

Муниципальное автономное общеобразовательное учреждение города
Новосибирска
«Инженерный лицей Новосибирского государственного технического
университета»

Рекомендовано решением
педагогического совета МАОУ
«Инженерный лицей НГТУ»
Протокол № 1 от 26.08. 2025

Утверждаю
Директор МАОУ
«Инженерный лицей НГТУ»
Безлекина М.А.
Приказ №228 от 26.08.2025



**Рабочая программа по курсу внеурочной деятельности
«Программирование»**

название учебного предмета, курса, дисциплины (модуля)

для класса(ов) 7-9

Количество часов: 1 час в неделю

Всего: 102 часа

Из них: 7 класс 34 часа

8 класс 34 часа

9 класс 34 часа

Разработчики программы:

Муль Павел Фридрихович учитель информатики высшей квалификационной категории,
Гурина Анна Анатольевна учитель информатики высшей квалификационной категории,
Цысс Евгения Сергеевна учитель информатики высшей квалификационной категории,
Орлова Анна Алексеевна учитель информатики первой квалификационной категории.

г. Новосибирск
2025

Программа обсуждалась на заседании кафедры / методического объединения
учителей математики и информатики МАОУ «Инженерный лицей НГТУ»
Протокол заседания № 1 от 26.08.2025 г.

Подолян Е.В / Подолян Е.В
Ф.И.О. руководителя кафедры/МО

Активный процесс развития информатизации характеризуется и новыми требованиями к общеобразовательной школе. Отражением этой тенденции является потребность в подготовке учащихся, ориентированных на изучение ИТ-технологий. Изучение алгоритмов и языков программирования является неотъемлемой частью информатики. Программирование вырабатывает у учащихся логическое и комбинаторное мышление, творческие способности. Программирование - стержень курса информатики. Владея одним из языков программирования, у учащегося появляется интерес к изучению других языков программирования.

Еще одним аспектом, направляющим на изучение программирования, является желание обучающихся в дальнейшем изучать данную область на профессиональном уровне. Развитию интереса учащихся в этой области, а также подготовке к олимпиадам по информатике способствует углубленное изучение таких разделов как «алгоритмизация» и «программирование», на изучение которых в основной программе отводится недостаточное количество времени. Курс внеурочной деятельности «Программирование» рассчитан на 3 года изучения в количестве 102 часов по 1 часу в 7 классе-34 часа, в 8 классе-34 часа, в 9 классе-34 часа. Состоит из модулей: «Программирование от Scratch до Android» - 7 класс, «Программирование на языке программирования Python» - 8 класс, «Программирование на языке программирования C++» - 9 класс.

Отличительная особенность данной программы от существующих образовательных программ заключается в том, что изучается материал, слабо представленный и не представленный в программе основного курса информатики и ИКТ. Он систематизирован, доступно и логично излагается, подкреплен мощным дидактическим материалом, направлен на практику программирования и подготовку к олимпиадам, на развитие творчества и самостоятельности обучающихся. На занятиях организуется деятельность, создающая условия для творческого развития воспитанников на различных возрастных этапах и предусматривающая их дифференциацию по степени одаренности. Данный курс позволяет повысить результативность обучения информатике и ИКТ при параллельном преподавании школьного основного курса и данного дополнительного.

Модуль 1. «Программирование от Scratch до Android».

Первый модуль, на данном этапе обучения позволит погрузить учащихся в мир логики, математического моделирования для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации. Углубленное изучение обозначенных разделов позволит учащимся развивать творческое мышление, находить самостоятельные индивидуальные решения в проблемных ситуациях, а полученные умения и навыки применять в жизни. Развитие творческих способностей будет способствовать дальнейшему профессиональному самоопределению. Алгоритмы и программирование можно рассматривать как резерв для формирования логического мышления. Постановка «дисциплины действий» служит компьютерное программирование (проектирование). Оно хорошо тем, что, во-первых, школьники учатся самостоятельно и коллективно работать. Компьютерное проектирование включает в себя следующие этапы: анализ аналогов и предметной области, подготовка технического задания, распределение работы на этапы, техническая работа над проектом и его презентация. Во-вторых, учащиеся сталкиваются с какой-либо предметной областью (зачастую не связанной непосредственно со школьными предметами) и учатся работать с материалом. Таким образом, происходит актуализация знаний, полученных на других предметах.

Изучение модуля будет осуществляться посредством работы с объектно-ориентированным исполнителем Scratch, языком программирования КуМир (Комплект Учебных МИРов), заканчивается курс созданием простых приложений для операционной системы Android в среде AppInventor.

В модуль включен дополнительный материал, который значительно расширит возможности формирования универсальных учебных и предметных навыков. В данном курсе на конкретных примерах рассматривается программирование на алгоритмическом языке с русской лексикой в системе КуМир с встроенными исполнителями Черепаха, Чертежник/Рисователь и Робот. А также объектные исполнители Scratch и среда для разработки

под AndroidAppInventor. С их помощью можно создавать и решать интересные задачи, которые почти в игровой форме помогут учащимся осваивать работу с ветвлениями и циклами.

КуМир объединяет в себе черты многих языков программирования, но в то же время, данный язык очень прост для учащихся этого возраста в силу своей близости к естественному языку. Следовательно, КуМир как начальный язык программирования позволит значительно облегчить дальнейшее изучение профессиональных языков. Он послужит вспомогательной ступенькой для изучения в старших классах языков более высокого уровня. Кроме того, система программирования КуМир имеет ряд неоспоримых плюсов:

- Среда разработана в НИИСИ РАН по заказу Российской Академии Наук, является СПО (Свободное программное обеспечение) и распространяется на условиях лицензии GNU GPL 2.0;

- КУМИР на русском языке, все конструкции команды и переменные задаются простыми русскими словами;

- включает 5 различных исполнителей, которые помогут учащимся понять, что такое алгоритм и научиться писать их;

- имеются готовые алгоритмические конструкции, которые можно вставлять с помощью команды меню Вставка в свою программу;

- отступы проставляются автоматически, что позволяет приучить учеников к правильно отформатированному коду и его наглядному представлению;

- в системе содержится подробно расписанная справка на русском языке, которая помогает в постижении основ программирования не только учащимся, но и сделает работу учителя информатики намного проще;

- система кроссплатформенная и возможно установление и работа с ней как в Windows так и в Linux.

Среды программирования Scratch и AppInventor так же являются кроссплатформенными и работают в браузере.

Специфика уроков модуля состоит в том, что они строятся на уникальной дидактической базе – предметно-практической деятельности, которая является для учащихся необходимым звеном целостного процесса духовного, нравственного и интеллектуального развития.

Основной целью модуля является естественно-научное обучение, воспитание и развитие обучающихся в области алгоритмизации и программирования.

Достижение цели предполагает решение следующих задач:

- уметь быстро ориентироваться в динамично развивающемся и обновляющемся информационном пространстве;

- получать, использовать и создавать разнообразную информацию;

- принимать обоснованные решения и решать жизненные проблемы на основе полученных знаний, умений и навыков;

- осваивать и систематизировать знания, относящиеся к математическим объектам информатики;

- строить описания объектов и процессов, позволяющих осуществлять их компьютерное моделирование; средства моделирования; информационные процессы в технологических и социальных системах;

- развивать алгоритмическое мышление, способности к формализации, элементы системного мышления;

- воспитывать культуру проектной деятельности, в том числе умения планировать, работать в коллективе; чувство ответственности за результаты своего труда, используемые другими людьми; установку на позитивную социальную деятельность в информационном обществе, недопустимость действий, нарушающих правовые и этические нормы работы с информацией;

- приобретать опыт создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; построения компьютерных моделей, коллективной реализации информационных проектов, преодоления трудностей в процессе интеллектуального проектирования, информационной деятельности в различных сферах, востребованных на рынке труда; решения сложных задач и олимпиадных задач программирования.

Модуль 2. «Программирование на языке программирования Python».

По результатам исследования компании Jet Brains (<https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/>) язык Python – самый изучаемый в 2019 году и один из самых востребованных на рынке труда. Изучение Python в школе откроет ученикам возможности дальнейшего развития в области ИТ и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решении задачий КЕГЭ.

Данный модуль направлен на подготовку творческой, технически грамотной, гармонично развитой личности, обладающей логическим мышлением, способной анализировать и решать задачи в команде в области информационных технологий, решать ситуационные кейсовые задания, основанные на групповых проектах. Занятия по данному курсу рассчитаны на общенаучную подготовку обучающихся, развитие их мышления, логики, математических способностей, исследовательских навыков. В рамках курса обучающиеся смогут познакомиться с физическими, техническими и математическими понятиями. Приобретённые знания будут применимы в творческих проектах.

Данный модуль способствует совершенствованию и развитию важнейших знаний и умений в области информатики, помогает оценить обучающемуся свои возможности при изучении информатики и более осознанно выбрать профиль дальнейшего обучения. Модуль «Основы программирования на языке Python» является логическим продолжением модуля для 7 класса «Программирование от Scratch до Android».

Основной целью модуля является формирование у обучающихся навыков программирования на языке Python.

Достижение цели предполагает решение следующих задач:

- сформировать у обучающихся алгоритмическую культуру;
- обучить структурному программированию как методу, предусматривающему создание понятных, локально простых и удобочитаемых программ, характерными особенностями которых является: модульность, использование унифицированных структур следования, выбора и повторения, отказ от неструктурированных передач управления, ограниченное использование глобальных переменных;
- выработать навыки алгоритмизации учащимися в ее структурном варианте; освоение всевозможных методов решения задач;
- развивать алгоритмическое мышление учащихся;
- формировать навыки грамотной разработки программ.

Модуль 3. «Программирование на языке программирования C++».

Изучение основ программирования связано с целым рядом умений и навыков (организация деятельности, планирование ее и т.д.), которые по праву носят общеинтеллектуальный характер и формирование которых - одна из приоритетных задач современной школы. Поэтому использование больших возможностей программирования, решения соответствующих задач для развития мышления школьников, формирования многих общеучебных, общеинтеллектуальных умений и навыков является актуальным в современном образовании.

Основной целью модуля является формирование у обучающихся навыков программирования на языке программирования C++.

Достижение цели предполагает решение следующих задач:

- научить создавать программы для решения прикладных задач разного уровня в среде Dev-C++ с открытым исходным кодом, включающая компилятор GCC;
- научить понимать основные принципы объектно-ориентированного программирования;
- развивать алгоритмическое мышление учащихся;
- развивать навыки грамотной разработки программ.
- формировать новый тип мышления – операционный, который направлен на выбор оптимальных решений;
- предоставить возможность узнать новое в области компьютерного программирования;

- формировать представления о роли компьютерного программирования в развитии общества, изменении содержания и характера деятельности человека.

Целесообразность изучения данного модуля определяется тем, что одним из главных достоинств языка программирования СИ++ является сочетание свойств как высокоуровневых, так низкоуровневых языков. Кроме того, важно то, что СИ++ создавался с одной важной целью: сделать более простым написание больших программ с минимумом ошибок по правилам процедурного программирования, не усложняя работу компилятора, как это всегда делают языки очень высокого уровня. Программы на С++ работают значительно быстрее, чем на других языках. Также, при выборе языка программирования был учтен фактор распространенности, имеющий как психологическое значение (влияя на мотивацию учащихся), так и практическое (востребованность получаемых знаний без необходимости переучивания).

Знания, полученные при изучении модуля «Программирование на Си++», учащиеся могут применить для решения прикладных задач разного рода, повышения качества подготовки к успешной сдаче ОГЭ, КЕГЭ. Полученные знания и умения являются основой для последующего изучения программирования в высших профессиональных образовательных учреждениях.

Предметом изучения являются принципы и методы программирования в среде бесплатно распространяемой оболочки [Dev-C](#) с открытым исходным кодом, включающая компилятор [GCC](#).

Предполагается, что данный модуль является продолжением модуля «Основы программирования на языке Python», учащиеся предварительно изучили модуль «Программирование от Scratch до Android», имеют соответствующие навыки.

Рабочая программа разработана на основе следующих нормативно-правовых документов:

- Федеральный закон от 29.12.2012 года № 273-ФЗ «Об образовании в Российской Федерации» с изм. от 27.12.2019 г.
- Федеральный государственный образовательный стандарт среднего общего образования (с изменениями и дополнениями)
- Основная образовательная программа основного общего образования МАОУ «Инженерный лицей НГТУ».
- Учебный план МАОУ «Инженерный лицей НГТУ».

Учебно-методическое и материально-техническое обеспечение образовательного процесса

Литература

Модуль 1. «Программирование от Scratch до Android»

1. Босова Л.Л. «Изучаем алгоритмiku. Мой КуМир»: Бином 2018
2. Голиков Д.В. «SCRATCH для юных программистов». – СПб: БХВ – Петербург, 2017 – 192с.:ил
3. Кущниренко А.Г., Лебедев Г.В, Зайдельман Я.Н.// Информатика 7-9 классы. Учебник для общеобразовательных учебных заведений.
4. Ливинец М.М., Ярмахов Б.Б «Программирование мобильных приложений в MITAppInventor» практикум: Академия мобильных приложений.
5. <http://Lyceum.nstu.ru/sdo/course/view.php?id=2§ion=6> Курс программирования для 7 класса Муль П.Ф.
6. <http://kpolyakov.spb.ru/school/kumir.htm> (сайт К. Полякова «Преподавание, наука и жизнь»).

Модуль 2. «Программирование на языке программирования Python»

1. «Программируем на Python» (автор: М. Доусон; СПб.: Питер, 2014г.),
2. «Python 3. Самое необходимое» (авторы: Н. Прохоренок , В. Дронов; БХВ Петербург, 2016г.).

3. Поляков К.Ю. Программирование. Python. С++. Часть 1: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 144 с.: ил.
4. Поляков К.Ю. Программирование. Python. С++. Часть 2: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 176 с.: ил.
5. Поляков К.Ю. Программирование. Python. С++. Часть 3: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 208 с.: ил.
6. Поляков К.Ю. Программирование. Python. С++. Часть 4: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 192 с.: ил.
7. <http://lyceum.nstu.ru/sdo/course/view.php?id=2§ion=6> Курс программирования для 8 класса Гурина А.А.
8. <http://kpolyakov.spb.ru/school/kumir.htm> (сайт К. Полякова «Преподавание, наука и жизнь»).

Модуль 3. «Программирование на языке программирования С++»

1. Поляков К.Ю. Программирование. Python. С++. Часть 1: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 144 с.: ил.
2. Поляков К.Ю. Программирование. Python. С++. Часть 2: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 176 с.: ил.
3. Поляков К.Ю. Программирование. Python. С++. Часть 3: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 208 с.: ил.
4. Поляков К.Ю. Программирование. Python. С++. Часть 4: учебное пособие /К.Ю. Поляков. – М.:БИНОМ. Лаборатория знаний, 2019.- 192 с.: ил.
5. Страуструп Б. Программирование: принципы и практика с использованием СИ++, 2-е изд.; Пер. с англ. - М.: ООО «И. Д. Вильямс», 2016.
6. Липпман С. Б., Лажойе Ж., Му Б.Э. Язык программирования СИ++. Базовый курс, 5-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2014.
7. Шилдт Г. СИ++. Базовый курс, 3-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
8. Лафоре Р. Объектно-ориентированное программирование в СИ++, 4-е изд. Пер. с англ. – СПб.: Питер, 2004.
9. Мюссер Д., Дердж Ж., Сейни А. СИ++ и STL: справочное руководство, 2-е изд. Пер. с англ. - М.: ООО «И. Д. Вильямс», 2010.
10. Литвиненко Н. А. Технология программирования на СИ++. Начальный курс.. – СПб.: БХВ-Петербург, 2010 . – 281 с.
11. Романов Е.Л. Си++. От дилетанта до профессионала – СПб.: БХВ-Петербург, 2014 .-600 с.
12. Федеральный государственный образовательный стандарт среднего (полного) общего образования. Утвержден приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413.
13. <http://lyceum.nstu.ru/sdo/course/view.php?id=2§ion=6> Курс программирования для 9 класса Муль П.Ф.
14. <http://kpolyakov.spb.ru/school/kumir.htm> (сайт К. Полякова «Преподавание, наука и жизнь»).

Для поддержки курса разработано электронное учебное пособие в системе СДО (<http://lyceum.nstu.ru/sdo/>) по каждому модулю, которое содержит теоретический материал, задания для выполнения практических работ, презентации, тесты, самостоятельные работы, используемые во время уроков для самостоятельной работы и в качестве справочника.

Технические средства

Требования к комплектации компьютерного класса

Для реализации учебного курса «Программирование от Scratch до Android» необходимо наличие компьютерного класса в соответствующей комплектации. Наиболее рациональным является установка в компьютерном классе 13–15 компьютеров (рабочих мест) для школьников и одного компьютера (рабочего места) для педагога. Предполагается объединение компьютеров в локальную сеть с возможностью выхода в Интернет, что позволяет использовать сетевые цифровые образовательные ресурсы.

Минимальные требования к техническим характеристикам каждого компьютера следующие:

- процессор – не ниже *Celeron* с тактовой частотой 2,8 ГГц;
- оперативная память – не менее 1Гб;
- жидкокристаллический монитор с диагональю не менее 15 дюймов;
- жёсткий диск – не менее 80 Гб;
- клавиатура;
- мышь;
- устройство для чтения компакт-дисков (желательно);
- аудиокарта и акустическая система (наушники или колонки).

Кроме того, в кабинете информатики должны быть:

- принтер на рабочем месте учителя;
- проектор на рабочем месте учителя;
- сканер на рабочем месте учителя;
- телекоммуникационный блок;
- устройства, обеспечивающие подключение к сети;
- устройства вывода звуковой информации;
- устройства для ручного ввода текстовой информации и манипулирования экранными элементами.

Требования к программному обеспечению компьютеров:

На компьютерах должна быть установлена операционная система *Windows* или *Linux*, а также необходимое программное обеспечение:

- файловый менеджер (в составе операционной системы или др.);
- антивирусная программа;
- программа-архиватор;
- программа-переводчик;
- мультимедиа-проигрыватель (входит в состав операционных систем или др.);
- программы Кумир, Scratch;
- браузер (входит в состав операционных систем или др.);
- программа для записи CD и DVD дисков.
- интегрированная среда для программирования на языках C и C++, работающая под управлением операционной системы Windows. Среда Dev-C, распространяемая свободно с исходными кодами (на Delphi) по лицензии GPL Dev-C <http://www.bloodshed.net/dev/devcpp.html>.
- текстовый редактор (Блокнот или Gedit) и текстовый процессор (MS Word или OpenOffice.org Writer);

Электронные образовательные ресурсы

1. <http://www.edu.ru> (Федеральный портал «Российское образование»)
2. <http://school.edu.ru> (Российский общеобразовательный портал)
3. <http://www.computer-museum.ru> (Виртуальный компьютерный музей)
4. <http://inf.1september.ru> (Газета «Информатика» издательского дома «Первое сентября»)
5. <https://www.niisi.ru/kumir/> (система программирования Кумир)
6. [Сайт разработчиков Кумира, версия 1.9](http://lpm.org.ru) (lpm.org.ru).
7. [Версия Кумир 2.x](http://lpm.org.ru) (lpm.org.ru).
8. [Кумир на сайте НИИСИ РАН](http://www.niisi.ru/kumir) (www.niisi.ru/kumir).
9. [https://scratch.mit.edu/Scratch](http://scratch.mit.edu), версия для ПК и браузерная
10. <http://appinventor.mit.edu/explore/AppInventor>
11. СДО Инженерного лицея НГТУ <http://lyceum.nstu.ru/sdo>
12. компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещенный на сайте авторского коллектива: <http://kpolyakov.spb.ru/school/ruscrp.htm>;
13. материалы для подготовки к итоговой аттестации по информатике в форме ОГЭ/ ЕГЭ, размещенные на сайте <http://kpolyakov.spb.ru/school/ege.htm>;

14. методические материалы для учителя, размещенные на сайте автора <http://kpolyakov.spb.ru/school/ruscpp.htm>;
15. подборка электронных образовательных ресурсов (далее ЭОР) с портала ФЦИОР (<http://www.fcior.edu.ru>);
16. сетевая методическая служба авторского коллектива для педагогов на сайте издательства <http://metodist.lbz.ru/authors/informatika/7/>;
17. интегрированная среда для программирования на языках С и С++, работающая под управлением операционной системы Windows. Среда Dev-C, распространяемая свободно с исходными кодами (на Delphi) по лицензии GPL [Dev-C](#).

Содержание курса

Модуль 1. «Программирование от Scratch до Android».

Исполнители и алгоритмы.

Исполнитель Scratch.

Среда программирования «КуМир»: алгоритмические конструкции и переменные.

Исполнитель среды КуМир: Рисователь.

AppInventor (среда для разработки под Android).

Модуль 2. «Программирование на языке программирования Python».

Знакомство с языком Python

Условные предложения

Циклы

Функции

Строки - последовательности символов

Сложные типы данных

Стиль программирования и отладка программ

Модуль 3. «Программирование на языке программирования C++».

Основные управляющие конструкции

Первые программы.

Диалоговые программы.

Компьютерная графика.

Процедуры.

Обработка целых чисел.

Обработка вещественных чисел.

Ветвлениия.

Циклы.

Анимация.

Тематическое планирование

Модуль 1. «Программирование от Scratch до Android» - 7 класс. (35 часов)

Название раздела (модуля), темы, количество часов	Характеристика основных видов деятельности ученика
1. Исполнители и алгоритмы – 2 часа	<p>Владеть понятиями алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма.</p> <p>Соблюдать структуру алгоритма.</p> <p>Отличать операторы ввода-вывода. применять операторы ввода-вывода в алгоритме. Использовать переменные.</p> <p>Применять полученные знания к решению задач.</p>
2. Исполнитель Scratch -7 часов	<p>Знать объекты среды Scratch, назначение этих объектов и правила поведения исполнителя, команды</p>

	<p>Scratch.</p> <p>Уметь определять и исправлять ошибки в программе.</p> <p>Знать и уметь использовать режимы выполнения программы: пошаговый, до курсора, полностью.</p> <p>Владеть навыками работы с Исполнителем Scratch.</p> <p>Соблюдать правила составления программы для Scratch.</p> <p>Уметь загружать задачи и выполнять программы.</p> <p>Создавать новый сюжет и ставить задачи для Scratch.</p> <p>Выполнять постановку задачи, анализ программы, составлять словесный алгоритм решения задачи.</p> <p>Отличать и уметь составлять линейные, разветвляющиеся и циклические конструкции.</p> <p>Применять полученные знания к решению задач.</p>
3. Среда программирования «КуМир»: алгоритмические конструкции и переменные – 7 часов	<p>Знать способы задания и правила написания команд в среде программирования КуМир.</p> <p>Создавать диалоговые программы.</p> <p>Отличать типы переменных.</p> <p>Владеть понятиями условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные.</p> <p>Представлять и применять сложные условия в написании алгоритмов.</p> <p>Владеть понятиями: циклический алгоритм, цикл с предусловием, цикл с постусловием, цикл по переменной.</p> <p>Составлять программы с использованием конструкций циклов различных видов.</p> <p>Применять полученные знания к решению задач.</p>
4. Исполнитель среды КуМир: Рисователь - 9 часов	<p>Знать способы задания и правила написания команд в среде программирования КуМир. Правила использования процедур в алгоритме. Принципы построения рекурсивного алгоритма.</p> <p>Писать, вводить, редактировать и выполнять программы для исполнителя Рисователь.</p> <p>Строить линии и замкнутые фигуры.</p> <p>Применять два формата задания цвета линий и цвета заливки.</p> <p>Использовать процедуры в алгоритме, решать рекурсивные задачи.</p> <p>Применять принципы анимации при создании анимированных объектов, в том числе с обработкой нажатия клавиши.</p>
5. AppInventor - 9 часов	<p>Знать объекты среды AppInventor, назначение этих объектов и правила поведения исполнителя, команды AppInventor, ошибки в работе с исполнителем.</p> <p>Соблюдать правила составления программы для AppInventor.</p> <p>Уметь загружать задачи и выполнять программы.</p> <p>Создавать новый сюжет и ставить задачи для AppInventor.</p> <p>Выполнять постановку задачи, анализ программы, составлять словесный алгоритм решения задачи.</p> <p>Отличать и уметь составлять линейные,</p>

	<p>разветвляющиеся и циклические конструкции.</p> <p>Загружать готовые проекты на Android устройство.</p> <p>Создавать новые приложения и ставить задачи для AppInventora.</p> <p>Отличать и уметь составлять линейные, разветвляющиеся и циклические конструкции.</p>
--	--

Модуль 2. «Программирование на языке программирования Python» - 8 класс. (36 часов)

Название раздела (модуля), темы, количество часов	Характеристика основных видов деятельности ученика
1. Знакомство с языком Python – 2 часа	<p>Знать структуру программы на Python, · режимы работы с Python.</p> <p>Выполнять установку среды программирования.</p> <p>Уметь составлять простейшую программу, запускать ее в интерактивной среде.</p> <p>Писать комментарии в программе.</p>
2. Переменные и выражения - 5 часов	<p>Знать операторы ввода-вывода.</p> <p>Уметь работать со справочной системой среды программирования.</p> <p>Решать задачи на элементарные действия с числами.</p> <p>Уметь пользоваться интерфейсом среды программирования Python.</p> <p>Использовать команды редактора программы.</p> <p>Организовывать ввод и вывод данных.</p> <p>Записывать арифметические выражения, учитывая приоритет действий.</p>
3. Условные предложения – 4 часов	<p>Знать назначение условного оператора, способы записи условного оператора.</p> <p>Знать и использовать логические операторы or, and, not.</p> <p>Создавать сложные условия с помощью логических операторов.</p> <p>Уметь решать задачи по теме "Условные операторы".</p> <p>Составлять программы с конструкциями ветвления, используя простые и сложные условия.</p>
4. Циклы - 6 часов	<p>Знать правила записи циклов условием, уметь использовать данные правила при написании программ.</p> <p>Знать и использовать назначение и особенности использования цикла с параметром, его форматы записи.</p> <p>Использовать при решении задач операторы for и while.</p> <p>Уметь реализовывать циклические алгоритмы, составлять программы с циклом.</p> <p>Уметь определять вид цикла, наиболее удобный для решения поставленной задачи.</p>

5. Функции -4 часа	<p>Знать способы описания функции. Уметь использовать при решении задач принципы структурного программирования. Уметь использовать локальные и глобальные переменные в подпрограммах. Знать понятие формальных и фактических параметров подпрограмм, уметь использовать различные способы передачи параметров подпрограмм. Уметь создавать и использовать функции простые и рекурсивные и использовать их при решении задач. Уметь использовать механизм параметров для передачи значений в программах.</p>
6. Строки - последовательности символов - 3 часа	<p>Знать назначение строкового типа данных. Использовать операторы для работы со строками. Знать процедуры и функции для работы со строками. Уметь использовать операции со строками при решении задач:</p> <ul style="list-style-type: none"> – описывать строки; – соединять строки; – находить длину строки; – вырезать часть строки; – находить подстроку в строке; – находить количество слов в строке; – находить количество символов в строке; – находить заданный символ в строке.
7. Сложные типы данных - 6 часов	<p>Знать сложные типы данных и способы описания списка. Уметь использовать при решении задач способы доступа к элементам списка, способы описания кортежа, способы описания словаря, операции, выполняемые со списками, кортежами и словарями, способы описания множества, операторы работы с множествами. Выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка при решении задач. Использовать вложенные списки; приводить примеры использования вложенных списков (матриц). Уметь определять с помощью программы принадлежность элемента множеству.</p>
8. Стиль программирования и отладка программ – 4 часа	<p>Знать, что такое стиль программирования. Уметь использовать правила именования объектов. Учитывать основные рекомендации «Эффективного кода» при написании программ. Уметь находить ошибки в программе, определять их тип. Выполнять тестирование и отладку программ.</p>

Модуль 3. «Программирование на языке программирования C++» - 9 класс. (34 часа)

Название раздела (модуля),	Характеристика основных видов деятельности
----------------------------	--

темы, количество часов	ученика
1. Первые программы- 2 часа	Владеть понятиями алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма. Соблюдать структуру программы. Отличать операторы ввода-вывода. применять операторы ввода-вывода в программе при создании диалога с пользователем в режиме компиляции. Использовать переменные.
2. Диалоговые программы- 3 часа	Создавать диалоговые программы. Отличать типы переменных. Владеть приемами применения типов переменных. Применять полученные знания к решению задач.
3. Компьютерная графика- 2 часа	Владеть понятиями: окно, координаты, оси координат, пиксель, цвет контура, цвет заливки, код цвета, прозрачный цвет. Использовать в программном коде библиотеку TX Library. Управлять пикселями. Строить линии и замкнутые фигуры.
4. Процедуры- 3 часа	Владеть понятиями: сложность алгоритма, подпрограмма, процедура, рефакторинг, аргументы, параметры, базовая точка. Отличать основные конструкции обработки числовой и текстовой информации. Строить и анализировать длинную программу. Выполнять рефакторинг программного кода. Использовать в программном коде процедуры с параметрами.
5. Обработка целых чисел- 4 часа	Владеть понятиями: предельное значение числа, вещественное и целое число, остаток, форматный вывод, случайные числа, зерно. Представлять предельные значения чисел. Отличать вещественные и целые числа. Представлять число в дискретной форме. Понимать и применять методы программного повышение точности вычислений и принципы хранения в памяти целых чисел: со знаком, без знака.
6. Обработка вещественных чисел- 4 часа	Владеть понятиями: вещественное число, научный формат, мантисса, форматный вывод, округление. Понимать и применять принципы хранения в памяти вещественных чисел и нормализованное представление вещественных чисел в компьютере. Применять в программном коде арифметические операции с вещественными числами: сложение и вычитание, умножение и деление.
7. Ветвления- 7 часов	Владеть понятиями условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные. Отличать полную и неполную формы записи условного оператора. Представлять и применять метод сложных условий в написании программ. Строить сложные условия в условном операторе. Применять переключатель switch в условиях с

	множественным выбором.
8. Циклы- 6 часов	<p>Владеть понятиями: циклический алгоритм, цикл с предусловием, цикл с постусловием, цикл по переменной.</p> <p>Отличать конструкции циклов с предусловием, с постусловием, с переменными (счетчиком), вложенных.</p> <p>Составлять программы с использованием конструкций цикла с предусловием, цикла с постусловием, цикла с переменными (счетчиком), вложенных циклов.</p>
9. Анимация - 3 часа	<p>Владеть понятиями: анимация, процедура, пауза, нажатие клавиши.</p> <p>Понимать и применять принципы анимации.</p> <p>Рисовать объемные фигуры.</p> <p>Применять принципы анимации при создании анимированных объектов, в том числе с обработкой нажатия клавиши.</p>

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ ПО ИНФОРМАТИКЕ НА УРОВНЕ ОСНОВНОГО ОБЩЕГО ОБРАЗОВАНИЯ

Изучение информатики на уровне основного общего образования направлено на достижение обучающимися личностных, метапредметных и предметных результатов освоения содержания учебного предмета.

ЛИЧНОСТНЫЕ РЕЗУЛЬТАТЫ

Личностные результаты имеют направленность на решение задач воспитания, развития и социализации обучающихся средствами учебного предмета.

В результате изучения информатики на уровне основного общего образования у обучающегося будут сформированы следующие личностные результаты в части:

1) патриотического воспитания:

ценостное отношение к отечественному культурному, историческому и научному наследию, понимание значения информатики как науки в жизни современного общества, владение достоверной информацией о передовых мировых и отечественных достижениях в области информатики и информационных технологий, заинтересованность в научных знаниях о цифровой трансформации современного общества;

2) духовно-нравственного воспитания:

ориентация на моральные ценности и нормы в ситуациях нравственного выбора, готовность оценивать своё поведение и поступки, а также поведение и поступки других людей с позиции нравственных и правовых норм с учётом осознания последствий поступков, активное неприятие асоциальных поступков, в том числе в Интернете;

3) гражданского воспитания:

представление о социальных нормах и правилах межличностных отношений в коллективе, в том числе в социальных сообществах, соблюдение правил безопасности, в том числе навыков безопасного поведения в интернет-среде, готовность к разнообразной совместной деятельности при выполнении учебных, познавательных задач, создании учебных проектов, стремление к взаимопониманию и взаимопомощи в процессе этой учебной деятельности, готовность оценивать своё поведение и поступки своих товарищей с позиции нравственных и правовых норм с учётом осознания последствий поступков;

4) ценностей научного познания:

сформированность мировоззренческих представлений об информации, информационных процессах и информационных технологиях, соответствующих современному уровню развития науки и общественной практики и составляющих базовую основу для понимания сущности научной картины мира;

интерес к обучению и познанию, любознательность, готовность и способность к самообразованию, осознанному выбору направленности и уровня обучения в дальнейшем;

овладение основными навыками исследовательской деятельности, установка на осмысление опыта, наблюдений, поступков и стремление совершенствовать пути достижения индивидуального и коллективного благополучия;

сформированность информационной культуры, в том числе навыков самостоятельной работы с учебными текстами, справочной литературой, разнообразными средствами информационных технологий, а также умения самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

5) формирования культуры здоровья:

осознание ценности жизни, ответственное отношение к своему здоровью, установка на здоровый образ жизни, в том числе и за счёт освоения и соблюдения требований безопасной эксплуатации средств информационных и коммуникационных технологий;

6) трудового воспитания:

интерес к практическому изучению профессий и труда в сферах профессиональной деятельности, связанных с информатикой, программированием и информационными технологиями, основанными на достижениях науки информатики и научно-технического прогресса;

осознанный выбор и построение индивидуальной траектории образования и жизненных планов с учётом личных и общественных интересов и потребностей;

7) экологического воспитания:

осознание глобального характера экологических проблем и путей их решения, в том числе с учётом возможностей информационных и коммуникационных технологий;

8) адаптации обучающегося к изменяющимся условиям социальной и природной среды:

освоение обучающимися социального опыта, основных социальных ролей, соответствующих ведущей деятельности возраста, норм и правил общественного поведения, форм социальной жизни в группах и сообществах, в том числе существующих в виртуальном пространстве.

МЕТАПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

Метапредметные результаты освоения программы по информатике отражают овладение универсальными учебными действиями – познавательными, коммуникативными, регулятивными.

Познавательные универсальные учебные действия

Базовые логические действия:

умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логические рассуждения, делать умозаключения (индуктивные, дедуктивные и по аналогии) и выводы;

умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

самостоятельно выбирать способ решения учебной задачи (сравнивать несколько вариантов решения, выбирать наиболее подходящий с учётом самостоятельно выделенных критериев).

Базовые исследовательские действия:

формулировать вопросы, фиксирующие разрыв между реальным и желательным состоянием ситуации, объекта, и самостоятельно устанавливать искомое и данное;

оценивать на применимость и достоверность информацию, полученную в ходе исследования;

прогнозировать возможное дальнейшее развитие процессов, событий и их последствия в аналогичных или сходных ситуациях, а также выдвигать предположения об их развитии в новых условиях и контекстах.

Работа с информацией:

выявлять дефицит информации, данных, необходимых для решения поставленной задачи;

применять различные методы, инструменты и запросы при поиске и отборе информации или данных из источников с учётом предложенной учебной задачи и заданных критерии;

выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления;

самостоятельно выбирать оптимальную форму представления информации и иллюстрировать решаемые задачи несложными схемами, диаграммами, иной графикой и их комбинациями;

оценивать надёжность информации по критериям, предложенным учителем или сформулированным самостоятельно;

эффективно запоминать и систематизировать информацию.

Коммуникативные универсальные учебные действия

Общение:

сопоставлять свои суждения с суждениями других участников диалога, обнаруживать различие и сходство позиций;

публично представлять результаты выполненного опыта (эксперимента, исследования, проекта);

самостоятельно выбирать формат выступления с учётом задач презентации и особенностей аудитории и в соответствии с ним составлять устные и письменные тексты с использованием иллюстративных материалов.

Совместная деятельность (сотрудничество):

понимать и использовать преимущества командной и индивидуальной работы при решении конкретной проблемы, в том числе при создании информационного продукта;

принимать цель совместной информационной деятельности по сбору, обработке, передаче, формализации информации, коллективно строить действия по её достижению: распределять роли, договариваться, обсуждать процесс и результат совместной работы;

выполнять свою часть работы с информацией или информационным продуктом, достигая качественного результата по своему направлению и координируя свои действия с другими членами команды;

оценивать качество своего вклада в общий информационный продукт по критериям, самостоятельно сформулированным участниками взаимодействия;

сравнивать результаты с исходной задачей и вклад каждого члена команды в достижение результатов, разделять сферу ответственности и проявлять готовность к предоставлению отчёта перед группой.

Регулятивные универсальные учебные действия

Самоорганизация:

выявлять в жизненных и учебных ситуациях проблемы, требующие решения;

ориентироваться в различных подходах к принятию решений (индивидуальное принятие решений, принятие решений в группе);

самостоятельно составлять алгоритм решения задачи (или его часть), выбирать способ решения учебной задачи с учётом имеющихся ресурсов и собственных возможностей, аргументировать предлагаемые варианты решений;

составлять план действий (план реализации намеченного алгоритма решения), корректировать предложенный алгоритм с учётом получения новых знаний об изучаемом объекте;

делать выбор в условиях противоречивой информации и брать ответственность за решение.

Самоконтроль (рефлексия):

владеть способами самоконтроля, самомотивации и рефлексии;

давать оценку ситуации и предлагать план её изменения;

учитывать контекст и предвидеть трудности, которые могут возникнуть при решении учебной задачи, адаптировать решение к меняющимся обстоятельствам;

объяснять причины достижения (недостижения) результатов информационной деятельности, давать оценку приобретённому опыту, уметь находить позитивное в произошедшей ситуации;

вносить коррективы в деятельность на основе новых обстоятельств, изменившихся ситуаций, установленных ошибок, возникших трудностей;

оценивать соответствие результата цели и условиям.

Эмоциональный интеллект:

ставить себя на место другого человека, понимать мотивы и намерения другого.

Принятие себя и других:

осознавать невозможность контролировать всё вокруг даже в условиях открытого доступа к любым объёмам информации.

ПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

К концу обучения **в 7 классе** у обучающегося будут сформированы следующие умения:

- соблюдать структуру алгоритма;
- отличать операторы ввода-вывода. применять операторы ввода-вывода в алгоритме.
- использовать переменные;
- применять полученные знания к решению задач.
- создавать и выполнять программы для решения несложных алгоритмических задач в среде Scratch;
- создавать диалог героя с пользователем;
- соблюдать структуру программы;
- приемам применения типов переменных;
- создавать новый сюжет и ставить задачи для Scratch.
- выполнять постановку задачи, анализ программы, составлять словесный алгоритм решения задачи.
- отличать и уметь составлять линейные, разветвляющиеся и циклические конструкции.
- стандартным приёмам составления программы для исполнителя Scratch для решения стандартной задачи с использованием основных алгоритмических конструкций;
- стандартным приемам отладки программ основных конструкций;
- понимать и анализировать программы, написанные для исполнителя Scratch;
- использовать готовые прикладные компьютерные программы по выбранной специализации.
- создавать и выполнять программы для решения несложных алгоритмических задач;
- применять операторы ввода-вывода в программе;
- создавать диалог с пользователем в режиме компиляции;
- соблюдать структуру программы.
- применять стандартные функции к решению задач;
- владеть приемами применения типов переменных;
- составлять программы с использованием конструкций ветвления и циклов.
- писать, вводить, редактировать и выполнять программы для исполнителя Рисователь;
- строить линии и замкнутые фигуры в координатной плоскости.
- строить рекурсивные алгоритмы;
- применять два формата задания цвета линий и цвета заливки;
- владеть понятиями: анимация, процедура, пауза, нажатие клавиши;
- понимать и применять принципы анимации;
- применять принципы анимации при создании анимированных объектов.
- использовать процедуры в алгоритме;
- решать рекурсивные задачи;

- применять принципы анимации при создании анимированных объектов, в том числе с обработкой нажатия клавиши;
- соблюдать правила составления программы для AppInventor;
- уметь загружать задачи и выполнять программы;
- выполнять постановку задачи, анализ программы, составлять словесный алгоритм решения задачи;
- отличать и уметь составлять линейные, разветвляющиеся и циклические конструкции;
- загружать задачи и выполнять программы на Android устройстве;
- выполнять постановку задачи, анализ программы, составлять словесный алгоритм решения задачи;
- отличать и уметь составлять линейные, разветвляющиеся и циклические конструкции.
- создавать новый сюжет для AppInventor;
- создавать новые приложения и ставить задачи для AppInventor

К концу обучения **в 8 классе** у обучающегося будут сформированы следующие умения:

- создавать и выполнять программы на языке программирования Python для решения несложных алгоритмических задач;
- применять операторы ввода-вывода в программе;
- создавать диалог с пользователем в режиме компиляции;
- соблюдать структуру программы;
- применять стандартные функции к решению задач;
- владеть приемами применения типов данных;
- составлять программы с использованием конструкций ветвления и циклов.
- стандартным приёмам написания программы на языке Python для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- стандартным приемам отладки программ основных конструкций;
- понимать и анализировать программы написанные на языке программирования Python;
- применять полученные знания к решению задач;
- использовать готовые прикладные компьютерные программы по выбранной специализации.

К концу обучения **в 9 классе** у обучающегося будут сформированы следующие умения:

- создавать и выполнять программы на языке программирования C++ для решения несложных алгоритмических задач;
- применять операторы ввода-вывода в программе
- создавать диалог с пользователем в режиме компиляции;
- соблюдать структуру программы;
- применять стандартные функции к решению задач;
- владеть приемами применения типов переменных;
- составлять программы с использованием конструкций ветвления и циклов;
- применять принципы анимации при создании анимированных объектов.
- стандартным приёмам написания программы на языке Си++ для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ;
- стандартным приемам отладки программ основных конструкций;
- понимать и анализировать программы написанные на языке программирования Си++;
- применять полученные знания к решению задач.

- использовать готовые прикладные компьютерных программ по выбранной специализации.

Критерии оценки предметных планируемых результатов обучающихся по курсу «Программирование»

Реализация программы предполагает следующие формы организации учебного процесса:

- урочная (учитель объясняет новый материал и консультирует обучающихся в процессе выполнения ими практических заданий на компьютере, ученики выполняют практические и творческие работы под руководством учителя);
- внеурочная (обучающиеся дома или в компьютерном классе самостоятельно выполняют практические задания, проекты, конкурсные работы);
- дистанционно–сетевая (обучающиеся получают информацию и обмениваются результатами работы между собой и с учителем по локальной и глобальной сетям, участвуют в сетевых олимпиадах и конкурсах, проходят обучение на сетевых курсах).

Данный курс имеет выраженную практическую направленность, которая и определяет логику построения материала учебных занятий. Основной формой обучения является практикум.

Основным методом обучения в данном курсе является *метод проектов*. Разработка каждого проекта реализуется в форме выполнения практической работы на компьютере. Кроме выполнения проектов обучающимся предлагаются практические задания для самостоятельного выполнения.

В представленной программе обуславливается возможность повысить результативность обучения информатике и ИКТ при параллельном преподавании школьного основного курса и данного дополнительного.

Оцениваются знания и умения учащихся с учетом их индивидуальных особенностей. Содержание и объем материала, подлежащего проверке, определяется программой.

Текущий контроль знаний осуществляется в форме выполнения практических работ через Интернет на сайте СДО Инженерного лицея НГТУ <http://lyceum.nstu.ru/sdo>. В конце разделов курса каждый учащийся выполняет индивидуальный проект в качестве зачетной работы.

1. Оценка практических и контрольных работ обучающихся

При проверке усвоения материала выявляется полнота, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях. Отметка зависит также от наличия и характера погрешностей, допущенных учащимися.

- *грубая ошибка* – полностью искажено смысловое значение понятия, определения;
- *погрешность* отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;
- *недочет* – неправильное представление об объекте, не влияющего кардинально на знания, определенные программой обучения;
- *мелкие погрешности* – неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные ошибки и т.п.

Эталоном, относительно которого оцениваются знания учащихся, является обязательный минимум содержания информатики и информационных технологий. Требовать от учащихся определения, которые не входят в школьный курс информатики – это, значит, навлекать на себя проблемы, связанные нарушением прав учащегося («Закон об образовании»).

Исходя из норм (пятибалльной системы), заложенных во всех предметных областях выставляется отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1-2 мелких погрешностей;

«4» ставится при наличии 1-2 недочётов или одной ошибки;

«3» ставится при выполнении $\frac{1}{2}$ от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что учащийся не владеет обязательными умениями по данной теме в полной мере (незнание основного программного материала).

Процент выполнения работы	Достигнутый уровень	Отметка
86 –100 %	Работа выполнена на высоком уровне	«5»
71– 85 %	Работа выполнена на среднем уровне	«4»
50 – 70 %	Работа выполнена на базовом уровне	«3»
0 – 49%	Неудовлетворительный уровень выполнения работы	«2»

Оценка индивидуального проекта обучающихся

Оценивание индивидуальных проектов, обучающихся проводится по следующим критериям:

«Отлично» выставляется в том случае, если:

– работа носит практический характер, содержит грамотно изложенную теоретическую базу, характеризуется логичным, последовательным изложением материала с соответствующими выводами и обоснованными предложениями;

– при защите работы обучающийся показывает достаточно глубокие знания вопросов темы, свободно оперирует данными исследованиями, вносит обоснованные предложения, во время выступления использует наглядные пособия (таблицы, схемы, графики, электронные презентации и т.д.) или раздаточный материал, легко отвечает на поставленные вопросы.

«Хорошо» выставляется в том случае, если:

– работа носит практический характер, содержит грамотно изложенную теоретическую базу, характеризуется последовательным изложением материала с соответствующими выводами, однако с не вполне обоснованными предложениями;

– при защите обучающийся показывает знания вопросов темы, оперирует данными исследования, вносит предложения, во время выступления использует наглядные пособия (таблицы, схемы, графики, электронные презентации и т.д.) или раздаточный материал, без особых затруднений отвечает на поставленные вопросы.

«Удовлетворительно» выставляется в том случае, если:

– работа носит практический характер, содержит теоретическую главу, базируется на практическом материале, но отличается поверхностным анализом и недостаточно критическим разбором, в ней просматривается непоследовательность изложения материала, представлены необоснованные предложения;

– при защите обучающийся проявляет неуверенность, показывает слабое знание вопросов темы, не дает полного, аргументированного ответа на заданные вопросы.