

**ПРИНЯТА**

Педагогическим советом  
Муниципального бюджетного  
общеобразовательного учреждения  
«Средняя школа №4 г. Нижний Ломов»  
«30» августа 2024г.  
Протокол № 10

**Дополнительная  
общеобразовательная  
общеразвивающая программа  
технической направленности  
«Компьютерные науки»**

Возраст учащихся - 16-17 лет

Срок реализации – 1 год

Автор программы-  
Саблина Наталья Николаевна  
учитель информатики  
МБОУ «Средняя школа №4  
г. Нижний Ломов»

**Информационная карта  
к дополнительной общеобразовательной программе**

1	Наименование образовательной организации, в которой реализуется дополнительная общеобразовательная программа	Муниципальное бюджетное общеобразовательное учреждение "Средняя школа №4 г. Нижний Ломов"
2	Адрес учреждения	441151, Пензенская область, г. Нижний Ломов, ул. Крылова, д. 6 телефон 8(84154)4-70-68 <a href="mailto:email_nlomov4@list.ru">email_nlomov4@list.ru</a>
3	Полное название программы	Дополнительная общеобразовательная общеразвивающая программа «Компьютерные науки»
4	Возраст детей, на которых рассчитана образовательная программа	Программа рассчитана на подростков в возрасте 16-17 лет.
5	Срок реализации программы	Срок реализации программы – 1 год
6	Количество детских объединений, занимающихся по программе	1 объединение
7	Сведения об авторах (ФИО, год рождения, домашний адрес, телефон, уровень квалификации, должность автора образовательной программы)	Саблина Наталья Николаевна, 1978 года рождения, адрес - г. Нижний Ломов, ул. Крылова, д.1, кв.75, высшая категория, ПГПУ им. В.Г. Белинского, г. Пенза, 2005 год, учитель информатики.
8	Характеристика программы:	
	- по типовому признаку	Авторская
	- по основной направленности	Техническая
	- по уровню освоения	Ознакомительный
	по образовательным областям	Программирование
	по целевым установкам	Техническая
	по формам организации содержания	Однопрофильная
9	сведения об эффективности программы	Эффективность программы оценивается автором с точки зрения: развития устойчивой потребности у учащихся к изучению языков программирования сложного уровня; изменения отношения учащихся к саморазвитию, формирование мотивации на расширение общеобразовательного кругозора.

## Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Компьютерные науки» *технической направленности*, ознакомительного уровня освоения. Программа адресована учащимся в возрасте от 16 до 17 лет, рассчитана на 1 год обучения.

Программа реализуется в МБОУ "СШ №4 г. Нижний Ломов" и направлена на развития устойчивой потребности у учащихся к изучению языков программирования сложного уровня; изменения отношения учащихся к саморазвитию, формирование мотивации на расширение общеобразовательного кругозора.

Программа разработана с учетом действующих *нормативно-правовых актов* в области образования РФ и нормативными актами учреждения:

- Федеральным законом Российской Федерации от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации»;
- «Порядком организации и осуществление образовательной деятельности по дополнительным общеобразовательным программам», утверждённый приказом Министерства просвещения Российской Федерации от 9 ноября 2018 года №196;
- СанПиН 2.4.3648-20 «Санитарно - эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Распоряжением Правительства РФ от 24 апреля 2015 г. № 729-р «Концепция развития дополнительного образования детей»;
- Распоряжением Правительства РФ от 29 мая 2015 г. № 996-р «Стратегия развития воспитания в Российской Федерации на период до 2025 года»;
- Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Устава и Положения о дополнительной общеобразовательной общеразвивающей программе муниципального бюджетного общеобразовательного учреждения «Средняя школа №4 г. Нижний Ломов»
- Положения о промежуточной аттестации и аттестации по завершению программы.

### Актуальность программы

Программа актуальна для подростка и для общества, так как сегодня национальная экономика России и международный рынок четко и совершенно однозначно определяют приоритет не только общих фундаментальных знаний выпускников школ и ВУЗов в области алгоритмизации и программировании, знании современных объектно-ориентированных языков и прикладного направления в подготовке будущих специалистов – но и требуют реальной подготовки выпускников в овладении принципиальными (базовыми) и, одновременно, практическими знаниями и навыками. И именно данная программа дает базовый навык по написанию программ.

## **Педагогическая целесообразность программы**

Заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач **программирования**. **Программирование** мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков.

## **Отличительные особенности**

В существующих программах и УМК по информатике и ИКТ приоритетное внимание отдают освоению офисного пакета программ. Изучаются языки программирования, не предоставляющие прикладного выхода знаний учащимся. Не уделяется внимание самостоятельной проектной работе учащихся.

Данная программа ориентирована на получение одновременно и фундаментальных знаний в области информатики и ИКТ, и на формирование практического навыка применения этих знаний в процессе выполнения учебных проектов в рамках реально используемых в экономике, промышленности и науке современных компьютерных и информационных технологий и подходов.

## **Принципы** лежащие в основе программы

- доступность (простота, соответствие возрастным и индивидуальным особенностям);
- демократичность и гуманизм (взаимодействие воспитателя и воспитанника в социуме, реализация собственных творческих потребностей);
- научность (обоснованность, наличие методологической базы и теоретической основы);
- «от простого к сложному» (научившись элементарным навыкам работы

## **Новизна программы**

Заключается в изучении языков программирования, представляющих прикладной выход знаний учащимся, а также параллельное изучение алгоритмических структур на нескольких языках программирования одновременно.

## **Цель:**

приобретение учащимися базового набора знаний, умений и навыков по программированию на языке Java(параллельно разбор данных задач на языках программирования Pascal ABS.NET и Python)

## **Задачи:**

- научить учащихся основам программирования с использованием системы программирования Java(параллельно изучаются языки программирования Pascal ABS.NET и Python);
- научить составлению и оформлению программ (правила хорошего тона в программировании) в соответствии с нормативными требованиями языка программирования;
- развивать логическое и аналитическое мышление школьников
- расширить кругозор и познавательные интересы у учащихся, формировать умения применять на практике знания, полученные во время занятий;
- развивать универсальные учебные действия, такие как: умения учащихся осуществлять целеполагание, планирование, прогнозирование, контроль, коррекцию, оценку, саморегуляцию.

### **Адресат программы**

Программа адресована подросткам 16-17 лет.

Возрастной диапазон 16-17 лет. Подростки 16 лет умеют рассуждать по-взрослому, спорить, аргументируя свою точку зрения, строить логические цепочки, предугадывать возможный ход событий и т.д., то есть имеет все предпосылки для успешного изучения программированию.

Программа рассчитана на 1 год обучения, недельная нагрузка составляет 2 часа. В год - 72 часа. Наполняемость группы – 15 человек.

**Форма обучения** – очная, очно – заочная, заочная

### **Срок реализации программы 1 год**

Всего часов по программе: 72 часа.

### **Режим занятий**

Занятия проводятся 2 раза в неделю по часу. Продолжительность одного академического часа – 45 минут.

### **Методы и формы обучения**

При обучении программированию используются вербальные (при изложении лекционного материала) и практические (выполнение лабораторных работ, практикумов, решение задач) методы, причем основной акцент делается на практические методы, в процессе применения которых воспитанники не только получают новые знания, но и приобретают практические навыки. Учитель при этом инструктирует, указывает цели работы, направляет и проверяет ход ее исполнения. В деятельности учащихся преобладает практическая работа, в ходе которой особую роль играет самостоятельный мыслительный процесс, позволяющий осуществить поиск данных и парадигмы решения задачи. По основным видам дидактических проблем, решаемых на занятии, можно выделить методы приобретения знаний, формирования умений, применения знаний, методы творческой деятельности и методы проверки знаний, умений и навыков.

Ведущей формой организации занятий является индивидуальная. Наряду с индивидуальной формой работы, во время занятий осуществляется групповая работа. Каждое занятие носит практико-ориентированный характер.

### **Планируемые результаты обучения:**

Данная программа ориентирована на получение одновременно и фундаментальных знаний в области информатики и ИКТ, и на формирование практического навыка применения этих знаний в процессе выполнения учебных проектов в рамках реально используемых в экономике, промышленности и науке современных компьютерных и информационных технологий и подходов.

В результате освоения данного курса воспитанник

#### **должен знать:**

- требованиям к дружественности интерфейса современных компьютерно-информационных продуктов;
- основным свойствам алгоритма;
- основным алгоритмическим конструкциям;
- выбирать соответствующие требованиям эффективного решения (создания программы) инструменты языка;
- как пользоваться справочными системами и другими источниками справочной информации. Оценивать достоверность информации.

#### **должен уметь:**

- планировать, создавать и отлаживать программы на языке программирования (Java, Pascal, ABS.NET и Python);
- создавать программы в соответствии с поставленной общей задачей;
- методам введения, обработки и вывода информации;
- выполнять базовые операции над объектами: цепочками символов, числами, проверять свойства этих объектов;
- реализовывать в программной среде типы алгоритмических конструкций: следование, ветвление, цикл; понятие вспомогательного алгоритма – и пути их реализации в рамках доступных программно-информационных инструментов и продуктов;
- диагностировать ошибки программного обеспечения и устранять простейшие неисправности его работы.
- интерпретировать результаты, получаемые в ходе исполнения реальных программ;
- оценивать сравнительную эффективность разных вариантов практического решения поставленной задачи (разных программ, достигающих сходного результата);
- выбирать адекватные программные и аппаратные средства для решения поставленной задачи.

**Планируемые результаты** (личностные, метапредметные и предметные результаты, получаемые учащимися в результате освоения программы).

**Личностные результаты** – развитие положительных личностных качеств учащихся (трудолюбия, упорства, настойчивости, умения работать в коллективе, уважение к людям).

**Метапредметные результаты** – формирование универсальных учебных действий (УУД).

**Регулятивные УУД:**

- понимать задачи, поставленные педагогом;
- знать последовательность выполнения работы;
- правильно оценивать результаты своей деятельности.

**Познавательные УУД:**

- научиться проявлять технические способности;
- развивать сенсорные и моторные навыки.

**Коммуникативные УУД:**

- учиться работать в коллективе;
- проявлять индивидуальность и самостоятельность.

**Предметные результаты** – формирование практического навыка применения знаний в процессе выполнения учебных проектов в рамках реально используемых в экономике, промышленности и науке современных компьютерных и информационных технологий и подходов.

### **Формы контроля и оценочные материалы**

Результативность освоения программного материала отслеживается систематически в течение года с учетом уровня знаний и умений учащихся на начальном этапе обучения. С этой целью используются разнообразные виды контроля:

- **входящий контроль** проводится в начале учебного года для определения уровня знаний и умений учащихся на начало обучения по программе;
- **текущий контроль** в виде педагогического наблюдения за качеством овладения учащимися приемами и навыками работы с различными возможностями программы;
- итогового обсуждения, анализа выполненных работ в первую очередь самими учащимися, затем педагогом в конце каждого занятия; задач и заданий по скоростному программированию в конце каждой темы.

### **Учебно-тематический план**

№ урока	Тема	Количество часов		Формы контроля
		теория	практика	
1	Основные моменты истории программирования	1	0	-
2	Базовая структура программы на Java. Значение комментариев при написании программы. Этапы работа: написание, отладка, прогон.	0,5	0,5	Тест
3	Переменные и основные типы переменных. Объявление и инициализация переменных.	0,5	0,5	Решение задачи
4	Символьные и строковые типы.	0,5	0,5	-

5	Команды вывода. Команды ввода.	0,5	0,5	Решение задачи
6-7	Практикум	0	2	Соревнование по скоростному программированию
8	Присвоение данных между разными типами переменных. Инициализация переменных. Специальные операторы.	0,5	0,5	-
9	Операторы «вычисление остатка» и «вычисление частного».	0,5	0,5	Решение задачи
10-11	Практикум	0	2	Соревнование по скоростному программированию
12	Оператор "ветвления" (выбора) – общие соображения.	1	0	-
13	Простой if (без else и с единственным условием).	0,5	0,5	Решение задачи
14	Простой if (с else и с единственным условием).	0,5	0,5	Решение задачи
15	Практикум	0	1	Соревнование по скоростному программированию
16	Простой if с блоком (блоками).	0,5	0,5	Решение задачи
17-18	Практикум	0	2	Соревнование по скоростному программированию
19	Оператор if со сложным условием.	0,5	0,5	Решение задачи
20	Практикум	0	1	Соревнование по скоростному программированию
22-23	Логический тип (булевы значения – «истина» и «ложь»).	0,5	1,5	-



23-24	Наиболее используемые функции библиотеки Math. Использование функции random – работа со случайными числами.	0,5	1,5	Решение задачи
25-26	Практикум	0	2	Соревнование по скоростному программированию
27	Цикл for	0,5	0,5	-
28	Цикл for – использование счетчика.	0,5	0,5	Решение задачи
29-30	Цикл for – использование сумматора.	0,5	1,5	Решение задачи
31	Цикл for – определение максимума и минимума.	0,5	0,5	Решение задачи
32-36	Практикум		5	Соревнование по скоростному программированию
37-40	Символьные и строковые типы.	1	4	-
41-43	Структура и работа цикла while.	0,5	2,5	Соревнование по скоростному программированию
44-45	Практикум	0	2	Решение задачи
46-49	Вложенные циклы.	0,5	3,5	Соревнование по скоростному программированию
50-53	Использование методов в классах (программах) на языке Java.	0,5	3,5	Соревнование по скоростному программированию
54-56	Массив – описание, инициализация, свойства.	1	2	Соревнование по скоростному программированию
57-59	Массивы и методы.	1	2	Соревнование

				по скоростному программирова нию
60-62	Массивы счетчиков и массива сумматоров.	1	2	Соревнование по скоростному программированию
63-65	Практикум	0	3	Решение задачи
66-68	Рекурсивные алгоритмы	0,5	2,5	Соревнование по скоростному программированию
69-72	Практикум.		4	Решение задачи
Итого		16	56	

### Содержание (72 часа)

#### **1. Основные моменты истории программирования (1 час)**

История языков программирования. Классы языков программирования. Машинные языки. Машинные коды. Языки высокого уровня. Ассемблер. Языки Фортран, Паскаль, Бейсик, С. История возникновения языка Java.

#### **2. Базовая структура программы на Java**

##### **Значение комментариев при написании программы. Этапы работа: написание, отладка, прогон (1 час)**

Обязательные элементы программы (класса) на языке Java. Краткое объяснения каждого обязательного элемента. Допустимые изменения. Назначения каждой основной части класса. Комментарии и исполняемые части класса. Значение комментариев и правила их написания. Редактор DrJava. Правила установки и запуска. Основные этапы создания класса. Написание кода и возможные ошибки. Отладка кода и правила исправления ошибок. Сообщения об ошибках. Прогон класса и логические ошибки.

#### **3. Переменные и основные типы переменных.**

##### **Объявление и инициализация переменных (1 час)**

Смысл понятия "переменная". Цель использования переменных в классе. Переменные и значения (данные). Типы данные и типы переменных. Разнообразие типов данные и переменных. Правило соответствия типа данных и типа переменных – и наоборот. Объявление переменных. Возможные варианты объявления переменных. Инициализация переменных. Возможные варианты инициализации переменных. Объединение объявления переменных с инициализацией и отдельное выполнение объявления и инициализации. Возможные ошибки и их исправление.

#### **4. Символьные и строковые типы (1 час)**

Математические и нематематические типы данные и переменных. Особенности нематематических типов. Определение символьного и строкового типов. Объявление и инициализация символьного и строкового типов. Возможное использование этих типов в классах (программах).

#### **5. Команды вывода. Команды ввода. (1 час)**

Статический по отношению к исполнению класс. Идея динамичности (интерактивности) связи между переменными и данными. Изменение значения переменных. Понятие ввода данные с клавиатуры. Использование метода Scanner и идеология "переменной-диспетчера". Команды ввода для различных типов данных. Команды вывода данные на экран. Два варианта команд вывода данные на экран и особенности их исполнения. Понятие о "дружественном вводе и выводе данных".

#### **6. Практикум (2 часа)**

Отработка практических навыков самостоятельного написания, отладки и прогона классов с использование переменных разных типов, вводом данные разных типов с клавиатуры, вывода данные на экран разными способами. Требование "дружественности" и использование комментариев.

#### **7. Присвоение данных между разными типами переменных. Инициализация переменных. Специальные операторы. (1 час)**

Возможности изменения значений для занесения их в переменные разных типов. Изменение величины значения и изменение типа значения. Команды изменения типа значений для установления соответствия между типами данных и типами переменных. Операторы выполнения вычислительных (математических) операций и типы данных и переменных. Специальные операторы для "приведения" типов. Возможные ошибки и правила их исправления.

#### **8. Операторы «Вычисление остатка» и «вычисление частного» (1 час)**

Оператор деления и типы переменных в вычисляемом выражении. Особенности оператора – многозначимость и зависимость исполняемой операции от типа переменных. Деление и вычисление остатка – особенности математического смысла и особенности записи оператора в классе. Оператор вычисления частного. Возможные ошибки – синтаксические и логические. Примеры заданий на использование вычисления результата деления, остатка и частного.

#### **9. Практикум (2 часа)**

Отработка навыков написания классов с использование деления, вычисления остатка и частного при делении. Задачи из курса математики и их реализация в классах (программах). Задачи общего типа, в которых вычисления результатов деления, значений остатка и частного имеют "общий" смысл.

#### **10. Оператор "ветвления" (выбора) – общие соображения (1 час)**

Последовательный принцип выполнения команд класса. Принцип выбора в "жизненных" ситуациях и необходимость моделирования выбора в программировании. Понятие ветвления и особенности исполнения команд класса при ветвлении. Выбор и условие. Логические выражения и логические значения. Понятие условия. Исполнение выбора в зависимости от образующегося в условии логического значения.

### **11. Простой if (без else и с единственным условием) (1 час)**

Команда выбора в минимальной форме. Условие в команде выбора и правила его записи. Логика исполнения команды выбора в минимальной форме. Использование команд и выражений в операторе выбора.

### **12. Простой if (с else и с единственным условием) (1 час)**

Команда выбора в полной форме. Логика исполнения команды выбора при выполнении условия. Логика выполнения команды выбора при неисполнении условия. Использование команд и выражений в команде выбора в полной форме. Понятие "игнорирования" команд.

### **13. Практикум (1 час)**

Отработка навыка написания классов с использованием минимальной и полной форм оператора выбора. Использование в классе нескольких команд выбора. Влияние команд выбора на результаты вывода данных на экран и значения переменных. Возможные синтаксические ошибки и их исправление. Возможные логические ошибки и их устранение.

### **14. Простой if с блоком (блоками) (1 час)**

Понятие блока. Правила написания блока в классе (программе). Использование блока в операторах выбора, правила написания таких блоков. Возможные ошибки и их исправление.

### **15. Практикум (2 часа)**

Отработка навыков написания классов с использованием команд выбора, содержащих блоки.

### **16. Оператор if со сложным условием (1 час)**

Сложное условие и его связь с простым условием. Понятие логических связей между условиями. Логическое "и". логическое "или". Правила написания сложного условия. Приоритеты выполнения логических вычислений. Основные правила логических действий. Возможные синтаксические ошибки при написании сложных условий. Логические ошибки при написании сложных условий.

### **17. Практикум (1 час)**

Отработка навыков написания классов с использованием операторов выбора, содержащих сложные условия. Возможные ошибки и их устранение.

### **18. Логический тип (булевы значения – «истина» и «ложь») (2 часа)**

Переменные логического (булевого) типа и значения логического (булевого) типа. Значение "истина" и значение "ложь". Связь между переменными логического типа и математическими выражениями. Использование переменных логического типа. Особенности использования переменных логического типа в командах вывода на экран.

### **19. Наиболее используемые функции библиотеки Math.**

#### **Функция random – работа со случайными числами (2 часа)**

Необходимости использования математических и логических действий, которые не всегда можно выразить в виде стандартных выражений. Понятие системной библиотеки в языке Java. Системная библиотека Math. Правила использования этой библиотеки. Понятие случайного числа и смысл генерации случайных чисел. Правило генерации случайного числа с использованием системной библиотеки Math. Преимущества и недостатки использования генерации случайных чисел как замены вводу данных с клавиатуры.

#### **20. Практикум (2 часа)**

Отработка навыков написания классов с использованием системной библиотеки Math. Отработка навыков написания классов с использованием случайных чисел. Возможные ошибки (синтаксические и логические) и их исправление.

#### **21. Цикл for (1 час)**

Необходимость повторных действий в программировании. Ситуации с повторными действиями в повседневной жизни и проблема их моделирования в программировании. Понятие цикла как инструмента повторного исполнения. Возможные варианты повторного исполнения. Понятие о цикле for. Структура цикла. Заголовок цикла и тело цикла. Правила написания заголовка цикла for. Понятие о переменной-счетчике повторов. Особенности описания и инициализации переменной-счетчика повторов. Варианты написания заголовка цикла for. Особенности написания тела цикла for. Возможные ошибки (синтаксические и логические) при использовании цикла for.

#### **22. Цикл for – использование счетчика (1 час)**

Отработка навыков написания классов с использованием цикла for. Поэтапное усложнения цикла. Особенности выбора переменной-счетчика повторов. Особенности написания классов с циклом и командами ввода и вывода. Особенности использования в цикле генерации случайных чисел. Понятие "счетчика случаев". Требования к инициализации переменной-счетчику случаев. Использование переменной-счетчика случаев в цикле.

#### **23. Цикл for – использование сумматора (2 часа)**

Понятие "сумматора". Примеры использования сумматора в жизни и необходимость моделирования этих ситуаций в программировании. Требования к инициализации переменной-счетчику случаев. Использование переменной-счетчика случаев в цикле. Сходство и отличие между счетчиком случаев и сумматором.

#### **24. Цикл for – определение максимума и минимума (1 час)**

Понятие максимума и минимума. Примеры использования максимума и минимума в жизни и необходимость моделирования этих ситуаций в программировании. Определение максимума и минимума с помощью оператора выбора. Определение максимума и минимума с помощью системной библиотеки Math. Использование цикла для определения максимума и минимума в серии данных.

### **25. Практикум (5 часов)**

Отработка навыков написания классов с использованием циклов для выполнения подсчетов (числа случаев и суммы значений) и определения максимума и минимума. Использование в этих классах команд выбора, случайных чисел системной библиотеки Math.

### **26. Символьные и строковые типы (5 часов)**

Работа с символьными и строковыми типами данных. Операции над символами и строками.

### **27. Структура и работа цикла while (3 часа)**

Понятие цикла с "предустановленным" числом повторов и понятие "гибкого" цикла. Цикл while. Особенности этого цикла. Понятие "условия повтора". Принципиальные различия между циклами for и while. Анализ поставленной задачи с точки зрения выбора подходящего к решению типа цикла. Цикл while и использование логических переменных.

### **28. Практикум (2 часа)**

Отработка навыков написания классов с использованием цикла while. Возможные синтаксические и логические ошибки и их исправление.

### **29. Вложенные циклы (4 часа)**

Использование в одном классе нескольких циклов. "Соседние" циклы и "цикл в цикле" (вложенные циклы). Примеры использования вложенных циклов в жизни и необходимость моделирования этих ситуаций в программировании. Правила исполнения вложенных циклов. Возможные синтаксические и логические ошибки и их устранение.

### **30. Использование методов в классах (программах) на языке Java (4 часа)**

Главный метод и класс. Команды и главный метод. Дополнительные методы и их связь с главным методом. Понятие метода. Особенности выполнения класса, содержащего методы. Вызов метода (обращение к методу). Передача данных методу. Методы, возвращающие значение и методы, не возвращающие значений. Заголовок метода и тело метода. Правила написания заголовка метода. Команда return в методах, возвращающих значения. Типы методов. Классы с использованием методов и классы без использования методов – сравнение (преимущества и недостатки). Возможные ошибки (синтаксические и логические) и их устранение.

### **31. Массив – описание, инициализация, свойства (3 часа)**

Необходимость хранения в классе одновременно большого числа значений. "Стандартная" переменная и переменная с внутренней структурой. Понятие массива. Использование массива для хранения большого числа значений одновременно. Доступ к значениям массива. Описание массива в зависимости от типа данных для хранения. Инициализация массива. Ввод данных в массив (заполнение массива). Возможные ошибки при использовании массива и их устранение. Понятие длины (размера) массива. Использование массива и циклы.

### **32. Массивы и методы (3 часа)**

Использование методов при работе с массивами. Массив в главном методе и массив в дополнительных методах. Глобальный и локальный массив. Возможные ошибки (синтаксические и логические) и их устранение.

### **33. Массивы счетчиков и массива сумматоров (3 час)**

Использование массива для случаев, когда необходимы большое количество счетчиков и сумматоров. Правила использования массива счетчиков и массива сумматоров – в отличие от правил использования массива данных. Рекурсивные алгоритмы

### **34. Практикум (3 часа)**

Отработка навыков написания классов (программ) с использованием массивов (данных, счетчиков и сумматоров) и всех изученных до этого инструментов (ввода и вывода данных, команд выбора, системной библиотеки Math, случайных чисел, циклов и так далее).

### **35. Рекурсивные алгоритмы (3 часа)**

Использовать рекурсию для составления программы с использованием рекурсивных функций; выражений. Применять рекурсивные функции при составлении программ

### **36. Практикум (4 часа)**

Отработка навыков написания классов (программ) с использованием всех изученных до этого инструментов (ввода и вывода данных, команд

## **Условия реализации программы**

### **Организационно-методические основы реализации программы**

Программа предусматривает преподавание материала от простейшего к сложному, то есть на каждом занятии используется разноуровневый материал для детей с разным уровнем логического мышления (от простейших задач к задачам высокого уровня сложности)

Программа предполагает работу с детьми в форме занятий, совместной работы учащихся с педагогом, а также их самостоятельной деятельности.

Для закрепления полученного учебного материала используются разноплановые задания, в том числе и задачи олимпиадного уровня сложности и задачи их ЕГЭ. Теоретический материал, небольшой по объему, дается перед практической частью.

Основная часть урока посвящена практической деятельности по программированию различных задач:

Программа ориентирует учащихся на самостоятельность в поисках плана решения задачи и на уроках практикумах самостоятельность в выборе языка программирования.

В процессе работы необходимо систематически напоминать детям о правилах техники безопасности при работе в компьютерном классе.

Планируемая на занятиях практическая работа не является жестко регламентированной по времени, каждый воспитанник работает в своем темпе. Это позволяет повысить уровень знаний и интереса учащихся к программированию.

При проведении занятий учитывается:

- уровень знаний, умений и навыков учащихся,
- самостоятельность ребенка,
- его активность,
- его индивидуальные особенности,
- особенности памяти, мышления и познавательные интересы.

**Формы занятий содержат в своей основе три блока:**

- групповые, индивидуальные, парные;
- теоретические и практические;
- учебные и практикумы.

На занятиях востребованы технологии развития критического мышления

Вопросы укрепления психического и физического здоровья, развития навыков социальной компетенции решаются посредством, тренингов, дискуссий, бесед с учащимися и родителями, других форм в рамках психолого-педагогического сопровождения.

#### ***Кадровое обеспечение***

Реализовать программу может педагог, имеющий высшее педагогическое образование, обладающий достаточными знаниями и опытом практической работы с учащимися старших классов.

#### ***Материально-техническое обеспечение***

Компьютеры или ноутбуки с установленными языками программирования.  
Стенд по правилам техники безопасности.

#### ***Оборудование и материалы***

Компьютеры, ноутбуки, проектор, ЛВС

#### ***Методическое обеспечение***

Электронные учебники по языкам программирования

#### **Литература:**

1. <http://www.it-village-penza.ru/?modul=1409120787>
2. <https://kpolyakov.spb.ru/school/ege.htm>
3. <https://pythonworld.ru/samouchitel-python>



