

Государственное бюджетное нетиповое образовательное учреждение

**САНКТ-ПЕТЕРБУРГСКИЙ ГОРОДСКОЙ ЦЕНТР
ДЕТСКОГО ТЕХНИЧЕСКОГО ТВОРЧЕСТВА**

РАССМОТРЕНО
на педагогическом совете
СПбГЦДТТ
Протокол № 1 от 27 августа 2020 г.

УТВЕРЖДЕНА
Приказом № 55/5 от 31.08.2020
Директор СПбГЦДТТ
А.Н. Думанский



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«АЗЫ ПРОГРАММИРОВАНИЯ»

Возрастной состав обучающихся: 8 - 12 лет

Продолжительность обучения: 2 года

Разработчик:
Румянцева Мария Юрьевна,
педагог дополнительного образования
СПбГЦДТТ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «Азы программирования» реализуется в рамках деятельности отдела компьютерных технологий.

Направленность программы: техническая.

Программирование — процесс создания компьютерных программ.

По известному выражению Никлауса Вирта «Программы = алгоритмы + структуры данных»; иными словами, ключевыми непосредственными задачами программирования являются создание и использование алгоритмов и структур данных.

В более широком смысле под программированием понимают весь спектр деятельности, связанный с созданием и поддержанием в рабочем состоянии программ — программного обеспечения. Эта инженерно-техническая дисциплина называется «программная инженерия». Сюда входят анализ и постановка задачи, проектирование программы, построение алгоритмов, разработка структур данных, написание текстов программ, отладка и тестирование программы (испытания программы), документирование, настройка (конфигурирование), доработка и сопровождение.

Программирование основывается на использовании языков программирования, на которых записываются инструкции для компьютера. Современное приложение содержит множество таких инструкций, связанных между собой.

Программирование занимает одно из первых мест среди современных профессий по требованиям к владению навыками алгоритмирования и абстрагирования, а так же базовых навыков программирования, которыми необходимо овладевать как можно раньше из-за динамичности развития области и требовательности ее к навыкам адаптации к новым технологиям.

Актуальность программы обусловлена:

- потребностью в алгоритмически грамотных специалистах,
- необходимостью алгоритмической грамотности при изучении любого языка программирования,
- необходимостью профессионального самоопределения, предоставлением возможности определиться в дальнейшем развитии,
- быстро растущими требованиями к новым поколениям программистов.

Адресат программы. Дети в возрасте 9-13 лет, интересующиеся программированием.

Уровень освоения программы: Общекультурный

Цель программы: сформировать у учащихся системы компетенций, связанных со знаниями в области алгоритмизации и программирования, общих закономерностей развития IT-области для дальнейшего их применения при освоении различных областей программирования и выполнения работ, связанных с программированием и алгоритмизацией, включая проектные. Развить индивидуальные познавательные способности каждого ребенка; максимально выявить, инициировать, использовать, «окультурить» индивидуальный (субъектный) опыт ребенка; помочь личности познать себя, самоопределиться и самореализоваться, а не формировать заранее заданные свойства.

Задачи:

Обучающие:

- Приобретение учениками базовых знаний в области алгоритмизации
- Овладение учениками основными способами представления алгоритма

Развивающие

- Развитие у учащихся логического мышления
- Развитие способностей анализа, обобщения
- Развитие аналитических способностей

Воспитательные

- Воспитание у учащихся культуры работы с программным кодом
- Воспитание у учащихся навыков командного решения задач

С целью создания условий для повышения гражданской ответственности за судьбу страны, воспитания гражданина, любящего свою Родину и семью, имеющего активную жизненную позицию, учащиеся принимают участие в мероприятиях, направленных на воспитание гражданственности и патриотизма, проводимых в СПбГЦДТТ, а также на занятиях проводятся беседы по патриотическому воспитанию.

В данной программе на занятиях предусмотрены пятиминутки для профилактики коррупционных действий с целью формирования правового сознания и антикоррупционного мировоззрения учащихся. Учащиеся принимают участие в информационно-просветительских мероприятиях СПбГЦДТТ.

Отличительной особенностью данной программы является направленность образовательного процесса на формирование способностей к алгоритмическому мышлению, крайне необходимому для дальнейшего развития учащегося в области программирования, а так же позволяющая выделить одаренных детей на раннем этапе их развития и способствующая раннему самоопределению в профессиональной области.

Также особенностью программы является абстрагированность от конкретных сред и языков программирования, что позволит сохранять актуальность программы без значительных изменений для различных педагогов.

Процесс достижения цели и задач программы осуществляется в сотрудничестве учащихся, педагога дополнительного образования и родителей.

Педагогические основы программы

Программа основана на следующих педагогических принципах:

- личностной ориентированности
- природосообразности

Личностно ориентированное обучение призвано раскрыть и использовать субъектный опыт ученика, помочь становлению личностно значимых способов познания путем организации целостной учебной (познавательной) деятельности.

Природосообразность педагогический принцип, согласно которому педагог в своей деятельности руководствуется факторами естественного, природного развития ребенка, согласования организации всего педагогического процесса не только с возможностями самого ребенка на определенных этапах его развития, но и с природой, в которой он живет, ее изменениями.

Основой программы является:

- самостоятельность в составлении и решении практических задач,
- запас технических знаний,
- богатство понятий, способность комбинировать, рассуждать,
- наличие технических интересов.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ:

Реализация дополнительной общеобразовательной общеразвивающей программы осуществляется в очном режиме или с использованием дистанционных технологий и электронных образовательных ресурсов.

Данная программа рассчитана на детей 8-13 лет.

Продолжительность обучения – 2 года

Занятия проводятся 1 раз в неделю по 2 часа.

–Первый год – 72 часа и 8 часов в летний период.

–Второй год – 72 часа и 8 часов в летний период.

Программа нацелена на обучение основам алгоритмического мышления и способов разработки алгоритмов. Процесс обучения строится с учётом знаний, умений и навыков, приобретаемых школьниками на уроках в соответствии с обязательным образовательным минимумом.

В детское объединение «Азы программирования», на первый и второй год обучения, принимаются все желающие, без предварительного отбора.

Техническое оснащение занятий:

Компьютерный класс, оборудованный компьютерами типа IBM-PC.

Интерактивная доска.

Программное обеспечение:

–Среда визуального программирования «Scratch»

–Интерпретатор языка программирования Python.

Программное обеспечение процесса обучения

Скретч — визуальная объектно-ориентированная среда программирования для обучения школьников младших и средних классов. Название произошло от слова *scratching* — техники, используемой хип-хоп-диджеями, которые крутят виниловые пластинки взад-вперёд руками для того, чтобы смешивать музыкальные темы.

Скретч создан как продолжение идей языка Лого и конструктора Лего. Скретч 1 был написан на языке Squeak, Скретч 2 ориентирован на работу онлайн и переписан на Flash/ActiveScript. Скретч разрабатывается небольшой командой программистов для детей в Массачусетском технологическом институте. Т

Программы на Скретче состоят из графических блоков, подписи к которым зависят от выбранного для интерфейса языка. Может быть выбран один из 50 языков интерфейса, включая русский. Для подключения интерфейса на новом языке используются стандартные gettext файлы.

В данной программе используется технология интерактивного обучения на основе использования новейших технических средств, состоящая из следующих этапов:

Первый год обучения:

- Изучение основных элементарных алгоритмов на понятийном уровне;
- Анализ алгоритмов, встречающихся в повседневности;
- Оптимизация примеров «повседневных» алгоритмов;
- Ввод понятия подпрограмма (функция)
- Создание алгоритмов в программной системе для изучения программирования на начальном уровне;
- Анализ созданных алгоритмов;

- Изучение принципов структурного программирования
- Изучения способа представления алгоритма «Блок-схема»
- Реализация алгоритмов с использованием блок-схем

Второй год обучения:

- Изучение способа представления алгоритма «Псевдокод»
- Повторение элементарных алгоритмов и их представления в псевдокоде
- Написание программ на базе языке программирования
- Изучение теоретических основ программирования

Основными принципами используемой системы обучения являются:

- Принцип обратной связи. Используется как педагогом, так и учащимся. Одному она нужна для понимания учебного материала, другому для коррекции приоритетов, с учетом конкретных способностей и возможностей учащихся.
- Разделение материала программы на отдельные самостоятельные, но взаимосвязанные порции информации и учебные задания. Последовательность шаговых учебных процедур образует обучающую программу - основу технологии программированного обучения.
- Индивидуальность работы учащихся по программе. Каждый учащийся имеет возможность продвигаться в учении с наиболее благоприятной для его познавательных сил скоростью.
- Принцип использования специальных технических средств для подачи программируемых учебных материалов подразумевает использование компьютера для подачи некоторых частей программы и контроля их усвоения.

Программа логически разделена на 4 блока продолжительностью по полгода.

В первом блоке программа больше нацелена на развитие навыка разработки алгоритма с помощью выполнения заданий в программной системе для изучения базовых навыков программирования.

Во втором блоке программы рассматриваются более сложные алгоритмы, изучаются формальные способы представления алгоритмов.

В третьем и четвертом блоке работа ведется на базе языка программирования высокого уровня, и рассматриваются различные, постепенно усложняющиеся, алгоритмы, так же изучаются некоторые элементарные алгоритмы, имеющих особое значение в программировании, также особое внимание уделяется «красоте» написания кода.

В четвертом блоке программы уделяется особое внимание оптимизации алгоритмов.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ.

К концу обучения учащиеся должны овладеть следующими ключевыми компетенциями:

Личностные

- Культура работы с программным кодом
- Приобретение навыков командного решения задач

Предметные

- Знание основных терминов алгоритмизации и программирования
- Знание принципов процедурного программирования
- знание и понимание цели каждого из этапов создания программного продукта
- Знание основных принципов написания «Красивого» программного кода
- Знание основных принципов написания «Красивого» программного кода

- Знание этапов создания программного продукта
- Умение создать алгоритм любой сложности, основанный на принципах процедурного программирования
- Умение конкретизировать и абстрагировать алгоритм
- Умение работать в среде программирования «ПиктоМир», «КуМир»
- Умение использовать ПК для создания программных продуктов
- Умение представить алгоритм в виде псевкода
- Умение составить грамотное техническое задание к программному продукту
- Умение работать в команде
- Умение принимать социальные и этические обязательства

Метапредметные

- Развитие у учащихся логического мышления
- Развитие способностей анализа, обобщения
- Развитие аналитических способностей

Формы контроля

Контроль уровня усвоения знаний проводится по мере прохождения ключевых тем и разделов программы и включает в себя выполнения различного вида практических работ. Кроме этого в течении каждого учебного года проводится вводный (контроль начального уровня подготовки детей), текущий (контроль уровня знаний проводящийся в середине каждого года обучения) и итоговый контроль (контроль степени усвоения программы определенного года обучения в целом).

УЧЕБНЫЙ ПЛАН ПЕРВОГО ГОДА ОБУЧЕНИЯ

№	Наименование раздела, темы	В том числе			Формы контроля
		Всего	Теория	Практика	
1.	Вводное занятие. Задачи обучения. Техника безопасности. История информационных технологий.	2	1	1	Практическое задание.
	Понятие алгоритма.	2	1	1	Опрос
3.	Знакомство программной системой изучения программирования.	4	2	2	Тест
4.	Изучение линейных алгоритмов	12	6	6	Практическое задание
5.	Изучение условных алгоритмов	14	7	7	Практическое задание
6.	Изучение циклических алгоритмов	14	7	7	Практическое задание

7.	Изучение функций и подпрограмм	14	7	7	Практическое задание
8.	Блок-схемы.	8	4	4	Практическое задание
9.	Итоговое занятие.	2	-	2	-
10.	Повторение и закрепление пройденного материала	8	2	6	
	Итого	80			

УЧЕБНЫЙ ПЛАН ВТОРОГО ГОДА ОБУЧЕНИЯ

	Наименование раздела, темы	В том числе			Формы контроля
		Всего	Теория	Практика	
1.	Вводное занятие. Задачи обучения. Техника безопасности. История языков программирования.	2	1	1	-
2.	Изучение среды и языка программирования	20	10	10	Тест
3.	Структурное программирование: основы.	8	4	4	Практическое задание
4.	«Красивый» код.	6	3	3	Опрос
5.	Алгоритмы.	20	10	10	Практическое задание
6.	Разработка собственного проекта.	14	7	7	-
7.	Итоговое занятие.	2	-	2	-
8.	Повторение и закрепление пройденного материала	8	2	6	
	Итого	80			

Календарный учебный график

Год обучения	Дата начала обучения по программе	Дата окончания обучения по программе	Всего учебных недель	Количество учебных часов	Режим занятий
1 год	1 сентября	По мере выполнения программы 1 года обучения	40	80 часов по учебному расписанию	1 раз в неделю по 2 часа
2 год	1 сентября	По мере выполнения программы 2 года обучения	40	80 часов по учебному расписанию	1 раз в неделю по 2 часа

УТВЕРЖДАЮ

Директор СПбГЦДТТ

_____ А.Н. Думанский

РАБОЧАЯ ПРОГРАММА
К ДОПОЛНИТЕЛЬНОЙ ОБЩОБРАЗОВАТЕЛЬНОЙ
ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ

«Азы программирования»

202__ – 202__ учебный год

Год обучения первый

Группа № _____

Румянцева Мария Юрьевна
педагог дополнительного образования
СПбГЦДТТ

РАБОЧАЯ ПРОГРАММА ПЕРВОГО ГОДА ОБУЧЕНИЯ

Рабочая программа 1-го года обучения составлена на основе дополнительной общеобразовательной общеразвивающей программы «Азы программирования» технической направленности.

Цель программы: Сформировать у учащихся системы компетенций, связанных со знаниями в области алгоритмизации, общих закономерностей развития IT-области для дальнейшего их применения при освоении различных областей программирования и выполнения работ, связанных с программированием и алгоритмизацией.

Задачи 1 года обучения:

Обучающие

- Изучение основных элементарных алгоритмов на понятийном уровне;
- Анализ алгоритмов, встречающихся в повседневности;
- Оптимизация примеров «повседневных» алгоритмов;
- Ввод понятия подпрограмма (функция)
- Создание алгоритмов в среде «Пиктомир»;
- Анализ созданных алгоритмов;
- Изучение принципов структурного программирования;
- Изучения способа представления алгоритма «Блок-схема»;
- Реализация алгоритмов с использованием блок-схем.
- Приобретение учениками базовых знаний в области алгоритмизации;
- Владение учениками основными способами представления алгоритма.

Развивающие

- Развитие у учащихся логического мышления;
- Развитие способностей анализа, обобщения;
- Развитие аналитических способностей.

Воспитательные

- Воспитание у учащихся культуры работы с программным кодом;
- Воспитание у учащихся навыков командного решения задач.

Условия организации учебно-воспитательного процесса:

Отличительной особенностью данной программы является направленность образовательного процесса на формирование способностей к алгоритмическому мышлению, крайне необходимому для дальнейшего развития учащегося в области программирования.

Программа нацелена на обучение основам алгоритмического мышления и способов разработки алгоритмов. Процесс обучения строится с учётом знаний, умений и навыков, приобретаемых школьниками на уроках в соответствии с обязательным образовательным минимумом.

В данной программе используется технология интерактивного обучения на основе использования новейших технических средств.

Оборудованы 9 рабочих мест, оснащенных персональными компьютерами с программным обеспечением, необходимым для реализации данной программы.

Контроль знаний проводится по результатам итогового контроля и анализа данных.

Вводный контроль. Проводится в форме устного собеседования, направленного на выявление способностей к алгоритмическому и логическому мышлению.

Текущий и итоговый контроль состоит из теоретической и практической части.

Теоретическая:

- знание определений, связанных с алгоритмизацией,
- контроль понимания назначений и типов алгоритмов.

Практическая:

- решение задачи в среде «ПиктоМир»,
- написание простого алгоритма с помощью псевдокода.

Требования к полученным знаниям и умениям учащихся 1 года обучения:

должны ЗНАТЬ:

- Знание основных терминов алгоритмизации и программирования;
- Знание принципов процедурного программирования.

должны УМЕТЬ:

- Умение конкретизировать алгоритм;
- Умение абстрагировать алгоритм;
- Умение использовать ПК для построения алгоритма;
- Умение работать в среде «Scratch»

СОДЕРЖАНИЕ ПРОГРАММЫ ПЕРВОГО ГОДА ОБУЧЕНИЯ

1. Вводное занятие.

Теория: вводный инструктаж. Определение понятия программист, определение сферы деятельности программиста. Свободное знакомство со средой Scratch.

Практика: знакомство с учениками, вводный контроль.

2. Понятие алгоритма.

Теория: алгоритм, определение и основные свойства

Практика: анализируются различные алгоритмы из повседневной жизни, желательно, предложенные самими учениками. (Для примера можно привести алгоритм заваривания чая или рецепт приготовления пирога). Конкретные примеры зависят от интересов учащихся.

3. Знакомство со средой программирования.

Теория: информация по интерфейсу программной системы изучения программирования, введение основных понятий, связанных с алгоритмами: цикл, функция, условие. Виды алгоритмов.

Практика: изучение интерфейса, создание простейших алгоритмов в программной системе.

4. Изучение линейных алгоритмов.

Теория: Изучение способа представления алгоритма в системе изучения программирования, создание простейших по структуре линейных алгоритмов

Практика: реализация серии задач с решением в виде линейного алгоритма в программной системе обучения программированию.

5. Изучение условных алгоритмов

Теория: повторение определения «Условие», где они встречаются в жизни, в чем их необходимость использовании в программах.

Практика: реализация серии задач с решением в виде разветвленного алгоритма в программной системе обучения программированию.

6. Изучение циклических алгоритмов.

Теория: повторение определения «Цикл», где они встречаются в жизни, в чем их преимущество при использовании в программах. Обучение поиску неявных циклов в задачах, поиск неявных функций в задачах.

Практика: реализация серии задач с решением в виде циклического алгоритма в программной системе обучения программированию.

7. Дополнительные возможности Scratch.

Теория: Изучение различных дополнительных возможностей среды программирования Scratch.

Практика: реализация серии задач с использованием дополнительных возможностей.

8. Введение в Python

Теория: Сравнение Python и Scratch. Основы синтаксиса Python.

Практика: разбор программ на языке Python и выполнение простейших задач.

9. Итоговое занятие.

Практика: проведение итогового контроля.

Календарно-тематическое планирование 1 года обучения

№	Кол-во часов	Дата по плану	Раздел	Тема занятия	Методическое обеспечение занятия
1	2		Вводное раздел		Инструкции по требованиям безопасности (Приложение)
				Знакомство с обучающимися. Обсуждение понятия программирования, желаний обучающихся. Заполнение «дерева ожиданий». <i>Практика:</i> Просмотр видео о Scratch. Свободное знакомство со средой программирования.	Текущая образовательная программа. Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.10
2	2		Понятие алгоритма Вводный контроль.	Игра «Программирование танца». Знакомство с алгоритмами. Создание алгоритмов из жизни. <i>Практика:</i> Вводный контроль. Задание «Обо мне»	ru.wikipedia.org/wiki/Алгоритм Задание вводного контроля. Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.26
3	2		Знакомство с программной системой изучения программирования.	Знакомство с координатной плоскостью. Команды движения Scratch. <i>Практика:</i> Создание игры «Убеги от дракона»	Программа «Scratch». Руководство пользователя.

					Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, Дэниел Маккаферти Спрайты– с 28
4	2			Повторение команд движения. Простейшие сенсоры. <i>Практика:</i> Создание игры «Лабиринт»	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.77
5	2		Базовые возможности Scratch.	Знакомство с событиями. Управление спрайтами с клавиатуры. <i>Практика:</i> Создание игры «Пинг-понг»	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.78
6	2		Простые циклы	Простые циклы – повторить, всегда. Клоны. <i>Практика:</i> Создание игры «Вертолёт»	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.79
7	2		Переменные	Знакомство с переменными. Счёт. <i>Практика:</i> Создание игры «Голодная рыба»	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.80

8	2			Повторение изученного. Практика: Творческое задание 10 негритят.	Презентация, инструкция по разработке программы «10 негритят »
9	2			Работа с переменными. <i>Практика:</i> Задание:	Презентация, инструкция по разработке программы «Арифметический тренажёр»
10	2			Работа с переменными. Работа с пользователем. Практика: игра «Викторина»	Презентация, инструкция по разработке программы «Викторина»
11	2			Переменные. Списки.	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.83
			Расширения и повторения.	Уровни. Мышь. Перезагрузка. Меню. Мультиплеер. Практика: Усложнение игры «Пинг-понг»	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.83

12	2			Выполнение заданий в рамках всероссийской акции “Час кода”	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.84
13	2			Уровни. Мышь. Перезагрузка. Меню. Мультиплеер. Практика: Усложнение игры «Пинг-понг»	http://www.coderussia.ru/
14	2			<i>Практика:</i> Новогодняя открытка в Scratch.	Креативное программирование Карен Бреннан, Кристиан Болкх, Мишель Чунг, Гарвардская Высшая школа образования - С.84
15	2			<i>Практика:</i> Новогодняя открытка в Scratch.	
16	2			Подведение итогов первого полугодия. Повторение изученного.	
17	2		Продвинутые возможности Scratch.	Инструктаж по Технике безопасности. Знакомство с пером. Работа с пером. <i>Практика:</i> Проект «Paint»	Инструкции по требованиям безопасности (Приложение) Презентация по разработке проекта «Paint»

18	2			<i>Практика:</i> Завершение работы над проектом Paint.	Презентация по разработке проекта «Paint»
19	2			Работа с операторами. <i>Практика:</i> «Арифметический тренажёр»	Презентация по разработке проекта «Арифметический тренажёр»
20	2			Калькулятор на Scratch.	https://www.youtube.com/watch?v=lbu4RBk3zoM&list=PUSBeL28cCqIyHFxmCTK1Ejw&index=37
21	2			Задачи со строками и вводом. <i>Практика:</i> игра «Ерунда»	
22	2			Работа с захватом видео.	
23	2			Работа с захватом видео.	
24	2			Работа с захватом видео.	
25	2		Разработка игр	<i>Практика:</i> Змейка	
				<i>Практика:</i> Распан	
26	2			<i>Практика:</i> 2048	

27	2			<i>Практика:</i> Растения против зомби	
28	2		Хакатон	Подготовка к мини-хакатону.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 60
29	2			Мини-хакатон.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 60
30	2		Знакомство с Python.	Сравнение языков Python и Scratch. Особенности Python. Знакомство со средой CodeCombat.	Codecombat.ru
31	2			Выполнение заданий в среде CodeCombat.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 55
32	2			Выполнение заданий в среде CodeCombat.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 87

33	2			Выполнение заданий в среде CodeCombat.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 97
34	2			Выполнение заданий в среде CodeCombat.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 102
				Выполнение заданий в среде CodeCombat.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 102
35	2			Итоговый контроль.	Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, ДэниелМаккаферти Решение и ветвление – с 110
36	2		Итоговое занятие	Итоговое занятие. Подведение итогов года. Рефлексия изученного за год. Постановка задач второго года обучения.	

37	2		Хакатон	Инструктаж по Технике безопасности. Знакомство с форматом хакатона. Составление команд. Выбор проекта для выполнения. Постановка задач для выполнения.	Инструкции по требованиям безопасности (Приложение) Программирование для детей, Кэрол Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, Дэниел Маккаферти
38	2			Выполнение проекта в командах.	
39	2			Выполнение проекта в командах.	
40	2			Окончание работы над проектами. Защиты проектов.	
Итого:			80		

УТВЕРЖДАЮ

Директор СПбГЦДТТ

_____ А.Н. Думанский

РАБОЧАЯ ПРОГРАММА
К ДОПОЛНИТЕЛЬНОЙ ОБЩОБРАЗОВАТЕЛЬНОЙ
ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЕ

«Азы программирования»

202__ – 202__ учебный год

Год обучения второй

Группа № _____

Румянцева Мария Юрьевна,
педагог дополнительного образования
СПбГЦДТТ

Рабочая программа 2-го года обучения составлена на основе дополнительной общеобразовательной общеразвивающей программы «Азы программирования» технической направленности.

Цель программы: Сформировать у учащихся системы компетенций, связанных со знаниями в области алгоритмизации, общих закономерностей развития IT-области для дальнейшего их применения при освоении различных областей программирования и выполнения работ, связанных с программированием и алгоритмизацией.

Задачи 2 года обучения:

Обучающие

- Изучение способа представления алгоритма «Псевдокод»;
- Повторение элементарных алгоритмов и их представления в псевдокоде;
- Написание программ на базе языка программирования Java;
- Изучение теоретических основ программирования.
- Приобретение учениками базовых знаний в области алгоритмизации;
- Владение учениками основными способами представления алгоритма.

Развивающие

- Развитие у учащихся логического мышления;
- Развитие способностей анализа, обобщения;
- Развитие аналитических способностей.

Воспитательные

- Воспитание у учащихся культуры работы с программным кодом;
- Воспитание у учащихся навыков командного решения задач.

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Данная программа рассчитана на детей 9-13 лет.

Занятия проводятся 1 раз в неделю по 2 часа.

Общее количество часов первого года обучения – 72 часа и 8 часов в летний период.

В детское объединение «Азы программирования» принимаются все желающие, без предварительного отбора.

Условия организации учебно-воспитательного процесса:

Отличительной особенностью данной программы является направленность образовательного процесса на формирование способностей к алгоритмическому мышлению, крайне необходимому для дальнейшего развития учащегося в области программирования.

Программа нацелена на обучение основам алгоритмического мышления и способов разработки алгоритмов. Процесс обучения строится с учётом знаний, умений и навыков, приобретаемых школьниками на уроках в соответствии с обязательным образовательным минимумом.

В данной программе используется технология интерактивного обучения на основе использования новейших технических средств.

Оборудованные 8 рабочих мест, оснащенных персональными компьютерами с программным обеспечением, необходимым для реализации данной программы.

Контроль знаний проводится по результатам итогового контроля и анализа данных.

Вводный контроль. полностью повторяет итоговый контроль первого года обучения.

Итоговый контроль состоит из теоретической и практической части.

Теоретическая часть направлена на контроль теоретических знаний по всему курсу обучения.

Практическая:

- решение задачи с помощью Блок-схем;
- написание программы на языке Python.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

**Требования к полученным знаниям и умениям учащихся 2 года обучения:
должны ЗНАТЬ:**

- основные термины алгоритмизации и программирования;
- знать и понимать цели каждого из этапов создания программного продукта;
- основные принципы написания «Красивого» программного кода;
- этапы создания программного продукта.

должны УМЕТЬ:

- создать алгоритм любой сложности, основанный на принципах процедурного программирования;
- работать в среде программирования «Eclipse»;
- использовать ПК для создания программных продуктов;
- представить алгоритм в виде псевдокода;
- составить грамотное техническое задание к программному продукту;
- работать в команде;
- принимать социальные и этические обязательства.

СОДЕРЖАНИЕ ПРОГРАММЫ ВТОРОГО ГОДА ОБУЧЕНИЯ

1. Вводное занятие.

Теория: вводный инструктаж.

Практика: знакомство с учениками, вводный контроль.

2. Изучение среды и языка программирования.

Теория: знакомство с интерфейсом среды программирования. Изучение основ синтаксиса языка программирования. Основные приемы при работе с незнакомой средой программирования и языком программирования.

Практика: знакомство с интерфейсом, знакомство с документацией. Анализ простейшего примера программы на языке программирования под руководством педагога, запуск этой программы учениками.

3. Структурное программирование основы.

Теория: определение, основные принципы, основные конструкции: последовательное исполнение, ветвление, цикл, пошаговость разработки. Подробный анализ основных свойств и методов подхода.

Практика: анализ нескольких алгоритмов (написанных учениками с помощью блок-схем ранее или новых, написанный в КуМир'е) с точки зрения структурного программирования. Решение задач, заостряя внимание на структурном подходе.

4. «Красивый» код.

Теория: основные правила написания красивого и хорошо читаемого кода, правила оформления, именования объектов.

Практика: анализ одного и того же алгоритма в двух вариантах написания: с соблюдением правил оформления кода и с грубыми нарушениями, заострение внимания на удобстве повторного использования и редактирования кода. Задачи по исправлению алгоритмических ошибок в «красивом» и «некрасивом» коде.

5. Алгоритмы.

Теория: алгоритмическое введение в задачу, повторение математических основ, используемых в задаче.

Практика: написание программ в соответствии с заданием.

6. Разработка собственного проекта.

Теория: вводная информация по возможным знаниям на проект, разработка идеи проекта, возможен выбор из предложенных педагогом тем. Возможность сопровождения ПП, этапы разработки, планирование. Основы презентации программного продукта.

Практика: планирование архитектуры проекта, написание части алгоритмов. Написание основной части кода проекта. Завершение разработки, анализ созданного ПП, подготовка доклада к защите. Презентация ПП перед другими учащимися объединения

7. Итоговое занятие.

Практика: итоговый контроль.

Календарно-тематическое планирование 2 года обучения

№ занятия	Кол-во часов	Дата по факту	Раздел	Тема занятия	Методическое обеспечение занятия
1	2		Вводное занятие. Техника безопасности. История языков программирования.	Инструктаж по Технике безопасности. Свободное выполнение задания (свободная тема) на Scratch. Вводный контроль.	Инструкции по требованиям безопасности (Приложение). Текущая образовательная программа, задания вводного контроля.
2	2		Изучение среды и языка программирования	Знакомство с IDLE. Ввод-вывод. <i>Практика:</i> Вывод узора.	Текст программы «Ерунда»
3	2			Практика ввода-вывода. <i>Практика:</i> Игра «Ерунда»	Текст программы «Ерунда»
4	2			Практика ввода-вывода. <i>Практика:</i> Решение задач.	http://pythontutor.ru/
5	2			Простые условия и циклы. <i>Практика:</i> Игра «Почемучка»	Текст программы «Почемучка»

6	2			Изучение и модификация простого примера программы Практика:	Pythontutor.ru
7	2			Повторение синтаксиса использования условий. Практика: программа Paint Составление программы с использованием условий Практика: «Крестики-нолики»	Текст программы Paint Задача "Максимально допустимый рейтинг фильма"
8	2			Изучение синтаксиса использования циклов for. Изучение и модификация простого примера программы с использованием цикла for.	Пример программы с использованием цикла for. Пример программы с использованием цикла for.
9	2			Изучение синтаксиса использования циклов while Изучение и модификация простого примера программы с использованием цикла while.	Пример программы с использованием цикла while. Пример программы с использованием цикла while.
10	2			Изучение синтаксиса использования циклов do {} while. Изучение и модификация простого примера программы с использованием цикла do {} while.	Пример программы с использованием цикла do {} while. Пример программы с использованием цикла do {} while.

11	2			<p>Повторение синтаксиса использования циклов. Составление простого примера программы с использованием циклов.</p>	<p>Пример программы с использованием цикла for, Пример программы с использованием цикла while, Пример программы с использованием цикла do{ }while. Задача "Вывод заданной последовательности символов".</p>
12	2		<p>Структурное программирование. Основы.</p>	<p>Определение, основные принципы. Текущий контроль.</p>	<p>Джозеф Джарратано, Гари Райли Глава 10. Процедурное программирование // «Экспертные системы: принципы разработки и программирование» : Пер. с англ. — М. : 2006. — 779—851 стр., «Вильямс» Филд А., Харрисон П. Функциональное программирование = FunctionalProgramming. — М.: Мир, 1993. — 637 с. — ISBN 5-03-001870-0.</p>

13	2			Анализ алгоритмов из повседневной жизни, основываясь на принципах структурного программирования.	Н. А. Роганова Функциональное программирование: Учебное пособие для студентов высших учебных заведений — М.: ГИНФО, 2002. — 260 с. Стр. 14 п. 3.1. Ленивые и энергичные вычисления Примеры алгоритмов, самостоятельная работа учеников.
14	2			Подробный анализ принципа универсальности.	Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms. — 3-е изд. — М.: Вильямс, 2006. — С. 720.- ISBN 0-201-89683-4 Колмогоров А. Н. Теория информации и теория алгоритмов. — М.: Наука, 1987. — 304 с. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман Глава 8. Введение в теорию машин Тьюринга // Введение в теорию автоматов, языков и вычислений = Introduction to Automata Theory, Languages, and Computation. — М.: «Вильямс», 2002. — С. 528. — ISBN 0-201-44124-1

15	2			Написание нескольких алгоритмов	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 4.6, 4.36, 4.41, 4.47.
16	2		«Красивый» код	«Зачем и Почему»	http://www.slideshare.net/ptsukanov/solid-12138776 http://usarov.net/842-kak-pisat-krasivyy-kod
17	2			Основные правила	http://ru.wikipedia.org/wiki/Стандарт_оформления_кода Д. Ван Тассел Стиль, разработка, эффективность, отладка и испытание программ = Programstyle, design, efficiency, debugging, and testing. — 2-е изд. — М.: Мир, 1985. -332 с.
18	2			Инструктаж по Технике безопасности. Сравнение примеров «Красивого» и «Некрасивого» программного кода.	Инструкции по требованиям безопасности (Приложение) http://www.webcitation.org/61FnsVgXy Код, написанный учениками на занятии
19	2		Алгоритмы	Решение задач с простыми линейными алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 2.1, 2.2.

20	2			Решение задач с простыми линейными алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 2.6, 2.9.
21	2			Решение задач с разветвленными алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 4.6, 4.36, 4.41, 4.47.
22	2			Решение задач с разветвленными алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 4.68, 4.71.
23	2			Решение задач с циклическими алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 5.1, 5.2, 5.7.
24	2			Решение задач с циклическими алгоритмами	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 5.13, 5.14, 5.23.
25	2			Понятие массива, задачи	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 11.1, 11.7.

26	2			Понятие массива, задачи	Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил. Задание 11.14, 11.19
27	2			Простейший алгоритм сортировки строк	Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms. — 3-е изд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4 http://rain.ifmo.ru/cat/view.php/theory/school
28	2			Простейший алгоритм сортировки чисел	Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms. — 3-е изд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4 http://rain.ifmo.ru/cat/view.php/theory/school
29	2		Разработка собственного проекта	Вводное занятие к проекту	Златопольский Д.М. — Интеллектуальные игры в информатике. 0151 СПб.: БХВ-Петербург, 2004ю — 400с. с.: ил.
30	2			Принципы разработки программного продукта	http://www.kv.by/index2008354201.htm http://dou.ua/lenta/articles/scrum-for-developers/

31	2			Разработка архитектуры проекта	http://ru.wikipedia.org/wiki/Архитектура_программного_обеспечения Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. FundamentalAlgorithms. — 3-еизд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4
32	2			Написание проекта	Руководство пользователя программы «КуМир» Д. Ван Тассел Стил, разработка, эффективность, отладка и испытание программ = Programstyle, design, efficiency, debugging, andtesting. — 2-е изд. — М.: Мир, 1985. — 332 с.
33	2			Написание проекта	Дональд Кнут Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. FundamentalAlgorithms. — 3-еизд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4 ФилдА., ХаррисонП. Функциональное программирование = FunctionalProgramming. — М.: Мир, 1993. — 637 с. — ISBN 5-03-001870-0
34	2			Подготовка к защите проекта Итоговый контроль	http://habrahabr.ru/company/piter/blog/171545/ http://ru.wikipedia.org/wiki/Автореферат_диссертации . Задания итогового контроля.

35	2			Защита проекта	Оценка проектов учащихся.
36	2		Итоговое занятие	Подведение итогов года.	
37	2		Хакатон	Инструктаж по Технике безопасности. Знакомство с форматом хакатона. Составление команд. Выбор проекта для выполнения. Постановка задач для выполнения.	Инструкции по требованиям безопасности (Приложение) Программирование для детей, Кэрл Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, Дэниел Маккаферти
38	2			Выполнение проекта в командах.	
39	2			Выполнение проекта в командах.	
40	2			Окончание работы над проектами. Защиты проектов.	
Итого			80		

ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Требования к уровню освоения дополнительной общеобразовательной общеразвивающей программы «Азы программирования»

Уровень освоения программы	Показатели		Целеполагание	Результат освоения уровня (показатели результативности) Требования к результату
	Срок реализации	Максимальный объем программы (в год)		
Базовый	2 года	80	Создание условий для личностного самоопределения и самореализации обеспечение процесса социализации и адаптации к жизни в обществе; выявление и поддержка детей, проявивших выдающиеся способности; развитие у учащихся мотивации к творческой деятельности интереса к научной и научно-исследовательской деятельности.	Освоение программы. Презентация результатов на уровне района, города. Участие учащихся в районных, городских и Всероссийских мероприятиях; наличие призеров и победителей в районных, городских, Всероссийских соревнованиях.

ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

Контроль знаний проводится по результатам итогового контроля и анализа созданных программ.

Первый год обучения:

- Вводный контроль. Проводится в форме устного собеседования, направленного на выявление способностей к алгоритмическому и логическому мышлению.
- Итоговый контроль состоит из теоретической и практической части.

Теоретическая:

- знание определений, связанных с алгоритмизацией,
- контроль понимания назначений и типов алгоритмов.

Практическая:

- решение задачи в программной среде, используемой в обучении
- написание простого алгоритма с помощью псевдокода.

Второй год обучения:

- Вводный контроль оценивает уровень остаточных знаний языка Scratch и способность к языку Python.
- Итоговый контроль состоит из теоретической и практической части.

Теоретическая часть направлена на контроль теоретических знаний по всему курсу обучения.

Практическая:

- решение задачи с помощью Блок-схем,
- написание программы на изучаемом языке программирования.

Пример занятия. Вводный контроль

ПДО: Румянцева Мария Юрьевна

Дополнительная общеобразовательная программа: «Азы программирования»

Год обучения: первый

Цель: выявление начального уровня знаний обучающихся и их способностей к обучению программированию.

Разделы:

Введение

Форма контроля: 2 практических задания

Содержание задания:

- 1) практическое задание в среде визуального программирования Scratch, для выполнения которого нужно было найти ошибки в готовых программах и исправить их. Таких было пять алгоритмов. Данное задание нацелено на выявление способности к анализу алгоритмов и выявление ошибок. Максимальная оценка за задание – 5 баллов.
- 2) Составление своей несложной программы из готовых блоков. Данное задание нацелено на оценку способности собственноручно создавать программы и анализировать их. Максимальная оценка за задание – 5 баллов.

Подробный текст задания см. в приложении 1.

Критерии оценки:

Номер задания	Критерий оценки
1	5 баллов – полностью выполнил задания без или с незначительной подсказкой педагога. 4 – нашёл ошибки в 3 или 4 программах из 5 или требовалось более одной подсказки педагога 3 – нашёл ошибки в 2-3 программах, требовались частые подсказки педагога. 2 – нашёл ошибки в 1-2 программах с помощью педагога, но пытался делать самостоятельно. 1 – нашёл ошибки в 1 программе из 5, после длительного нахождения педагога рядом. 0 – не приступил к заданию или не смог выполнить его в принципе.

2	<p>5 баллов – полностью и самостоятельно выполнил задание без помощи педагога.</p> <p>4 балла – полностью выполнил задания, но была необходима помощь педагога, либо были допущены ошибки при выполнении.</p> <p>3 балла – задание выполнено частично, понимание работы со средой программирования недостаточное</p> <p>2 балла – задание частично выполнено с помощью педагога</p> <p>1 балл – задание начато, но не выполнено</p> <p>0 баллов – не приступил к заданию или не смог его выполнить.</p>
---	---

Максимально в сумме можно набрать 10 баллов за два задания.

Уровень освоения материала			
	низкий	средний	высокий
Кол-во баллов	0—4	5—8	9—10

Приложение 1:

Задание 1:

10 блоков

Что вы можете создать используя только 10 блоков Скретч?

Создайте проект используя только эти 10 блоков Скретч. Применяйте их один, два или более раз, но используйте каждый блок хотя бы раз.

Для начала

- Апробируйте идеи экспериментируя с каждым блоком.
- Сочетайте блоки разными способами.
- Повторите!

Застрали?

Всё в порядке!
Попробуйте следующее:

- Апробируйте идеи экспериментируя с различными сочетаниями блоков. Перемешивайте и сочетайте блоки до тех пор пока вы не найдете то, что заинтересовало вас!
- Попробуйте мозговой штурм идей с соседом!
- Исследуйте другие проекты, чтобы увидеть, что другие делают в Скретче. Это может быть отличным способом, чтобы найти вдохновение!

Завершили?

-
-
-
-
-
-
-
-
-
-

- + Добавьте свой проект в студию «10 блоков»: <http://scratch.mit.edu/studios/475480>
- + Играйте с добавлением различных спрайтов, костюмов или фонов.
- + Творите больше! Посмотрите как много различных проектов вы можете создать, используя только 10 блоков.
- + Обменяйтесь проектами со своими партнерами и создайте новую версию.

Задание 2:

Отладка

Помогите! Вы сможете отладить эти пять программ в Скретч?

В этом задании вы научитесь разбираться в том, что пошло не так и найдете решения каждой из пяти проблем «Отладки».

Для начала

- Перейдите в студию занятия 1 «Отладка»: <http://scratch.mit.edu/studios/475483>
- Протестируйте и отладьте каждую из 5 задач в студии.
- Запишите свое решение или сделайте программу ремикс со своим решением.

Застрали?

Всё в порядке!
Попробуйте следующее:

- Создайте список возможных ошибок в программе.
- Делайте отчет о проделанной работе! Это может быть полезным напоминанием того, что вы уже испробовали и направит вас к тому, что следует попробовать дальше.
- Делитесь и сравнивайте свои ошибки и пути их решения с соседом, пока не найдете то, что вам нужно!

Отладка № 1.1 <http://scratch.mit.edu/projects/10437040>
Когда нажат зеленый флажок, оба Гобо и Скретч Кот должны начать танцевать. Но танцует только Скретч Кот! Как исправить программу?

Отладка № 1.2 <http://scratch.mit.edu/projects/10437249>
В этом проекте, когда нажат зеленый флажок Скретч кот должен начать движение с левой стороны сцены, сказать что-то, пока он там находится, а затем плыть в правую сторону сцены и сказать что-то на правой стороне. Программа работает, когда вы нажимаете первый раз на зеленый флажок, а снова нет. Как исправить программу?

Отладка № 1.3 <http://scratch.mit.edu/projects/10437366>
Скретч кот должен сделать сальто, когда нажата клавиша пробел. Но когда нажат пробел, ничего не происходит! Как исправить ошибку?

Отладка № 1.4 <http://scratch.mit.edu/projects/10437439>
В этом проекте Скретч кот должен двигаться по сцене вперед и назад, когда на него щелкнут. Но Скретч кот переворачивается и идет вниз головой! Как исправить ошибку?

Отладка № 1.5 <http://scratch.mit.edu/projects/10437476>
В этом проекте, когда нажат зеленый флажок Скретч кот должен говорить (звук) и подумать «Мяу, мяу, мяу!». Но звук запаздывает и Скретч кот говорит «Мяу» один раз. Как исправить ошибку?

Завершили?

- + Обсудите с партнером полученный опыт тестирования и отладки. Найдите сходства и различия в ваших стратегиях.
- + Добавьте комментарий к коду, нажав правой кнопкой мыши на блоки в ваших скриптах. Это может помочь другим понять различные части вашей программы!
- + Помогите соседу!

Педагогические методики и технологии.

Процесс достижения поставленных целей и задач программы осуществляется в сотрудничестве учащихся и педагога. При этом реализуются различные методы осуществления целостного педагогического процесса. На различных его этапах ведущими выступают отдельные, приведенные ниже методы:

Методы организации учебно-познавательной деятельности:

- Словесные - беседа, рассказ, монолог, диалог,
- Наглядные - демонстрация готовых сайтов и т.д.,
- Практические - решение творческих заданий и др.,
- Репродуктивные - работа по шаблонам,
- Проблемно –поисковые- изготовление страниц по собственному замыслу, решение творческих задач,
- Индивидуальные - задания в зависимости от достигнутого уровня развития учащегося).

Метод проектов используется на занятиях в течение всего периода обучения. Он способствует включению подростков в проектную культуру, формированию учащимися адекватной самооценки.

Методы стимулирования и мотивации учебно-познавательной деятельности (познавательные и развивающие игры, экскурсии, коллективные обсуждения и т.д.).

Методы воспитания: беседы, метод примера, педагогическое требование, создание воспитательных ситуаций, соревнование, поощрение, наблюдение, анкетирование, анализ результатов.

Выбор метода обучения зависит от содержания занятия, уровня подготовки и опыта учащихся.

На занятиях 1 и 3 блока программы преобладают методы: беседа, рассказ, диалог, выполнение упражнений.

На занятиях 2 и 4 блока программы — метод проектов, консультирования.

Основным методом проведения занятий является практическая работа, так как именно практика позволяет наиболее полно понять основные принципы алгоритмирования и начать постигать искусство программирования. На занятиях по всем темам проводится инструктаж по технике безопасности при работе с ПК.

Решению воспитательных задач, поставленных в программе, способствуют экскурсии на выставки смежных с компьютерными технологиями тематиками Санкт-Петербурга, например «Музей высоких технологий». Большое воспитательное воздействие оказывает также участие учащихся в выставках, конкурсах различного уровня.

Дидактический материал

- Инструкции по выполнению практических заданий по каждой теме занятия в формате презентаций.
- Распечатки текстов программ.
- ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
- ГОСТ 19.003-80. Схемы алгоритмов и программ. Обозначения условные, графические.
- Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240 с.: ил.

Информационные источники

Литература для преподавателей

1. Вирт Н. Алгоритмы + структуры данных = программы. — М.: Мир, 2013;
2. Креативное программирование.
3. Программирование для детей, Кэрл Вордерман, Джон Вудкок, Шон Макманус, Крейг Стили, Клэр Куигли, Дэниел Маккаферти
4. В.Г. Рындак, В.О.Дженжер, Л.В.Денисова. Проектная деятельность школьника в среде программирования Scratch. Учебно-методическое пособие. Оренбург — 2009.

Литература, рекомендуемая для учащихся:

1. Сборник задач по программированию. — 2-е изд., перераб. И доп. — СПб.: БХВ-Петербург, 2007. — 240с.
2. Голиков Д. Scratch для юных программистов. – СПб.: БХВ-Петербург. – 2017. – 192 с
3. Python для детей

Интернет-источники

1. <https://scratch.mit.edu>
2. <http://younglinux.info/scratch>
3. <https://pythontutor.ru>
4. <http://itgenio.by>
5. <http://code.org>
6. <http://codecombat.com>
7. <http://coderussia.ru>