

I. Комплекс основных характеристик программы

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ JAVA» технической направленности, рассчитана на обучение детей 13-16 лет в течение 1 года.

Содержание программы основано на следующих нормативно-правовых документах:

* Федеральный Закон РФ от 29.12.2012 г. № 273 «Об образовании в РФ» (с изменениями и добавлениями);

* Приказ Министерства образования и науки Российской Федерации от 9 ноября 2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

* Постановление Главного государственного санитарного врача РФ от 28.09.2020 N 28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

* Постановление Главного государственного санитарного врача РФ от 28 января 2021 года N 2 об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;

* Распоряжение Правительства РФ от 29 мая 2015 года № 996 – р «Об утверждении Стратегии развития воспитания в РФ на период до 2025 года»;

* Федеральный проект «Успех каждого ребенка» (утвержден протоколом заседания комитета по национальному проекту «Образование» от 07.12.2018 г. №3);

* Письмо Минобрнауки России № 09-3242 от 18.11.2015 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»;

* Устав МАОУ ДО ЦДТТ;

* Локальные акты учреждения: «Положение о порядке разработки и оформления дополнительной общеобразовательной общеразвивающей программы МАОУ ДО ЦДТТ», «Положение о формах периодичности и порядке текущего контроля успеваемости и промежуточной аттестации учащихся МАОУ ДО ЦДТТ».

Актуальность программы для общества

Национальная экономика современной России, с постоянно увеличивающейся ролью и долей инновационных технологий, международный динамично развивающийся рынок информационных технологий – демонстрируют непрерывно растущую количественно и все более жесткую качественно потребность в IT-специалистах. При подготовке кадров для данной сферы необходима преемственность обучения на всех ступенях образования. Выпускник российской школы, решивший стать участником этого процесса, должен сегодня иметь знания сверх тех, дополнительные к тем, которые он получает в рамках базового курса информатики. Он должен владеть практическими навыками работы с компьютерными и информационными технологиями, быть способным создавать программные и информационные продукты начального (но уже достаточно профессионального) уровня. Эффективное обучение в ВУЗе, успешное трудоустройство и карьерный рост напрямую зависят от степени профессиональных компетенций в сфере компьютерных и информационных технологий.

Актуальность программы для учащегося

Актуальность программы обусловлена следующими факторами:

- она соответствует тенденциям развития дополнительного образования и, согласно Концепции развития дополнительного образования, способствует формированию и развитию навыков операционного и логического мышления, исследовательских и творческих качеств личности;

-программа отвечает современным потребностям детей и родителей в получении теоретических знаний и навыков в области программирования для личностного развития и успешной социализации в современном информационном обществе;

-программа не дублирует общеобразовательные программы в области информатики, а дополняет курс информатики и информационно-коммуникационных технологий общеобразовательной школы, необходимый учащимся для поступления в профильные классы и колледжи по специальностям, связанным с программированием или обслуживанием компьютеров.

Педагогическая целесообразность

Педагогическая целесообразность программы заключается в:

-получении подростками знаний и приобретению навыков, которые необходимы современному школьнику в условиях всевозрастающего интереса к информационным технологиям и их возможностям;

-предоставлении большого и постоянно увеличивающегося числа междисциплинарных связей, причем как на уровне понятийного аппарата, так и на уровне инструментария;

-возможности получения ключа к пониманию многочисленных явлений и процессов в различных областях жизни;

-приобретении знаний и умений в области IT-технологий для личностного развития учащихся и создания основы для их будущего выбора крайне востребованной в современном обществе профессии программиста;

-в повышении информационно-коммуникативной компетенции учащихся (способность осуществлять различные операции с информацией; строить общение, работать в коллективе; проявлять ответственность, самостоятельность; применять технические знания и навыки в различных сферах деятельности).

Отличительные особенности программы

Отличительные особенности программы определяются содержанием учебного материала и его структурой.

Новизна программы

Новизна программы заключается в создании благоприятных условий для более раннего, по сравнению со школьной программой, «погружения» школьников в мир основ программирования, что предоставляет учащимся возможность выбора дальнейшего профессионально-ориентированного обучения в соответствии с их способностями и предпочтениями.

Данная учебная 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.

Цель программы:

повышение информационно-коммуникативной компетентности учащихся в области IT-технологий через практическое овладение основами языка программирования Java.

Задачи программы:

развивать у учащихся логическое и алгоритмическое мышление, способности к формализации;

формировать теоретические знания по языку программирования Java;

формировать практические умения и навыки разработки алгоритмов и написания программ (кодов) на языке Java;

формировать умение видеть проблемы (при изучении школьных предметов, в повседневной жизни) и находить пути их решения с помощью самостоятельно

разработанных программ на языке Java.

Адресат программы: учащиеся среднего школьного возраста: 13-16 лет (7-11 классы). В объединение могут быть приняты все желающие. Группы детей (10-15 человек) постоянны.

**Возрастные особенности детей,
участвующих в реализации программы**

13-16 лет - средний школьный возраст (подростковый): переход от детства к взрослости. Возраст учащихся, участвующих в реализации данной программы, характеризуется тем, что это период бурного и неравномерного физического развития.

К 13-16 годам характер уже, как правило, сформирован, но существует неустойчивая самооценка, комплексы. Молодые люди любят дебаты и споры, их очень интересуют собственные особенности и свой внешний облик. Они хотят социального утверждения. Устанавливаются эмоциональные контакты с взрослыми на более высоком сознательном уровне.

Занимаясь техническим творчеством, а конкретно программированием в объединении «Информатика и вычислительная техника», подросток реализует свою потребность в самообразовании, получает новые знания и конкретный практический опыт, а также имеет возможность определения наличия у него склонностей и интересов к данной непростой сфере деятельности.

Категория учащихся 13-16 лет (7-11 классы).

Срок реализации программы 1 год.

Количество часов: 108.

Общее количество часов освоения программы: Занятия проводятся 1 раз в неделю по 3 часа..

Форма обучения: очная.

Группы детей (12-15 человек) постоянны.

Режим занятий:

Продолжительность занятия, час	Количество час./нед	Количество часов	Возраст учащихся
40мин.	1 раз по 2 часа, 2 модуля	144	13-16 лет

*Продолжительность 1 академического часа – 40 минут.

Особенности организации образовательного процесса

Содержание учебного материала опирается на следующие принципы обучения:

научность;

доступность;

индивидуальность;

систематичность и последовательность;

от простого к сложному;

связь обучения с жизнью;

связь теории с практикой (органическое сочетание необходимых теоретических знаний и практических умений и навыков);

наглядность (широкое использование дидактических и наглядных пособий, технических средств обучения, делающих учебно-воспитательный процесс более эффективным);

межпредметность (связь программы с другими областями науки, что повышает научный уровень обучения).

Педагог поддерживает тесную связь с родителями, вырабатывая общую стратегию и взгляды на воспитание детей. В начале учебного года проводится собрание, где родители знакомятся с программой объединения, с расписанием занятий, материальной базой объединения. В течение учебного года родители знакомятся с деятельностью ребенка в объединении.

Ожидаемые результаты обучения

Личностные результаты:

результатами обучения являются воспитание и развитие социально значимых личностных качеств, ценностных установок, раскрывающих отношение к труду, систему норм и правил межличностного общения, обеспечивающую успешность совместной деятельности.

у учащегося будут сформированы:

- широкая мотивационная основа художественно-творческой деятельности, включающая социальные, учебно-познавательные и внешние мотивы;
- адекватное понимания причин успешности (неуспешности) творческой деятельности;
- устойчивый интерес к новым способам познания.

Метапредметные результаты:

Метапредметными результатами обучения по программе является освоение учащимися универсальных способов деятельности (регулятивных, коммуникативных, познавательных), применяемых, как в рамках образовательного процесса, так и в реальных жизненных ситуациях.

Регулятивные:

Учащийся научится:

- принимать и сохранять учебно-творческую задачу;
- планировать свои действия;
- осуществлять пошаговый контроль;
- адекватно воспринимать оценку педагога;
- различать способ и результат действия;
- вносить коррективы в действия на основе их оценки и учета сделанных ошибок;
- проявлять познавательную инициативу;
- преобразовывать практическую задачу в познавательную;
- самостоятельно находить варианты решения творческой задачи.

учащийся будет уметь:

- разрабатывать алгоритмы и программы для решения различного типа задач;
- разрабатывать алгоритмы и составлять программы для решения прикладных задач из различных предметных областей (информатика, математика, биология, физика, экономика);

Коммуникативные:

учащийся сможет:

- допускать существование различных точек зрения и различных вариантов выполнения поставленной творческой задачи;
- учитывать разные мнения, стремиться к координации при выполнении коллективных работ;
- формулировать собственное мнение и позицию;
- договариваться, приходить к общему решению;

- соблюдать корректность в высказываниях;
- задавать вопросы по существу;
- использовать речь для регуляции своего действия;
- осуществлять взаимный контроль и оказывать партнерам в сотрудничестве необходимую взаимопомощь;

Познавательные:

учащийся научится:

- осуществлять поиск нужной информации для выполнения задачи с использованием учебной и дополнительной литературы в открытом информационном пространстве;
- использовать знаки, символы, модели, схемы для решения познавательных и творческих задач и представления их результатов;
- анализировать объекты, выделять главное;
- осуществлять синтез (целое из частей);
- обобщать (выделять класс объектов по какому-либо признаку);
- устанавливать аналогии;

использованию методов и приёмов технического творчества в учебном процессе и повседневной жизни.

Предметные результаты:

Предметными результатами являются доступные по возрасту сведения о программировании на языке Java, об основах культуры труда, элементарные умения предметно –преобразовательной деятельности, знания о различных профессиях и умения ориентироваться в мире профессий. Элементарный опыт творческой и проектной деятельности.

Учащийся будет знать:

- интегрированную среду разработки DrJava;
- синтаксис языка программирования Java;
- структуру программы;
- стандартные типы данных Java (простой, структурированный);
- встроенные (стандартные) функции;
- процедуры ввода / вывода;
- понятие простого и составного оператора;
- основные операторы языка:
- присваивание
- ветвление;
- все видов циклов (со счётчиком, с предусловием, с постусловием)
- базовые понятия теории алгоритмов;
- учащийся будет владеть:
- навыками разработки алгоритмов и составления программ при решении прикладных задач.

Учащийся будет уметь:

разрабатывать программы от простейших до более сложных с использованием всех типов данных и применением простых и составных операторов.

Учебно-тематический план

№ темы	Тема	Количество часов			Формы контроля
		всего	теория	практика	
1	Основные моменты истории программирования	3	3	0	Опрос

2	Переменные и основные типы переменных	6	3	3	Опрос
3	Команды вывода. Команды ввода	9	3	6	Практическая работа
4	Операторы «вычисление остатка» и «вычисление частного»	6	1	5	Практическая работа
5	Оператор «ветвления» (выбора) – общие соображения	18	3	15	Практическая работа
6	Наиболее используемые функции библиотеки Math	3	1	2	Опрос
7	Цикл for	18	6	12	Практическая работа
8	Структура и работа цикла while	9	3	6	Практическая работа
9	Использование методов в классах (программах) на языке Java	3	1	2	Практическая работа
10	Массив – описание, инициализация, свойства	15	6	9	Практическая работа
11	Практикум по программированию. Аттестация по итогам прохождения программы	18	0	18	Практическая, контрольная работа
	ИТОГО:	108	30	78	

Содержание занятий

1. Основные моменты истории программирования.

Теория История языков программирования. Классы языков программирования. Машинные языки. Машинные коды. Языки высокого уровня. Ассемблер. Языки Фортран, Паскаль, Бейсик, С. История возникновения языка Java. Значение комментариев при написании программы

Обязательные элементы программы (класса) на языке Java. Краткое объяснение 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.

Контроль Наблюдение за работой учащихся, опрос по теме.

2. Переменные и основные типы переменных.

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

Практика Решение задач, составление программ.

Контроль Наблюдение за работой учащихся, опрос по теме.

3. Команды вывода. Команды ввода.

Теория 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». 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.

Практика Решение задач, составление программ. Отработка практических навыков самостоятельного написания, отладки и прогона классов с использованием переменных разных типов, вводом данные разных типов с клавиатуры, вывода данные на экран разными способами. Требование «дружественности» и использование комментариев.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

4. Операторы «Вычисление остатка» и «вычисление частного».

Теория 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.

Практика Решение задач, составление программ. Отработка навыков написания классов с использованием 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.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

5. Оператор «ветвления» (выбора) – общие соображения.

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

Практика Решение задач, составление программ. 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.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

6. Наиболее используемые функции библиотеки Math.

Теория 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. Функция `random` – работа со случайными числами

Правило генерации случайного числа с использованием системной библиотеки Math. Преимущества и недостатки использования генерации случайных чисел как замены вводу данных с клавиатуры.

Практика Решение задач, составление программ. 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.

Контроль Наблюдение за работой учащихся, опрос по теме.

7. Цикл for.

Теория 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. 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. 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. 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.

Практика Решение задач, составление программ. 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 и системной библиотеки Math.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

8. Структура и работа цикла while.

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

Практика Решение задач, составление программ.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

9. Использование методов в классах (программах) на языке Java.

Теория 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.

Практика Решение задач, составление программ.

Контроль Анализ практической работы учащихся, педагогическое наблюдение.

10. Массив – описание, инициализация, свойства.

Теория 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. 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. 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.

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

Контроль Анализ контрольной работы учащихся, педагогическое наблюдение.

11. Практикум по программированию. Аттестация по итогам прохождения программы.

Практика Решение задач, составление программ. Отработка навыков написания классов (программ). Работа с индивидуальными проектами учащихся.

Контроль Анализ практической, контрольной работы учащихся, педагогическое наблюдение.

Календарный учебный график

№ учебной недели	Тема	Количество часов			Примечание
		всего	Т	П	
1	Основные моменты истории программирования.	3	3	0	
2	Переменные и основные типы переменных. Решение задач по программированию	3	2	1	
3	Решение задач по программированию	3	1	2	
4	Команды вывода. Команды ввода.	3	1	2	
5	Решение задач по программированию	3	1	2	
6	Решение задач по программированию	3	1	2	
7	Операторы «вычисление остатка» и «вычисление частного». Решение задач по программированию	3	1	2	
8	Решение задач по программированию	3	0	3	
9	Оператор "ветвления" (выбора) – общие соображения.	3	1	2	
10	Решение задач по программированию	3	1	2	
11	Решение задач по программированию	3	1	2	
12	Решение задач по программированию	3	0	3	
13	Решение задач по программированию	3	0	3	
14	Решение задач по программированию	3	0	3	
15	Наиболее используемые функции библиотеки Math.	3	1	2	
16	Цикл for	3	2	1	
17	Решение задач по программированию	3	1	2	
18	Решение задач по программированию	3	1	2	
19	Решение задач по программированию	3	1	2	
20	Решение задач по программированию	3	1	2	
21	Решение задач по программированию	3	0	3	

22	Структура и работа цикла while.	3	3	0	
23	Решение задач по программированию	3	0	3	
24	Решение задач по программированию	3	0	3	
25	Использование методов в классах (программах) на языке Java.	3	1	2	
26	Массив – описание, инициализация, свойства.	3	2	1	
27	Решение задач по программированию	3	2	1	
28	Решение задач по программированию	3	1	2	
29	Решение задач по программированию	3	1	2	
30	Решение задач по программированию	3	0	3	
31	Практикум программирования	3	0	3	
32	Практикум программирования	3	0	3	
33	Практикум программирования	3	0	3	
34	Индивидуальные проекты	3	0	3	
35	Индивидуальные проекты	3	0	3	
36	Резервное время	3	0	3	
	ИТОГО:	108	30	78	

II. Комплекс организационно-педагогических условий

Календарный учебный график

Год обучения	Дата начала обучения	Дата окончания обучения	Всего учебных недель	Всего учебных дней	Количество учебных часов	Режим занятий
2022-2023	01.09.2022	31.05.2023	36	36	108	1 раз в неделю по 3 часа

Оценка результативности

В процессе аттестации выясняются следующие вопросы:

- достигается ли цель учебно-воспитательного процесса;
- существует ли положительная динамика в развитии учащегося по сравнению с результатами предыдущих диагностических исследований.

В течение учебного года осуществляется следующие виды аттестаций:

-текущий контроль (после каждой темы, раздела) позволяет установить фактический уровень теоретических знаний по модулям дополнительной общеобразовательной программы, их практических умений и навыков;

-промежуточная аттестация (в конце полугодия) позволяет выявить достигнутый на данном этапе уровень обученности учащихся, соответствие его прогнозируемому и на этой основе оценить успешность выбранных форм и методов обучения, а также при необходимости скорректировать их;

-аттестация по итогам прохождения программы (в конце учебного года) позволяет

определить качество усвоения учащимися общеобразовательной программ, реально достигнутый уровень обученности детей в объединении.

При приёме учащегося в объединение педагог проводит входную или «стартовую» аттестацию, которая позволяет выявить предварительные знания, умения и навыки «стартового» уровня обученности детей и готовности их к изучению данного курса.

Помимо исследования результатов учебно-воспитательной деятельности объединения проводится анализ количества учащихся в объединении и его сохранность. Количественные данные учащихся в объединениях анализируются два раза в течение учебного года. Анализируется количество учащихся, возрастной состав, сохранность контингента учащихся, количество мальчиков и девочек.

Сроки проведения аттестаций определяется «Положением о формах периодичности и порядке текущего контроля успеваемости и промежуточной аттестации учащихся МАОУ ДО ЦДТТ».

Условия реализации программы

Кадровое обеспечение

Требования к педагогу дополнительного образования:

- высокий уровень профессионализма в области робототехники;
- высокий уровень квалификации и педагогического мастерства;
- владение современными педагогическими технологиями;
- знание современных педагогических технологий в области дополнительного образования детей учреждений научно-технической направленности;
- владение педагогической этикой;
- знание психолого-педагогических основ развития творческого и логического мышления детей.

Методическое обеспечение

Материал преподаётся в доступной и занимательной форме, что облегчает усвоение ребёнком сложных, на первый взгляд, тем. На занятиях предполагается разнообразие различных **видов деятельности**: применение практических упражнений, экспериментирования, проектирования. В целях лучшего усвоения курса применяются следующие **формы проведения занятий**: групповые, индивидуальные.

Задания и упражнения в разных темах программы подбираются с учётом **принципов**:

– целостность и гармоничность интеллектуальной, эмоциональной, практико-ориентированной сфер деятельности личности;

– практико-ориентированность, обеспечивающая отбор содержания, направленного на решение практических задач: планирование деятельности, поиск нужной информации, инструментирования всех видов деятельности на базе общепринятых средств информационной деятельности, реализующих основные пользовательские возможности 3D-моделирования и 3D-печати. При этом исходным является положение о том, что компьютер может многократно усилить возможности человека, но не заменить его.

– принцип развивающего обучения – обучение ориентировано не только на получение новых знаний, но и на активизацию мыслительных процессов, формирование и развитие у обучающихся обобщенных способов деятельности, формирование навыков самостоятельной работы.

– осуществление поэтапного дифференцированного и индивидуализированного перехода от репродуктивной к проектной и творческой деятельности.

– наглядность с использованием пособий, интернет-ресурсов, делающих образовательный процесс более эффективным.

– последовательность усвоения материала от «простого к сложному», в соответствии с возрастными особенностями обучающихся.

Систем занятий построена с учётом **компетентного** подхода, направленного на формирование ключевых компетенций: ценностно- смысловых, общекультурных, учебно-познавательных, информационных, исполнительских, коммуникативных; повышается практическая направленность за счёт системы творческих заданий.

Во время занятий педагог использует разные **формы организации деятельности** с детьми:

- фронтальная – учебно-познавательная часть (презентация нового материала, постановка учебной задачи, обсуждение и анализ, рефлексия полученного результата);
- групповая – практическая часть (выполнение поставленной задачи, анализ полученного результата, поиск и исправление ошибок, рефлексия – эстетично и в соответствии ли выполнена поставленная задача);
- индивидуальная - практическая часть (поиск и исправление ошибок, рефлексия);
- самостоятельная – практическая часть (создание модели по замыслу, обсуждение и анализ полученного результата, рефлексия полученного результата).

Основная форма проведения занятий – практикум. Для поддержания интереса к занятиям, используются разнообразные формы проведения занятий:

- беседы, из которых дети узнают информацию об объектах моделирования;
- самостоятельное проектирование для закрепления теоретических знаний и осуществления собственных незабываемых открытий;
- коллективные работы, где дети могут работать группами, парами, все вместе;
- соревнования (практическое участие детей в разнообразных мероприятиях);
- проектная деятельность.

Основные методы обучения, применяемые в процессе освоения программы:

1. Проблемный.
2. Частично-поисковый.
3. Исследовательский.
4. Проектный.
5. Формирование и совершенствование умений и навыков (изучение нового материала, практика).
6. Обобщение и систематизация знаний (самостоятельная работа, творческая работа, дискуссия).
8. Создание ситуаций творческого поиска.
9. Стимулирование (поощрение).

В ходе работе используются **педагогические технологии** образования, которые сориентированы на решение сложных психолого-педагогических задач (научить ребенка самостоятельно работать, общаться с детьми и взрослыми, прогнозировать и оценивать результаты своего труда, искать причины затруднений и уметь преодолевать их).

Личностно - ориентированная технология, в центре внимания которой - неповторимая личность, стремящаяся к реализации своих возможностей и способная на ответственный выбор в разнообразных жизненных ситуациях. Методическую основу этой технологии составляют дифференциация и индивидуализация обучения.

Групповая технология, которая предполагает организацию совместных действий, коммуникацию, общение, взаимопонимание, взаимопомощь, взаимокоррекцию.

Технология развивающего обучения, которая сочетает познавательную деятельность с методами активизации и развития мышления, что позволяет ребенку решать творческие и социальные задачи самостоятельно.

Технология проблемного обучения, при которой образовательный процесс строится как поиск новых познавательных ориентиров и подросток самостоятельно постигает ведущие понятия и идеи, а не получает их от педагога в готовом виде.

Коммуникативная технология обучения создает ситуации комфортного и безопасного личностно-значимого общения, взаимодействия.

Технология проектной деятельности направлена для решения поисковых, исследовательских, технологических и практических.

Здоровьесберегающие технологии, которые включают в себя:

- условия обучения учащегося в объединении (отсутствие стресса, адекватность требований, адекватность методик обучения и воспитания);
- рациональная организация учебного процесса (в соответствии с возрастными, половыми, индивидуальными особенностями и гигиеническими требованиями);
- соответствие учебной и физической нагрузки возрастным возможностям ребенка;
- необходимый, достаточный и рационально организованный двигательный режим.

В своей работе педагог использует:

- учебно-тематический план;
- методические разработки;
- дидактические материалы;
- диагностические материалы (тесты и т.п.).

Материально-техническое обеспечение

- учебный кабинет, оборудованный в соответствии с профилем проводимых занятий и санитарными нормами;
- стулья для педагога и учащихся;
- классная (магнитная) доска;
- персональные компьютеры с лицензионным программным обеспечением, включая специальное программное обеспечение по языку программирования JAVA (для педагога и учащихся); локальная сеть,
- мультимедийный проектор и экран.

Информационное обеспечение

Нормативно – правовые документы

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (принят ГД ФС РФ 21.12.2012) [Электронный ресурс]. – URL: <http://graph-kremlin.consultant.ru/page.aspx?1646176>

2. Приказ Министерства просвещения России от 09 ноября 2018 г. № 196 г. Москва «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» [Электронный ресурс]. – URL: <https://base.garant.ru/72116730/>

3. Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 г. N 28 «Об утверждении СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» [Электронный ресурс]. – URL: <http://docs.cntd.ru/document/566085656>

4. Распоряжение Правительства РФ от 29 мая 2015 г. № 996-р «Об утверждении стратегии развития воспитания на период до 2025 года» [Электронный ресурс]. – URL: <http://government.ru/docs/18312/>

5. Стратегия развития воспитания в РФ на период до 2025 года (утв. распоряжением Правительства РФ от 29 мая 2015 года №996-р) [Электронный ресурс]. – URL: <http://government.ru/media/files/f5Z8H9tgUK5Y9qtJ0tEFnyHlBitwN4gB.pdf>

6. Федеральный проект «Успех каждого ребенка» (утвержден протоколом заседания комитета по национальному проекту «Образование» от 07.12.2018 г. №3) [Электронный

ресурс]. – URL: <https://ioe.hse.ru/data/2020/07/17/1597041961/%D0%A4%D0%9F%20%D0%A3%D1%81%D0%BF%D0%B5%D1%85%20%D0%BA%D0%B0%D0%B6%D0%B4%D0%BE%D0%B3%D0%BE%20%D1%80%D0%B5%D0%B1%D0%B5%D0%BD%D0%BA%D0%B0.pdf>

7. Письмо Минобрнауки России № 09-3242 от 18.11.2015 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)» [Электронный ресурс]. – URL: http://summercamps.ru/wp-content/uploads/documents/document_metodicheskie-rekomendacii-po-proektirovaniyu-obscherazvivayuschih-program.pdf

8. Приказ Министерства образования и науки Российской Федерации и министерства просвещения Российской Федерации от 5.08.2020 г. № 882/391 «Об организации и осуществлении образовательной деятельности по сетевой форме реализации образовательных программ» [Электронный ресурс]. – URL: <http://publication.pravo.gov.ru/Document/View/0001202009110027>

9. Устав МАОУ ДО ЦДТТ [Электронный ресурс]. – URL: <http://utzar.ru/upload/docs/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B/%D0%9D%D0%BE%D0%B2%D1%8B%D0%B9-%D0%A3%D1%81%D1%82%D0%B0%D0%B2-%D0%A6%D0%94%D0%A2%D0%A2.pdf>

10. Локальные акты учреждения: «Положение о дополнительной общеобразовательной программе ЦДТТ г. Заречного», «Положение о формах периодичности и порядке текущего контроля успеваемости, промежуточной и итоговой аттестации учащихся МАОУ ДО ЦДТТ», «Положение о сетевой форме реализации дополнительных общеобразовательных программ МАОУ ДО ЦДТТ» [Электронный ресурс]. – URL: <http://utzar.ru/about/dokumenty.php>

Информационные ресурсы для педагога

1. Гудман Д. JavaScript и DHTML. Сборник рецептов. Для профессионалов. СПб: Питер, 2004.
2. Дмитриева М.В. Самоучитель JavaScript. СПб.: БХВ-Петербург, 2001.
3. Дунаев В. Самоучитель JavaScript. СПб: Питер, 2005.
4. Е.Г. Канель, В.М. Фрайман Сборник задач по курсу «Введение в компьютерные науки для учеников 10–11 классов средней школы. Язык Java» 2012.
5. Е.Г. Канель, В.М. Фрайман «Основы компьютерных наук для средней школы Язык Java» 2012.

Информационные ресурсы для детей и родителей

1. Е.Г. Канель, В.М. Фрайман Сборник задач по курсу «Введение в компьютерные науки для учеников 10–11 классов средней школы. Язык Java» 2012.
2. Е.Г. Канель, В.М. Фрайман «Основы компьютерных наук для средней школы Язык Java» 2012.

Приложение 1

Часть 1

Необходимо ответить на все вопросы 1–5. Каждый вопрос – 10 баллов.

1.

Дан алгоритм:

(1) $m \leftarrow 0$

- (2) Ввести число в переменную a
- (3) Ввести число в переменную b
- (4) Для k от a до b (включительно) выполнить
 - (4.1) Ввести число в переменную x
 - (4.2) Если $(x > a)$ и $(x < b)$
 - (4.2.1) $t \leftarrow t + 1$
- (5) Вывести значение переменной t

Проследите с помощью таблицы трассировки за выполнением алгоритма для следующего ввода: 8, 12, 4, 11, 9, 8, 35.

Запишите, каким будет вывод алгоритма.

Таблица трассировки должна включать:

- столбец для каждой из переменных алгоритма;
- столбец, показывающий, выполняется ли условие (4.2);
- столбец для выводимых данных.

2.

Напишите метод, который получит целое число, большее, чем 0, обозначающее количество колес транспортного средства. Метод вернет символ, обозначающий тип транспортного средства:

- «с» – для транспортного средства с 4 колесами;
- «b» – для транспортного средства с 8 колесами;
- «n» – для любого другого транспортного средства.

3.

Перед вами фрагмент программы:

x , y , **temp** – переменные целого типа.

```

y = x + y;
x = x - y;
y = y - x;
System.out.println ("x=" + x + "y=" + y);
if ( x >= y )
{
    temp = x;
    x = y;
    y = temp;
System.out.println ("x=" + x + "y=" + y);
}

```

А. Проследите с помощью таблицы трассировки за выполнением данного фрагмента и запишите, каким будет его вывод для следующих значений переменных: $x=3$, $y=7$.

Запишите, каким будет вывод.

Таблица трассировки должна включать:

- столбец для каждой из переменных алгоритма;
- столбец, показывающий, выполняется ли условие команды **if**;
- столбец для выводимых данных.

Б. Приведите пример значений переменных x и y , для которых будет выведена только одна строка.

4.

Дан одномерный массив a из 821 ячейки, содержащий целые числа.

Напишите фрагмент программы, который примет с клавиатуры целое число k .

Фрагмент программы выведет на экран индексы всех элементов массива, значение которых

равно k .

Примечание: Нет необходимости заполнять массив.

Нет необходимости проверять корректность вводимых данных.

5.

Напишите фрагмент программы, который будет получать пары целых чисел.

Для каждой полученной пары данный фрагмент вычислит их произведение и будет накапливать полученные произведения.

Ввод данных прекратится, когда будет введена пара чисел, произведение которых равно 0.

Фрагмент программы выведет на экран сумму всех произведений.

Примечание: Нет необходимости проверять корректность вводимых данных.

Часть 2

Необходимо ответить на 2 вопроса из вопросов 6–8. Каждый вопрос – 15 баллов.

6.

В буфете продают два вида бутербродов – один вид с творогом, а другой вид с сыром. К каждому виду бутерброда можно добавить зелень.

Каждый заказ в буфете включает в себя: количество бутербродов, вид бутерброда (с творогом или сыром), а также добавлена или не добавлена зелень.

В один заказ можно включить только один вид бутерброда. **Все** бутерброды в заказе будут с добавлением зелени или же **все** бутерброды будут без зелени.

Цена бутерброда с творогом составляет 100 руб.

Цена бутерброда с сыром составляет 120 руб.

Добавка зелени к одному бутерброду стоит 10 руб.

При заказе 10 или более бутербродов дается скидка от цены всего заказа в размере 200 руб.

Напишите программу, которая примет с клавиатуры заказы, сделанные в течение одного дня. Для каждого заказа программа вычислит его стоимость и выведет ее на экран.

Ввод данных прекратится, когда будет введено число 0 как количество бутербродов.

Программа также подсчитает, сколько всего заказов сделано в этот день, и выведет это число на экран.

Примечание: Нет необходимости проверять корректность вводимых данных.

7.

В кабинете матери и ребенка решили проверить, сколько мальчиков и девочек из 1000 детей, записанных в кабинете, родились в первые шесть месяцев 2009 года.

А. Напишите метод, который получит два целых числа. Метод вернет, если первое число – один из проверяемых месяцев (1–6), а второе число – проверяемый год (2009).

Б. Напишите фрагмент программы, который примет с клавиатуры для каждого из 1000 детей, записанных в кабинете, месяц его рождения, год рождения и его пол (мальчик или девочка).

Фрагмент программы подсчитает и выведет на экран, сколько мальчиков и сколько девочек родилось в первые шесть месяцев 2009 года.

Вы должны использовать метод, написанный вами в пункте А.

Примечание: Нет необходимости проверять корректность вводимых данных.

8.

Перед вами фрагмент программы:

```
Scanner input=new Scanner(System.in);
int[] arr = new int[8];
int i, k;
```

```

int num;
i = 0;
k = arr.length - 1;
while (i <= k)
{
    num = in.nextInt();
    if (num % 2 == 0)
    {
        arr[i] = num;
        i++;
    }
    else
    {
        arr[k] = num;
        k--;
    }
}
System.out.println("i = " + i + " k = " + k);

```

- А. Проследите с помощью таблицы трассировки за выполнением данного фрагмента для следующего ввода: 33, 78, 12, 4, 3, 7, 6, 8. Запишите, каким будет вывод. Таблица трассировки должна включать:
- столбцы для **i**, **k**, **num**, **arr[i]**, **arr[k]**;
 - столбец, показывающий выполняется ли условие команды **if**;
 - столбец для вывода.
- Б. Приведите пример ввода, для которого после выполнения данного фрагмента значение **i** будет равно 4.
- В. Приведите пример ввода, для которого после выполнения данного фрагмента значение **i** будет равно 0.

Часть 3 (20 баллов)

Необходимо ответить на 1 вопрос из вопросов 9–10.

9.

Дан двумерный массив размером $m \times n$, содержащий целые числа.

Определим **правую диагональную тройку** массива, которая начинается в позиции $[i][j]$ следующим образом:

- первый элемент – в позиции $[i][j]$;
- второй элемент – на одну строку вниз и на один столбец вправо от первого элемента;
- третий элемент – на одну строку вниз и на один столбец вправо от второго элемента.

Например, в приведенном ниже массиве размером 5×6 обозначена **правая диагональная тройка**, которая начинается в позиции $[1][3]$.

0	3	6	7	1	0
17	1	1	2	5	1
23	8	9	6	7	1
1	1	85	9	1	12
0	43	1	31	4	1

Дан двумерный массив размером $m \times n$, содержащий только числа 0 и 1.

Определим **единичную правую диагональную тройку** массива, которая начинается в позиции $[i][j]$ следующим образом: существует **правая диагональная тройка**, начинающаяся в позиции $[i][j]$, и значения всех элементов этой тройки равны 1.

Например, в приведенном ниже массиве размером 5×6 обозначена **единичная правая диагональная тройка**, которая начинается в позиции $[1][2]$.

0	0	0	0	1	0
1	1	1	0	0	1
0	0	0	1	0	1
1	1	1	1	1	0
0	0	1	0	0	0

А. **Напишите метод, который получит:**

- двумерный массив, содержащий только числа 0 и 1;
- два целых числа, обозначающих позицию элемента массива (первое число обозначает строку, второе число обозначает столбец).

Метод проверит, существует ли **правая диагональная тройка**, начинающаяся в этой позиции. Если да, метод вернет значение **true**, иначе вернет значение **false**.

Б. **Напишите метод, который получит:**

- двумерный массив, содержащий только числа 0 и 1;
- два целых числа, обозначающих позицию элемента массива (первое число обозначает строку, второе число обозначает столбец).

Метод проверит, существует ли **единичная правая диагональная тройка**, начинающаяся в этой позиции. Если да, метод вернет значение 1, иначе вернет значение 0.

Вы должны использовать метод, написанный в пункте А.

В. **Дан двумерный массив размером 36×52, содержащий только числа 0 и 1.**

Напишите программу, которая подсчитает и выведет на экран количество единичных правых диагональных троек в данном массиве. Вы должны использовать метод, написанный в пункте Б.

Примечание: Нет необходимости вводить данные в массив.

Нет необходимости проверять корректность данных в массиве.

10.

В бассейне есть три разных аттракциона: трамплин, джакузи и сауна. Прыгать с трамплина можно, начиная с 12 лет. Пользоваться джакузи можно с 16 лет. Пользоваться сауной можно с 18 лет.

А. **Напишите метод, который получит возраст посетителя бассейна. Метод вернет одномерный массив из 3 целых чисел, каждое из которых соответствует одному из аттракционов бассейна. В массиве будет содержаться число 1 для тех аттракционов, которые разрешены данному посетителю, и число 0 для запрещенных ему аттракционов.**

Б. **Напишите программу, которая для определенного дня примет с клавиатуры возраст каждого из посетителей бассейна. Программа подсчитает число посетителей, которые имели право пользоваться каждым из аттракционов, и выведет полученные числа на экран.**

Ввод данных завершится, когда будут введены данные о посетителе, которому нельзя пользоваться ни одним из аттракционов.

Вы должны использовать метод, написанный вами в пункте А.

Примечание: Нет необходимости проверять корректность вводимых данных.