

Муниципальное автономное общеобразовательное учреждение

«Лицей № 56»

ПРИНЯТА
на заседании Педагогического совета
протокол № 16 от «30» августа 2024 г.

УТВЕРЖДЕНА
приказом директора МАОУ «Лицей № 56» №
276к «30» августа 2024 г.

Дополнительная общеобразовательная общеразвивающая программа

технической направленности

«Школа программирования»

Возраст обучающихся 11-17 лет

Срок реализации – 3 года

Новоуральский городской округ, 2024

Пояснительная записка

Применение инструментов и средств IT-индустрии давно стало нормой практически во всех отраслях экономики. Реальный сектор, сфера услуг, телекоммуникации и бизнес – все сферы общества буквально пронизаны высокотехнологичными IT-решениями, превратившими постиндустриальное общество конца XX века в современное цифровое информационное пространство. Ключевыми, базовыми навыками, являющимися неотъемлемой частью квалификации современного IT-специалиста, являются навыки программирования, разработки программных решений.

Дополнительная общеобразовательная общеразвивающая программа "Школа программирования» (далее программа) разработана в соответствии с Законом Российской Федерации "Об образовании в Российской Федерации" от 29 декабря 2012 N 273-ФЗ (с изменениями и дополнениями) с учетом:

- нормативных документов Министерства просвещения РФ, Министерство образования и молодёжной политики Свердловской области, регламентирующих деятельность по реализации программ дополнительного образования;

- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденного приказом Министерства просвещения Российской Федерации от 9 ноября 2018 г. N 196;

- Письма Министерства образования и науки РФ от 18.11.2015 № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ»;

- Устава МАОУ «Лицей № 56»;

- других документов, регламентирующих деятельность по реализации дополнительных образовательных программ.

Актуальность программы заключается в комплексном подходе к результатам обучения. Обучение по программе позволяет обучающимся не просто накапливать знания, но и развивать интеллект, способность мыслить, формировать логическое мышление. И даже если в будущем ребенок не станет профессиональным программистом, то алгоритмическое мышление, умение мыслить наперед, просчитывать все возможные варианты, а также развитое логическое и структурное мышление сейчас просто необходимы успешному человеку в любой профессии.

Ключевыми принципами организации образовательной деятельности по программе являются:

- **принцип гуманистической направленности образования**, который предполагает отношение педагога к воспитанникам как к ответственным

субъектам собственного развития, а также стратегию взаимодействия, основанную на субъект-субъектных отношениях;

- **принцип доступности на основании** которого система дополнительного образования детей является своего рода механизмом социального выравнивания возможностей получения персонифицированного образования;

- **принцип природосообразности**, который предполагает, что образование основывается на научном понимании взаимосвязи природных и социокультурных процессов; что учащиеся воспитывают сообразно их полу и возрасту, формируют у них ответственность за развитие самих себя, за экологические последствия своих действий и поведения;

- **принцип культуросообразности**, который предполагает, что воспитание основывается на общечеловеческих ценностях, строится в соответствии с ценностями и нормами национальной культуры и региональными традициями, не противоречащими общечеловеческим ценностям;

- **принцип индивидуальности**, который реализует право ребенка на овладение знаниями и умениями в индивидуальном темпе и объеме, на смену в ходе образовательной деятельности предмета и вида деятельности, конкретного объединения и даже педагога. При этом успехи ребенка принято сравнивать в первую очередь с предыдущим уровнем его знаний и умений, а стиль, темп, качество его работы - не подвергать порицаниям.

- **принцип разновозрастного единства** обеспечивает сотрудничество обучающихся разных возрастов и педагогов. Особенно в разновозрастных объединениях ребята могут проявить свою инициативу, самостоятельность, лидерские качества, умение работать в коллективе, учитывая интересы других.

- **принцип открытости системы** направлен на совместную работу лица, семьи, других социальных институтов, учреждений культуры и образования, что обеспечивает каждому ребёнку максимально благоприятных условий для духовного, интеллектуального и физического развития, удовлетворения его творческих и образовательных потребностей.

Содержание программы распределяется по годам обучения 1 год обучения – стартовый уровень, 2 год – базовый, 3 год – продвинутый. Материал занятий предполагает переход от минимальной сложности, позволяющую каждому ребенку освоить базовые навыки и умения в области программирования, к более продвинутым.

Направленность программы – техническая. Программа ориентирована на развитие и формирование научного мировоззрения, освоение методов научного познания мира, развитие исследовательских, прикладных, конструкторских, инженерных способностей обучающихся в области точных наук и технического творчества.

Срок освоения программы составляет 3 года.

Общий объём занятий по программе 272 часа. Распределение часов по годам обучения

1 год - 68 часов в год (2 академических часа в неделю).

2 год - 102 часа в год (3 академических часа в неделю).

3 год - 102 часа в год (3 академических часа в неделю). Количество учебных недель составляет 34 недели ежегодно. Занятия проводятся в группе в соответствии с расписанием.

Форма проведения учебных занятий – групповая (командная) и индивидуальная, соревновательная (контексты) и другие. В основном применяется деятельностный подход, технология проблемного обучения. Программа рассчитана на детей 11-17 лет, не имеющих навыков программирования и/или составления и/или записи алгоритмов при помощи формальных средств визуализации обучения, в группе от 6 до 12 человек.

Цель: овладение навыками составления программ с использованием различных технологий средств, сред и технологий программирования для осуществления профессиональной деятельности, связанной со сферой ИКТ.

Задачи:

1. Создать условия для осознанного использования учащимися таких понятий как "алгоритм", "система", "программа".

2. Сформировать у учащихся умения организации собственной учебной деятельности, включающими: целеполагание, планирование, прогнозирование, контроль, коррекция, оценка.

3. Сформировать у учащихся умения и навыки моделирования как основного метода приобретения знаний, умение применять алгоритмические конструкции как модель действий, проверять адекватность модели объекту и цели моделирования.

4. Сформировать у учащихся умения и навыки использования средств информационных и коммуникационных технологий для решения прикладных задач.

5. Сформировать у учащихся умения и навыки управления объектами с помощью составленных для них моделей;

6. Развивать у учащихся умения и навыки продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: умения правильно, четко и однозначно формулировать мысль в понятной собеседнику форме; умения работы в группе; умения выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ.

7. Воспитывать ценностное отношение к достижениям науки и техники, к результатам исследовательских и проектных работ.

Для успешного выполнения программы используются различные методы и формы работы – групповая (командная) и индивидуальная, соревновательная (контексты) и другие. В основном применяется деятельностный подход, технология проблемного обучения.

Содержание программы

1. Среда визуального программирования Scratch. Общие сведения о составлении программ и об интерфейсе среды программирования Scratch. Общие навыки работы в среде Scratch. Простейшие команды Scratch для рисования графических примитивов. Система координат на плоскости. Команды позиционирования персонажа. Расширенные команды управления персонажем. Переходы по координатам, изменение координат персонажа.

Алгоритмические конструкции. Практикум составления интерактивных программ. Простейшие алгоритмические конструкции: цикл, условия. Программирование повторяющихся действий. Программирование изменения внешности персонажа – костюмы, визуальные эффекты. Сенсоры. Работа с несколькими персонажами. Взаимодействие персонажей, взаимодействие с пользователем.

Разработка творческих проектов. Практикум программирования интерактивных программ (в т.ч. игры, обучающие программы, викторины).

Общие сведения о программировании, языке программирования Pascal и среде программирования PascalABC. Общие сведения о программировании и о языках программирования высокого уровня. Парадигма императивного программирования. Общие сведения о средствах разработки программ. Компиляторы и интерпретаторы. Инструментальные средства разработки ПО. Сведения о программировании для разных операционных систем. Интерфейс Pascal ABC.net. Основные приёмы работы с файлом исходного кода программы.

Основы программирования на языке Pascal. Основы языка Pascal. Структура программы, операторы ввода/вывода, программирование простейших программ. Типы данных, переменные, оператор присваивания, операторы ввода/вывода. Практикум решения простейших вычислительных задач. Алгоритмические конструкции (условие, цикл). Практикум решения задач базового уровня сложности. Простейшие алгоритмические конструкции: цикл, условия. Практикум решения разных задач. Задачи на накопление сумм. Структуры данных в Pascal (массивы). Программирование продвинутых алгоритмов. Массивы, сортировка массивов. Практикум решения задач повышенной сложности.

2. Общие сведения и основы языка программирования Python и среды программирования Python IDLE. Общие сведения о программировании и о языках программирования высокого уровня. Парадигма императивного программирования. Общие сведения о средствах разработки программ. Компиляторы и интерпретаторы. Инструментальные средства разработки ПО. Сведения о программировании для разных операционных систем. Интерфейс среды Python IDLE. Основные приёмы работы с файлом исходного кода программы. Структура программы, операторы ввода/вывода, программирование простейших программ. Структура программы. Типы

данных, переменные, оператор присваивания, операторы ввода/вывода. Практикум решения простейших вычислительных задач.

Алгоритмические конструкции и базы данных. Практикум решения задач базового уровня сложности. Условия, циклы. Краткая и полная форма условного оператора. Цикл с параметром, функция range. Цикл while. Структуры данных в Python. Программирование продвинутых алгоритмов. Списки, словари, множества в Python. Процедуры и функции работы со строками. Двумерные списки.

3. Общие сведения об интерфейсе инструментальных сред программирования от JetBrains. Интерфейс PyCharm (Python), CLion (C++). Общие принципы организации рабочего пространства, идеология "проектов", практика редактирования кода и отладки программ.

Основы языка Python. Элементарные приёмы решения простых олимпиадных задач. Понятие эффективности алгоритма: подсчёт при вводе, обращение индексов, сортировки. Эффективность по времени и по памяти. Принципы оценки эффективности алгоритма. O-нотация.

Алгоритмы и структуры данных. Алгоритмы динамического программирования, рекурсия. Простые структуры данных – связанные списки, стек, очередь. Строковые алгоритмы. Алгоритмы на графах, деревья. Жадные алгоритмы.

Структуры данных в Python. Знакомство с прикладным программированием: практикум программирования прикладных задач. Реализация интерфейса пользователя, принципы организации ввода/вывода. Вопросы "юзабилити".

Соревнования и фестивали по программированию. Проведение соревнований между учащимися группы. Организация участия обучающихся в конкурсных и обучающих мероприятиях различного уровня.

Планируемые результаты

К концу освоения программы обучающиеся должны знать:

- общие принципы императивного программирования;
- принципы работы в блочной визуальной среде программирования;
- принципы работы простейших алгоритмических конструкций;
- правила и приёмы составления базовых алгоритмов;
- синтаксис языка программирования высокого уровня Pascal;
- принципы работы простейших алгоритмических конструкций в языке программирования Pascal;
- правила и приёмы составления базовых алгоритмов;
- синтаксис языка программирования высокого уровня Python;
- принципы работы простейших алгоритмических конструкций в языке программирования Python;
- принципы работы Python-специфичных конструкций и структур данных (списки, словари);

- правила и приёмы составления программ среднего уровня сложности;
- общие принципы программирования олимпиадных задач;
- общие принципы оценки эффективности алгоритмов;
- алгоритмы повышенного уровня сложности за пределами раздела "программирование" школьного курса "Информатика и ИКТ".

К концу освоения программы обучающиеся должны овладеть навыками:

- составления скриптов в блочно-визуальной среде программирования;
- формулирования задания для разработки программного решения прикладной задачи;
- реализации разработанного программного решения, его анализа, оценки и модернизации;
- составления программ на алгоритмическом языке высокого уровня;
- отладки программного кода;
- составления программ на алгоритмическом языке высокого уровня;
- составления эффективных программ на алгоритмическом языке высокого уровня;
- отладки программного кода;
- формулирования задания для разработки программного решения прикладной задачи;
- реализации разработанного программного решения, его анализа, оценки и модернизации.

В ходе освоения программы проводится оценка уровня освоения обучающимися основных понятий, принципов, правил программирования по следующим параметрам: правильность и скорость написания программ, скорость поиска ошибок, успешность корректировки и проработки программ.

Для оценки личных достижений обучающихся проводятся соревнования, открытые занятия внутри каждой группы, между группами одного и разных годов обучения. Результаты освоения программы обучающиеся демонстрируют через участие в конкурсных и фестивальных мероприятиях различного уровня, в том числе в дистанционных форматах.

Результаты освоения программы обучающиеся демонстрируют через участие в конкурсных и фестивальных мероприятиях различного уровня, в том числе в дистанционных форматах. Для оценки личного продвижения обучающихся проводятся соревнования, открытые занятия внутри каждой группы, между группами одного и разных годов обучения.

Учебный план

№	Модуль	Количество часов		
		общее	теория	практика
1.	Основы программирования	68	12	56
2.	Программирование Python	102	16	86
3.	Олимпиадное программирование	102	22	80
	Итого по программе	272	50	222

Промежуточная аттестация проводится в конце каждого учебного года в форме решения тестовых задач. Показателем, определяющим уровень освоения программы и динамику развития навыков обучающихся, является общее количество решённых задач на портале acm.timus.ru в соответствии с модулем обучения.

Календарный учебный график

Календарный учебный график определяет чередование учебной деятельности и плановых перерывов для отдыха и иных социальных целей (каникул) при освоении образовательной программы в течение учебного года.

Продолжительность учебного периода:

начало учебного года - не позднее 10 сентября

окончание учебного года - 29 мая

Нерабочие праздничные дни:

4 ноября - День народного единства

1, 2, 3, 4, 5, 6, 8 января - Новогодние каникулы; 7 января - Рождество Христово;

23 февраля - День защитника Отечества;

8 марта - Международный женский день;

1 мая - Праздник Весны и Труда;

9 мая - День Победы.

Продолжительность учебного периода – 34 недели. Каникулярный период определяется календарным учебным графиком образовательного учреждения на учебный год. Летние каникулы с 01 июня по 31 августа.

Режим образовательной деятельности:

Начало учебных занятий – согласно расписанию учебных групп.

Количество часов в неделю на каждую группу – 3 часа.

Продолжительность занятий - не более 45 минут.

Приложением к образовательной программе являются рабочие программы:

1. Основы программирования.
2. Программирование Python.
3. Олимпиадное программирование.

МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Формы и методы контроля, система оценивания

В процессе обучения используются следующие методы контроля:

Входной контроль проводится ежегодно начале учебного года. на первом году обучения в форме выполнения обучающимися тестовых заданий на составление скриптов для исполнителя Scratch. Примерами заданий входного контроля являются следующие:

- Составить скрипт для рисования простого рисунка из прямых линий (домик, квадрат)
- Составить скрипт для рисования орнамента средней сложности (повторяющиеся простые геометрические фигуры)

Второй, третий год обучения - в форме решения обучающимися тестовых задач на портале acm.timus.ru (сложность – до 60 условных пунктов)

Результаты входного контроля не фиксируются в официальных документах, но учитываются педагогом при формировании индивидуальной образовательной траектории обучающегося)

Текущий контроль 1 год обучения осуществляется непрерывно в ходе учебного процесса и состоит в оценке времени выполнения обучающимися учебных заданий на текущий момент. Уровень освоения темы и овладения навыками программирования по соответствующей теме оценивается максимально 2 баллами (низкий уровень, средний, высокий) исходя из следующей градации:

Раздел "Scratch"

Составление простых скриптов

- 5 – 10 минут – высокий уровень
- 10 – 20 минут – средний уровень
- более 20 минут – низкий уровень

Составление скриптов, содержащих команды повторения

- 10 – 15 минут – высокий уровень
- 15 – 25 минут – средний уровень
- более 25 минут – низкий уровень

Раздел "Паскаль"

Составление простых линейных вычислительных программ

- 10 – 15 минут – высокий уровень
- 15 – 25 минут – средний уровень

более 25 минут – низкий уровень

Составление программ базового уровня, содержащих алгоритмические конструкции и структуры данных

15 – 20 минут – высокий уровень

20 – 30 минут – средний уровень

более 30 минут – низкий уровень

Текущий контроль 2 и 3 года обучения осуществляется непрерывно в ходе учебного процесса и состоит в оценке времени выполнения обучающимися учебных заданий на текущий момент (количество решённых задач за одно занятие), а также общего количества решённых задач на портале pythontutor.ru или ulearn.me:

более 4 задач – высокий уровень развития навыков

от 2 до 4 задач – средний

менее 2 задач – низкий уровень

Промежуточная аттестация проводится в конце учебного года. Показателем, определяющим уровень освоения программы и динамику развития навыков обучающихся, является общее количество решённых задач на портале acm.timus.ru

Шкала оценивания 1 год обучения

меньше 5 – низкий уровень освоения программы

5 - 10 – средний уровень освоения программы

более 10 – высокий уровень освоения программы

второй год обучения

меньше 10 – низкий уровень освоения программы

10 - 15 – средний уровень освоения программы

более 15 – высокий уровень освоения программы

Показателем, определяющим уровень освоения программы и динамику развития навыков обучающихся, является максимальный уровень сложности решённой задачи на портале acm.timus.ru за время обучения:

меньше 50 условных пунктов – низкий уровень освоения программы

51 - 75 – средний уровень освоения программы

более 75 – высокий уровень освоения программы

С целью проверки знаний используются следующие методы:

- наблюдение за обучающимися, отслеживание динамики изменения их творческих, коммуникативных и иных способностей, личностных качеств обучающихся;
- беседы с обучающимися;
- выполнение творческих и иных заданий на занятиях;
- проведение соревнований на занятии в зависимости от его темы.

Методическое обеспечение

Отличительная особенность программы в том, что она является практико-ориентированной. Освоение обучающимися навыков программирования происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать IT-технологиями.

На занятиях используются:

- презентации, разработанные преподавателем с учётом конкретных задач;
- варианты демонстрационных программ;
- материалы по терминологии ПО;
- инструкции по настройке оборудования;
- учебная и техническая литература.

В ходе обучения используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Каждая тема программы начинается с постановки задачи – характеристики предметной области или конкретной программы на языках программирования. С этой целью проводится демонстрация презентации или самой программы, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки. Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

При реализации программы следует учитывать возрастные особенности обучающихся:

11-12 лет – младшие подростки, находящиеся в переходном периоде от младшего возраста к подростковому. Возраст связан с постепенным обретением чувства взрослости. Им характерны независимость, самостоятельность. У таких обучающихся высокая потребность в сборе информации об окружающем мире,

они хотят учиться новым навыкам. Возрасту характерна повышенная эмоциональность, тревожность.

– 13–14 лет – подростковый период. Характерная особенность – личное самосознание, сознательное проявление индивидуальности. Ведущая потребность – самоутверждение. В подростковый период стабилизируются интересы детей. Основное новообразование – становление взрослости как стремление к жизни в обществе взрослых. К основным ориентирам взросления относятся: социально моральные – наличие собственных взглядов, оценок, стремление их отстаивать; интеллектуально-деятельностные – освоение элементов самообразования, желание разобраться в интересующих подростка областях; культурологические – потребность отразить взрослость во внешнем облике, манерах поведения. Роль педагога в работе с подростками заключается в том, чтобы регулярно осуществлять их подготовку к самопрезентации социально значимой группе людей.

15–17 лет – юношеский возраст. Завершение физического и психического созревания. Социальная готовность к общественно полезному производительному труду и гражданской ответственности. В отличие от подросткового возраста, где 7 проявление индивидуальности осуществляется благодаря самоидентификации – «кто я», в юношеском возрасте индивидуальность выражается через самопроявление – «как я влияю». Основная задача педагога дополнительного образования в работе с детьми в возрасте 15–16 лет сводится к решению противоречия между готовностью их к полноценной социальной жизни и недопущением отставания от жизни содержания и организации их образовательной деятельности. Дети этого возраста отличаются внутренней уравновешенностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия. Ребятам также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценки поступков и действий ребёнка со стороны не только старших, но и сверстников. Ребёнок стремится завоевать в их глазах авторитет, занять достойное место в коллективе. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти своё место в детском коллективе. Также следует отметить, что дети данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало формированию начального профессионального самоопределения обучающихся.

На занятиях продвинутого уровня используются различные задания из олимпиадных практикумов, ориентированные на решение задач из разных областей с различными вводными данными.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый подросток получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы. Такая форма организации обучения стимулирует интерес обучающегося к предмету, активность и самостоятельность обучающихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания. Для самостоятельной работы используются разные по уровню сложности задания. Количество таких заданий в работе может варьироваться.

В образовательном процессе используются следующие методы: – объяснительно-иллюстративный; – метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой); – проектно-исследовательский; – наглядный (демонстрация видеосюжетов, схем, таблиц, диаграмм); использование технических средств; – практический (практические задания; анализ и решение проблемных ситуаций и т. д.).

Формы обучения:

– фронтальная – предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран. Активно используются Интернет-ресурсы;

– групповая – предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

– индивидуальная – подразумевает взаимодействие преподавателя с одним обучающимся. Как правило данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем обучающийся выполняют индивидуальные задания или общие задания в индивидуальном темпе;

– дистанционная – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающегося при самостоятельной работе дома. Налаженная система сетевого взаимодействия подростка и педагога, позволяет не ограничивать процесс обучения нахождением в учебной аудитории, обеспечить возможность непрерывного обучения в том числе, для часто болеющих детей или всех детей в период сезонных карантин (например, по гриппу) и температурных ограничениях посещения занятий.

Материально-технические требования

Для реализации программы каждому обучающемуся необходим компьютер с доступом в Интернет с установленным программным обеспечением Scratch, PascalABC версии не ниже 3.4, а также вспомогательным программным обеспечением общего назначения (браузер, программа для чтения pdf, средство просмотра jpeg, видеоплеер и т.п.).

Список литературы

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. М.: Наука, 2015
2. Банкрашков, А.В. Программирование для детей на языке Python – М.: АСТ, 2018
3. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования. Харьков: Фолио, Ростов н/Д: Феникс, 2016
4. Босова Л.Л. Практикум по информатике и информационным технологиям. Учебное пособие для общеобразовательных учреждений. М.: Бином. Лаборатория Базовых Знаний, 2016
5. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб.: Питер, 2017
6. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир, 2015
7. Вирт Н. Алгоритмы и структуры данных. М.: Мир, 2015.
8. Голиков Д. , Голиков А. Книга юных программистов на Scratch. Учебное пособие/Москва: Smashwords, 2014
9. Грогно П. Программирование на языке Паскаль. М.: Мир, 2015
10. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019
11. Дагене В.А., Григас Г. К., Аугутис К.Ф. 100 задач по программированию. М.: Просвещение, 2013
12. Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы; Пер. с англ. под ред. А. Шеня. — М.: МЦНМО, 2014
13. Климова М.П. Free Pascal. Практическое программирование. Издание 2. М.: ОМЕГА, 2015
14. Потопахин В. В. Pascal. Решение сложных задач. С-Пб.: БХВ, 2014
15. Рындак В.Г., Дженжер В.О., Денисова Л.В. Проектная деятельность школьника в среде программирования Scratch. Учебно-методическое пособие. Оренбург, Евразия: 2016
16. Сорокина Т.Е., поурочные разработки "Пропедевтика программирования со Scratch" для 5-го класса. М.: Просвещение, 2015 г.
17. Саммерфилд, М. Программирование на Python 3. Подробное руководство – М.: Символ, 2018

18. Шень А. Программирование: теоремы и задачи. М.: МЦНМО, 2017

Используемые Интернет-ресурсы:

scratch.mit.edu

<https://younglinux.info/scratch/>

<https://vk.com/scratchcommand>

acm.timus.ru

acmp.ru

codeforces.com infoimatics.mccme.ru