назначение оператора выбора switch, его запись на языке Си.

Обучающийся должен уметь:

↑ решать задачи с использованием оператора выбора.

8. Методы отладки программ (2 часа)

Основные понятия. Отладка. Трассировка. Отключение части кода. Точки останова.

Рассматриваемые вопросы. Методы отладки программ. Поиск ошибок. Работа с окнами среды программирования. Работа с клавиатурой

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ методы отладки программ.

Обучающийся должен уметь:

↑ выполнять работу программы по шагам.

9. Работа в графическом режиме (4 часа)

Основные понятия. Библиотека для работы с графикой graphics.h. Функции для работы с графикой.

Рассматриваемые вопросы. Система координат. Преобразование координат. Решение задач.

Защита проекта по теме «Работа в графическом режиме»

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ правила работы с графической библиотекой,

↑ основные функции работы с графикой.

Обучающийся должен уметь:

↑ преобразовывать систему координат,

↑ строить графики функций.

10. Процедуры (4 часа)

Основные понятия. Метод пошаговой детализации, процедура, локальные и глобальные переменные, формальные и фактические параметры, процедура без параметров, процедура с параметрами, механизм передачи параметров, передача значений по ссылке и по значению.

Рассматриваемые вопросы. Понятие подпрограммы. Процедуры. Формальные и фактические параметры. Вызов по ссылке и по значению. Локальные и глобальные переменные и подпрограммы. Решение задач с использованием подпрограмм.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ метод пошаговой детализации,

↑ нисходящий способ программирования,

↑ особенности использования процедур.

Обучающийся должен уметь:

↑ разбивать задачу на подзадачи,

↑ описывать переменные и параметры подпрограммы,

↑ устанавливать связи между основной программой и подпрограммами.

11. Анимация (4 часа)

Основные понятия. Анимация. Отскок от поверхности. Управление клавишами. Непрерывное движение

Рассматриваемые вопросы. Анимация. Движение объекта. Выход по клавише Escape. Управление клавишами-стрелками

Защита проекта по теме «Анимация»

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ что такое анимация.

Обучающийся должен уметь:

↑ применять процедуры и функции

↑ создавать движущееся изображение,

↑ управлять анимацией с помощью клавиш

12. Функции (4 часа)

Основные понятия. Функция. Локальные и глобальные переменные. Рекурсия. Перегрузка функций.

Рассматриваемые вопросы. Функции пользователя в языке Паскаль. Объявление функций. Определение и вызов функций. Локальные и глобальные переменные. Рекурсия. Использование рекурсивных функций. Перегрузка функций.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ формат объявления, определения и вызова функций пользователя в Паскаль,

↑ особенности локальных и глобальных переменных,

↑ определение рекурсии.

Обучающийся должен уметь:

↑ применять рекурсивные функции для решения задач,

↑ применять перегрузку функций в программе.

13. Случайные и псевдослучайные числа (4 часа)

Основные понятия. Случайные числа. Псевдослучайные числа. Распределение случайных чисел.

Рассматриваемые вопросы. Случайные, псевдослучайные числа. Равномерное, неравномерное распределение. Функции для работы со случайными числами. Случайные числа в заданном интервале. Снег на экране.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ равномерное и неравномерное распределение,

↑ функции для работы со случайными числами.

Обучающийся должен уметь:

↑ применять функции для работы со случайными числами при решении задач.

14. Массивы (8 часов)

Основные понятия. Одномерные массивы. Способы инициализации массива. Вывод массива. Обработка элементов массива. Сортировка элементов массива.

Рассматриваемые вопросы. Понятие массива. Одномерные массивы. Способы задания. Доступ к элементам массива. Сортировка одномерного массива. Методы сортировок одномерного массивов. Самостоятельная работа. Поиск элементов массива. Удаление элементов из массива. Вставка элементов в массив. Перестановка элементов. Решение задач.

Тест по теме «Одномерные массивы».

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

↑ определения массива, индекса элемента,

↑ особенности использования перечисляемого и ограниченного типа данных,

↑ описание и способы задания массива,

↑ алгоритмы вставки и удаления элемента массива.

Обучающийся должен уметь:

- ↑ использовать стандартные алгоритмы анализа элементов массива,
- ↑ вставлять и удалять несколько элементов массива по условию,
- ↑ менять местами элементы массива,
- ↑ использовать в решении задачи несколько массивов.

15. Моделирование (8 часов)

Основные понятия. Модель. Моделирование. Виды моделей.

Рассматриваемые вопросы. Модель. Моделирование. Виды моделей. Вращение. Вращение с остановкой. Использование массивов.

Защита проекта по теме «Моделирование».

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ определения модели, моделирования,
- ↑ виды моделей,

Обучающийся должен уметь:

- ↑ создавать графические модели,
- ↑ создавать модель движения.

16. Символьные строки (8 часов)

Основные понятия. Определение строки в языке Паскаль. Функции для работы со строками. Строковый тип данных (String), стандартные подпрограммы для работы со строкой, таблица кодов ASCII.

Рассматриваемые вопросы. Тип данных char. Операции над символами. Тип данных string. Строковые переменные, их описание. Длина строки. Операции над строками. Стандартные функции для работы со строками. Стандартные процедуры для работы со строками. Решение задач. Самостоятельная работа. Бегущая строка. Осыпающиеся буквы. Решение творческих задач.

Тест по теме «Строки»

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ представление символов в таблице кодов ASCII,
- ↑ операции над символами, строками,
- ↑ определение и описание строки,
- ↑ правила использования стандартных подпрограмм для работы со строкой.

Обучающийся должен уметь:

- ↑ применять операции и подпрограммы работы со строкой при решении задач,
- ↑ использовать возможности текстового режима для нестандартного вывода символов при решении задач творческого характера.

17. Рекурсивный перебор (6 часов)

Основные понятия. Рекурсия. Натуральное число. Факториал.

Рассматриваемые вопросы. Рекурсивные процедуры и функции. Нахождение факториала натурального числа. Косвенная рекурсия. Глубина рекурсии. Рекурсивный поиск. Рекурсивные фигуры.

Требования к уровню подготовки обучающихся и планируемые результаты.

Обучающийся должен знать:

- ↑ определение рекурсии, косвенной рекурсии,
- ↑ правила написания рекурсивного алгоритма.

Обучающийся должен уметь:

- ↑ применять рекурсивные алгоритмы при решении задач.

18. Матрицы (двухмерные массивы) (8 часов)

Основные понятия. Двумерный массив, строка и столбец массива. Сортировка двумерного массива. Методы сортировок двумерного массива.

Рассматриваемые вопросы. Понятие двумерного массива. Описание типа массива. Доступ к элементам массива. Способы задания. Решение задач. Работа с элементами

двумерного массива. Вставка и удаление строк и столбцов. Решение задач. Перестановка элементов массива. Решение задач. Квадратные массивы. Решение задач. Сортировка двумерного массива. Способы сортировки двумерного массива. Решение задач.

Тест по теме «Матрицы»

Требования к уровню подготовки обучающихся и планируемые результаты

Обучающийся должен знать:

- ↑ определения двумерного массива, индексов элемента,
- ↑ описание и способы задания двумерного массива,
- ↑ алгоритмы вставки и удаления строк и столбцов массива.

Обучающийся должен уметь:

- ↑ использовать стандартные алгоритмы доступа к элементам массива по условию,
- ↑ вставлять и удалять несколько строк и столбцов массива по условию,
- ↑ менять местами строки и столбцы массива,
- ↑ решать задачи на использование одномерных и двумерных массивов.

19. Файлы (8 часов)

Основные понятия. Файл. Текстовый файл. Двоичный файл.

Рассматриваемые вопросы. Файл. Работа с файлами. Последовательный доступ. input.txt, output.txt. Обработка текстовых данных. Обработка числовых данных. Решение задач на массивы с использованием файлов. Особенности работы с двоичными файлами. Чтение, запись по блокам. Работа с матрицами.

Тест по теме «Файлы»

Требования к уровню подготовки обучающихся и планируемые результаты

Обучающийся должен знать:

- ↑ определение файла,
- ↑ правила работы с текстовыми файлами,
- ↑ правила работы с двоичными файлами.

Обучающийся должен уметь:

- ↑ решать задачи с использованием файлов input.txt, output.txt,
- ↑ решать задачи с использованием двоичных файлов.

20. Подготовка к зачету (3 часа)

Систематизация полученных знаний, умений и навыков. Консультация.

21. Зачет за учебный курс (3 часа)

Проверка знаний, умений и навыков

Требования к уровню подготовки обучающихся

Обучающиеся должны знать/понимать:

- ↑ понятие алгоритма, его основные свойства, способы задания;
- ↑ основные этапы решения задач на компьютере;
- ↑ основные операторы языка программирования Си;
- ↑ основные алгоритмы обработки числовой и текстовой информации с помощью языка программирования Си;
- ↑ типы данных языка программирования Си;
- ↑ статическое и динамическое выделение памяти.

Обучающиеся должны уметь/владеть/использовать в практической деятельности:

- ↑ писать правильные программы на Си для задач, используя различные алгоритмические конструкции языка;
- ↑ пользовательские подпрограммы для решения подзадач;
- ↑ выбирать необходимые типы данных, статические и динамические структуры;
- ↑ сравнивать эффективность методов решения задач.

Условия реализации программы

Учебно-методическое обеспечение программы:

- ↑ Материалы К.Ю. Полякова

Материально-техническое обеспечение:

- ↑ компьютерный класс с 10 - 12 персональными компьютерами для обучающихся;
- ↑ локальная сеть с доступом в Интернет;
- ↑ проектор и демонстрационный экран (или демонстрационный монитор);

Программное обеспечение для компьютеров:

- ↑ Borland C++

Рекомендуемая литература

1. Джонс Р.Б Стюарт Я. Програмуем на Си / Пер. с англ. и предисл. М.Л. Сальникова, Ю.В. Сальниковой. – М.: Компьютер, ЮНИТИ, 1994. – 236с.: ил.
2. Культин Н.Б. С/С++ в задачах и примерах. – СПб.: БВХ-Петербург, 2007. – 288 с.: ил.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2007. 461 с.: ил.
4. Подбельский В.В. Язык С++. – М.: Финансы и статистика, 2000. – 560 с.
5. Подбельский В.В., Фомин С.С. Программирование на языке Си. – М.: Финансы и статистика, 2000. – 600 с.
6. Пол Ирэ. Объектно-ориентированное программирование с использованием С++: Пер. с англ. / Ирэ Пол. – К.: НИПФ «ДиаСофт Лтд.», 1995. – 480 с.
7. Сайт К.Ю. Полякова krolyakov.spb.ru