

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Муниципальное бюджетное общеобразовательное учреждение города
Новосибирска
"Средняя общеобразовательная школа №3 имени Бориса Богаткова"
МБОУ СОШ № 3

РАБОЧАЯ ПРОГРАММА

учебного предмета «Программирование»

для обучающихся 10– 11 классов

Пояснительная записка

Рабочая программа по учебному курсу «Программирование» разработана на основе:

- Федерального государственного образовательного стандарта среднего общего образования, утвержденного приказом Министерства образования и науки Российской Федерации от 17.05.2012 № 413 (с изменениями и дополнениями);
- Положении «О рабочей программе учебного предмета МБОУ СОШ № 3»

Настоящая рабочая программа является составной частью основной образовательной программы среднего общего образования МБОУ СОШ № 3

При реализации данной программы применяются электронные образовательные ресурсы и могут применяться дистанционные образовательные технологии. На изучение курса выделено 68 часов: 34 часа в 10 классе (1 час в неделю) и 34 часа в 11 классе (1 час в неделю).

Программа данного учебного курса ориентирована на рассмотрение отдельных вопросов информатики, которые входят в содержание единого государственного экзамена. С учетом перехода на Компьютерный ЕГЭ по информатике курс дополняет и развивает школьный курс информатики, а также является информационной поддержкой дальнейшего образования и ориентирован на удовлетворение образовательных потребностей старших школьников, их аналитических и синтетических способностей. Основная идея данного элективного курса заключена в расширении и углублении знаний учащихся по программированию на языке Python по некоторым разделам информатики, в обеспечении прочного и сознательного овладения учащимися системой знаний и умений, необходимых при сдаче выпускного экзамена, а для некоторых школьников - необходимых для продолжения образования.

Следует также отметить, что Всероссийские олимпиады для школьников по информатике и задания Единого государственного экзамена содержат не только задания на программирование (задания 24-27), но и задачи, которые можно решать с использованием программирования (задания 2,4,5,6,8,12,15,16,17,19-23). Таким образом, введение дополнительного 1 ч в неделю на изучение основ алгоритмизации и программирования на языке Python является необходимым и достаточным условием для реализации задачи обучения и воспитания нового поколения, отвечающего по своему уровню развития и образу жизни условиям информационного общества.

Кроме того, изучение основ программирования связано с развитием целого ряда таких умений и навыков (организация деятельности, ее планирование и т.д.), которые носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач современной школы.

В процессе освоения содержания данного курса ученики овладевают новыми знаниями, обогащают свой жизненный опыт, получают возможность практического применения своих интеллектуальных, организаторских способностей, развивают свои коммуникативные способности, овладевают общеучебными умениями.

Освоение предметного содержания курса и сам процесс изучения его становятся средствами, которые обеспечивают переход от обучения учащихся к их самообразованию.

Изучение курса предполагает обеспечение положительной мотивации учащихся на повторение ранее изученного материала, выделение узловых вопросов курса, предназначенных для повторения, использование схем, моделей, опорных конспектов, справочников, компьютерных тестов (в том числе интерактивных), самостоятельное составление (моделирование) тестов аналогичных заданиям КЕГЭ.

Методологической основой предлагаемого курса является деятельностный подход к обучению информатике. Данный подход предполагает обучение не только готовым знаниям, но и деятельности по приобретению этих знаний, способов рассуждений, доказательств. В связи с этим в процессе изучения курса учащимся предлагаются задания, стимулирующие самостоятельное открытие ими фактов, новых, ранее неизвестных, приемов и способов решения задач.

В связи со всем вышеизложенным возникла необходимость введения элективного учебного курса «Основы программирования», цель которого повысить общекультурный уровень учащихся, развить основы программирования, а также обеспечить выпускникам успешное выполнение заданий олимпиад и некоторых заданий КЕГЭ по информатике.

Цели курса:

1. Расширить и углубить знания учащихся по разделу «Основы программирования».
2. Обеспечить индивидуальное и систематическое сопровождение учащихся при подготовке к олимпиадам и КЕГЭ по информатике.

Задачи курса:

1. Расширение и углубление раздела программирование курса информатики.
2. Актуализация, систематизация и обобщение знаний учащихся по программированию.
3. Формирование у учащихся понимания роли полученных знаний, как инструмента позволяющего выбрать лучший вариант действий из многих возможных.
4. Развитие интереса учащихся к изучению программирования.
5. Расширение научного кругозора учащихся.
6. Обучение старшеклассников решению учебных и жизненных проблем, способам анализа информации, получаемой в разных формах.
7. Формирование понятия о методах при решении сложных задач с помощью языка программирования Python.
8. Психологическая подготовка к КЕГЭ.

Организация на занятиях элективного курса должна существенно отличаться от урочной: учащемуся необходимо давать достаточное время на размышление, приветствовать любые попытки самостоятельных рассуждений, выдвижения гипотез, способов решения задач.

В курсе заложена возможность дифференцированного обучения.

Применяются следующие **виды деятельности на занятиях**: обсуждение, тестирование, конструирование тестов, исследовательская деятельность, работа с текстом, диспут, обзорные лекции, мини-лекции, семинары и практикумы по решению задач, предусмотрены консультации.

Методы и формы обучения определяются требованиями ФГОС, с учетом индивидуальных и возрастных особенностей учащихся, развития и саморазвития личности. В связи с этим определены основные приоритеты методики изучения элективного курса:

- обучение через опыт и сотрудничество;
- интерактивность (работа в малых группах, ролевые игры, тренинги, вне занятий - метод проектов);

- личностно-деятельностный и субъект–субъективный подход (больше внимание к личности учащегося, а не целям учителя, равноправное их взаимодействие).

В процессе освоения содержания данного курса ученики овладевают новыми знаниями, обогащают свой жизненный опыт, получают возможность практического применения своих интеллектуальных, организаторских способностей, развивают свои коммуникативные способности, овладевают общеучебными умениями. Освоение предметного содержания курса и сам процесс изучения его становится средствами, которые обеспечивают переход от обучения учащихся к их самообразованию.

Методической основой предлагаемого курса является деятельностный подход к обучению программирования. Данный подход предполагает обучение не только готовым знаниям, но и деятельности по приобретению этих знаний, способов рассуждений, доказательств. В связи с этим в процессе изучения курса учащимся предлагаются задания, стимулирующие самостоятельное открытие ими фактов, новых, ранее неизвестных, приемов и способов решения задач.

В результате целенаправленной учебной деятельности, осуществляемой в формах учебного исследования, учебного проекта, получит дальнейшее развитие способность к информационно-поисковой деятельности: самостоятельному отбору источников информации в соответствии с поставленными целями и задачами. Учащиеся научатся систематизировать информацию по заданным признакам, критически оценивать и интерпретировать информацию. Изучение курса будет способствовать развитию ИКТ-компетентности учащихся.

Практическая значимость курса «Основы программирования» обусловлена тем, что его объектами являются фундаментальные структуры и количественные отношения действительного мира. Развитие мышления учащихся при изучении настоящего курса способствует усвоению предметов гуманитарного цикла. Практические умения и навыки необходимы для дальнейшей жизни, трудовой и профессиональной подготовки школьников.

Развитие у учащихся правильных представлений о роли программирования в научном познании и в практике способствует формированию научного мировоззрения учащихся, а также формированию качеств мышления, необходимых для адаптации в современном информационном обществе.

Требую от учащихся умственных и волевых усилий, концентрации внимания, активности развитого воображения, программирование развивает нравственные черты личности (настойчивость, целеустремленность, творческую активность, самостоятельность, ответственность, трудолюбие, дисциплину и критичность мышления) и умение аргументированно отстаивать свои взгляды и убеждения, а также способность принимать самостоятельные решения.

Изучение курса «Основы программирования» существенно расширяет кругозор учащихся. Активное использование задач на всех этапах учебного процесса развивает творческие способности школьников.

При обучении формируются умения и навыки умственного труда: планирование своей работы, поиск рациональных путей её выполнения, критическая оценка результатов. В процессе обучения школьники должны научиться излагать свои мысли ясно и исчерпывающе, лаконично и ёмко, приобрести навыки чёткого, аккуратного и грамотного выполнения задач на программирование.

Планируемые результаты освоения учебного курса

Изучение курса даёт возможность достижения обучающимися следующих результатов.

Личностные:

Сформированность мировоззрения, соответствующего современному уровню развития науки и техники;

- готовность и способность к образованию, в том числе самообразованию, на

протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;

- сформированные навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;

- эстетическое отношение к миру, включая эстетику научного и технического творчества;

- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

Метапредметные:

- умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать

- деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

- умение продуктивно общаться и взаимодействовать в процессе совместной

- деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;

- владение навыками познавательной, учебно-исследовательской и проектной

- деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

- готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

- умение использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Предметные результаты

учащиеся научатся:

- строить логические выражения с помощью операций дизъюнкции, конъюнкции, отрицания, импликации, эквиваленции; выполнять эквивалентные преобразования этих выражений, используя законы алгебры логики (в частности, свойства дизъюнкции, конъюнкции, правила де Моргана, связь импликации с дизъюнкцией);

- строить таблицу истинности заданного логического выражения; строить логическое выражение в дизъюнктивной нормальной форме по заданной таблице истинности; определять истинность высказывания, составленного из элементарных высказываний с помощью логических операций, если известна истинность входящих в него элементарных высказываний; исследовать область истинности высказывания, содержащего переменные; решать логические уравнения;

- строить дерево игры по заданному алгоритму; строить и обосновывать выигрышную стратегию игры;

- описывать графы с помощью матриц смежности с указанием длин ребер (весовых матриц); решать алгоритмические задачи, связанные с анализом графов, в частности задачу построения оптимального пути между вершинами ориентированного ациклического графа и определения количества различных путей между вершинами;

- понимать основные принципы устройства и функционирования современных стационарных и мобильных компьютеров; выбирать конфигурацию компьютера в соответствии с решаемыми задачами;

- понимать назначение, а также основные принципы устройства и работы современных операционных систем; знать виды и назначение системного программного обеспечения;
- владеть принципами организации иерархических файловых систем и именования файлов; использовать шаблоны для описания группы файлов;
- использовать динамические (электронные) таблицы, в том числе формулы с использованием абсолютной, относительной и смешанной адресации, выделение диапазона таблицы и упорядочивание (сортировку) его элементов; построение графиков и диаграмм;
- владеть основными сведениями о табличных (реляционных) базах данных, их структуре, средствах создания и работы, в том числе выполнять отбор строк таблицы, удовлетворяющих определенному условию; описывать базы данных и средства доступа к ним; наполнять разработанную базу данных;
- использовать компьютерные сети для обмена данными при решении прикладных задач;
- организовывать на базовом уровне сетевое взаимодействие (настраивать работу протоколов сети TCP/IP и определять маску сети);
- понимать структуру доменных имен; принципы IP-адресации узлов сети;
- представлять общие принципы разработки и функционирования интернет-приложений (сайты, блоги и др.);
- проектировать собственное автоматизированное место; следовать основам безопасной и экономичной работы с компьютерами и мобильными устройствами; соблюдать санитарно- гигиенические требования при работе за персональным компьютером в соответствии с нормами действующих СанПиН.
- использовать графы, деревья, списки при описании объектов и процессов окружающего мира; использовать префиксные деревья и другие виды деревьев при решении алгоритмических задач, в том числе при анализе кодов;
- использовать второй язык программирования; сравнивать преимущества и недостатки двух языков программирования;
- создавать программы для учебных или проектных задач средней сложности;
- осознанно подходить к выбору ИКТ-средств и программного обеспечения для решения задач, возникающих в ходе учебы и вне ее, для своих учебных и иных целей;
- использовать пакеты программ и сервисы обработки и представления данных, в том числе – статистической обработки;

Содержание курса

10 класс

Раздел 1. Среда и синтаксис Python. Стандартные и встроенные функции

Знакомство с IDLE Python. Переменные. Построение программы на языке Python. Синтаксис языка Python. Операторы. Типы данных int, float, str. Команды ввода и вывода. Целочисленная арифметика. Высказывания, логические операции, кванторы, истинность высказывания. Логические функции. Законы алгебры логики. Логические уравнения. Дизъюнктивная нормальная форма.

Раздел 2. Основные алгоритмические конструкции. Строки

Формализация понятия алгоритма. Алгоритмические конструкции. Запись алгоритмических конструкций в языке программирования Python. Условный оператор if - elif – else. Циклы в языке программирования Python. Цикл for Функция range. Цикл с предусловием while. Операторы break и continue. Встроенные функции min(), max(), abs(). Оператор in. Строковый тип данных: индексация и срезы. Методы строк. Алгоритмы, связанные с делимостью целых чисел. Алгоритм Евклида для определения НОД двух натуральных чисел.

Алгоритмы линейной (однопроходной) обработки последовательности чисел без использования дополнительной памяти, зависящей от длины последовательности (вычисление максимума, суммы, линейный поиск и т.п.). Алгоритмы анализа символьных строк, в том числе: подсчёт количества появлений символа в строке; разбиение строки на слова по пробельным символам; поиск подстроки внутри данной строки; замена найденной подстроки на другую строку.

Раздел 3. Списки. Кортежи. Матрицы. Множества

Кортежи. Словари. Множества. Основы работы со списками. Методы списков. Сортировка одномерных массивов. Квадратичные алгоритмы сортировки (пример: сортировка пузырьком). Слияние двух отсортированных массивов в один без использования сортировки. Алгоритмы анализа отсортированных массивов. Рекурсивная реализация сортировки массива на основе слияния двух его отсортированных фрагментов. Сложность алгоритма сортировки слиянием. Обработка элементов последовательности, удовлетворяющих определённому условию (вычисление суммы заданных элементов, их максимума и т.п.). Алгоритмы обработки массивов. Примеры: перестановка элементов данного одномерного массива в обратном порядке; циклический сдвиг элементов массива; заполнение двумерного числового массива по заданным правилам; поиск элемента в двумерном массиве; вычисление максимума и суммы элементов двумерного массива. Вставка и удаление элементов в массиве.

Раздел 4. Функции. Файловый ввод и вывод

Функции в программировании. Локальные и глобальные переменные. Процедуры. Рекурсия. Файлы. Работа с файлами. Менеджеры контекста with .. as. Работа с модулями: создание и подключение инструкций import .. from. Подпрограммы. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат. Алгоритмы исследования элементарных функций. Рекурсивные алгоритмы.

Раздел 5. Тренинг по вариантам

Решение варианта КЕГЭ. Разбор ошибок решенного варианта

11 класс

Раздел 1. Алгоритмизация и программирование

История языков программирования. Компиляция и интерпретация. Знакомство с Python и средой программирования IDLE. Цели изучения курса "Основы программирования". Актуализация знаний о языке Python. Условный оператор if - elif – else. Вложенный и каскадный условный оператор. Цикл for. Функция range. Цикл с предусловием while. Операторы break и continue. Программирование циклических алгоритмов на Python. Строковый тип данных: индексация и срезы. Методы строк. Последовательности. Кортежи. Словари. Множества и операции над ними. Списки. Срезы. Основы работы со списками. Методы списков. Вывод элементов списка. Строковые методы split и join. Методы списков. Списочные выражения. Файлы и операции над ними. Модули и пакеты. Обзор стандартной библиотеки. Модули sys, copy, math. Функции. Функции без параметров. Функции с параметрами. Локальные и глобальные переменные. Функции, возвращающие значения. Функции для обработки последовательностей: range, xrange, map, filter. Итераторы. Функции iter, enumerate, sorted. Модуль itertools. Функции itertools.chain, itertools.repeat, itertools.count.

Раздел 2. Программирование задач повышенного и высокого уровня сложности

Логические переменные. Символьные и строковые переменные. Операции над строками. Двумерные массивы (матрицы). Средства работы с данными во внешней памяти. Файлы. Представление о структурах данных. Примеры: списки, словари, деревья, очереди. Структурное программирование. Проверка условия выполнения цикла до начала выполнения тела цикла и после выполнения тела цикла: постусловие и предусловие цикла. Инвариант цикла. Методы проектирования программ «сверху вниз» и «снизу вверх». Разработка программ, использующих подпрограммы. Библиотеки подпрограмм и их использование. Объектно-ориентированное программирование. Объекты и классы. Использование модулей (компонентов) при разработке программ

Раздел 3. Тренинг по вариантам

Решение вариантов КЕГЭ по информатике, с последующим разбором ошибок.

Тематическое планирование

10 класс

№ п/п	Тема	Количество часов
1.	Среда и синтаксис Python. Стандартные и встроенные функции	4
2.	Основные алгоритмические конструкции. Строки	11
3.	Списки. Кортежи. Матрицы. Множества	6
4.	Функции. Файловый ввод и вывод	10
5.	Тренинг по вариантам	3
Итого:		34

11 класс

№ п/п	Тема	Количество часов
1.	Алгоритмизация и программирование	7
2.	Программирование задач повышенного и высокого уровня сложности	17
3.	Тренинг по вариантам	10
Итого:		34

Раздел «Среда и алгоритмические конструкции»

Номер урока	Тема урока	Параграф учебника (номер, название)	Практические работы (номер, название)	Работы компьютерного практикума (источник, номер, название)	Количество часов на тему	Содержание программы
1.	Язык PYTHON. Самая простая программа.	Часть 1. § 14. Первые программы.			1	Повторение: понятие алгоритма, свойства алгоритма, исполнитель алгоритма (виды, основные характеристики), способы записи алгоритма.
2.	Вывод текста на экран.	Часть 1. § 14. Первые программы.	Тест № 1. Оператор вывода.	ПР № 1. Вывод текста на экран.	1	Структура программы. Операторы ввода-вывода. Переменные. Ключевые слова: компилятор, препроцессор, отдельная компиляция, вывод данных, поток вывода, поток ввода, пространство имен, символьная строка.
3.	Диалоговые программы.	Часть 1. § 15. Диалоговые программы.		ПР № 2. Программа «Как тебя зовут?»	1	Диалоговые программы. Переменные и их типы. Сумма чисел.
4.	Переменные и их типы.	Часть 1. § 15. Диалоговые программы.	Тест № 2. Исправление ошибок в диалоговых программах		1	Ключевые слова: ввод данных, переменная, объявление переменной, входной поток.
5.	Сумма чисел.	Часть 1. § 15. Диалоговые программы.		ПР № 3. Сумма чисел.	1	

6.	Ограниченность значений целых чисел.	Часть 1. § 18. Обработка целых чисел.	Тест № 3. Сокращенная запись операций.		1	Предельные значения чисел. Различие между вещественными и целыми числами. Дискретное представление чисел. Программное повышение точности вычислений. Хранение в памяти целых чисел: со знаком, без знака. Повторение: получение дополнительного кода числа. Арифметические операции с целыми числами: сложение и вычитание, умножение и деление. Операция сравнения чисел. Повторение: поразрядные логические операции, сдвиги. Ключевые слова: арифметические выражения, частное, остаток, форматный вывод,
7.	Арифметические выражения.	Часть 1. § 18. Обработка целых чисел.		ПР № 4. Простые вычисления.	1	
8.	Деление и остаток. Вывод данных на экран.	Часть 1. § 18. Обработка целых чисел.	Тест № 4. Операции «/» и «%».	ПР № 5. Применение операций «/» и «%» к решению задач.	1	
9.	Случайные целые числа.	Часть 1. § 18. Обработка целых чисел.		ПР № 6. Случайное целое число на заданном отрезке [a, b].	1	

						случайные числа, зерно.
10.	Вещественные числа в языке PYTHON.	Часть 1. § 19. Обработка вещественных чисел.	Тест № 5. Типы вещественных чисел.		1	Повторение: хранение в памяти вещественных чисел. Нормализованное представление вещественных чисел в
11.	Ввод и вывод.	Часть 1. § 19. Обработка вещественных чисел.		ПР № 7. Формы записи вещественных чисел в	1	

				программах.		компьютере.
12.	Операции с вещественными числами.	Часть 1. § 19. Обработка вещественных чисел.		ПР № 8. Операции с вещественными числами.	1	Арифметические операции с вещественными числами: сложение и вычитание, умножение и деление.
13.	Случайные вещественные числа.	Часть 1. § 19. Обработка вещественных чисел.	Тест № 6. Случайные вещественные числа.	ПР № 9. Случайные вещественные числа.	1	Ключевые слова: вещественное число, научный формат, мантисса, форматный вывод, округление.
14.	Условный оператор.	Часть 1. § 20. Ветвления.	Тест № 7. Ветвления.	ПР № 10. Ветвления.	1	Условный оператор. Полная и неполная формы записи условного оператора. Сложные условия в условном операторе. Множественный выбор.
15.	Вложенный условный оператор.	Часть 1. § 20. Ветвления.		ПР № 11. Вложенные условные операторы.	1	
16.	Логические переменные.	Часть 1. § 20. Ветвления.	Тест № 8. Логические переменные.	ПР № 12. Логические переменные.	1	
17.	Сложные условия.	Часть 1. § 20. Ветвления.	Тест № 9. Сложные условия.	ПР № 13. Сложные условия.	1	
18.	Множественный выбор.	Часть 1. § 20. Ветвления.		ПР № 14. Множественный выбор.	1	
19.	Практикум: использование ветвлений.	Часть 1. § 20. Ветвления.		ПР № 15. Задачи на ветвления.	1	Ключевые слова: условный оператор, полная форма, неполная форма, составной оператор, вложенный условный оператор, логические переменные.
20.	Контрольная работа «Ветвления».	Часть 1. § 20. Ветвления.			1	
21.	Цикл с предусловием.	Часть 1. § 21. Циклы.	Тест № 10. Циклы с предусловием.	ПР № 16. Циклы с предусловием.	1	Понятие циклического алгоритма. Циклы с

22.	Циклы с постусловием.	Часть 1. § 21. Циклы.	Тест № 11. Циклы с постусловием.	ПР № 17. Циклы с постусловием.	1	предусловием. Циклы с постусловием. Циклы с переменными (счетчиком). Вложенные циклы. Ключевые слова: цикл с предусловием, цикл с постусловием, цикл по переменной.
23.	Вычисление квадратного корня.	Часть 1. § 21. Циклы.		ПР № 18. Вычисление квадратного корня.	1	
24.	Циклы по переменной.	Часть 1. § 21. Циклы.	Тест № 12. Циклы по переменной.	ПР № 19. Циклы по переменной.	1	
25.	Вложенные циклы.	Часть 1. § 21. Циклы.		ПР № 20. Вложенные циклы.	1	
26.	Контрольная работа «Циклы».				1	
27.	Библиотека TX Library. Управляем пикселями.	Часть 1. § 16. Компьютерная графика.	Тест № 13. Компьютерная графика.		1	Библиотека TX Library. Управляем пикселями. Линии и фигуры. Замкнутые фигуры. Ключевые слова: окно, координаты, оси координат, пиксель, цвет контура, цвет заливки, код цвета, прозрачный цвет.
28.	Линии и фигуры. Замкнутые фигуры.	Часть 1. § 16. Компьютерная графика.		ПР № 21. Линии и фигуры. Замкнутые фигуры.	1	
29.	Длинная программа.	Часть 1. § 17. Процедуры.		ПР № 22. Процедуры	1	Длинная программа. Рефакторинг. Процедуры с параметрами. Ключевые слова: подпрограмма, процедура, рефакторинг, аргументы, параметры, базовая точка.
30.	Рефакторинг.	Часть 1. § 17. Процедуры.		ПР № 23. Рефакторинг.	1	
31.	Процедуры с параметрами.	Часть 1. § 17. Процедуры.		ПР № 24. Процедуры с параметрами.	1	

32.	Принципы анимации. Рисуем шарик. Начальное положение.	Часть 1. § 22. Анимация.		ПР № 25. Рисуем шарик.	1	Принципы анимации. Рисуем шарик. Начальное положение. Анимация движения.
33.	Анимация движения.	Часть 1. § 22. Анимация.		ПР № 26. Анимация движения.	1	Обработка нажатия клавиши. Обработка нажатия клавиши. Ключевые слова: анимация, процедура, пауза, нажатие клавиши.
34	Творческая работа					

Предметные результаты

- Владение *универсальным языком программирования высокого уровня PYTHON*, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.
- Владение умением *понимать программы*, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня; знанием основных конструкций программирования; умением анализировать алгоритмы с использованием таблиц.
- Владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов.
- Овладение понятием *сложности алгоритма*, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки.

11 класс

Модуль «Алгоритмизация и программирование» (34 часа)

Номер урока	Тема урока	Параграф учебника (номер, название)	Практические работы (номер, название)	Работы компьютерного практикума (источник, номер, название)	Количество часов на тему	Содержание программы
1.	Метод пузырька (Сортировка простыми обменами).	Часть 3. § 13. Простые алгоритмы сортировки.		ПР № 1. Метод пузырька (Сортировка простыми обменами).	1	Сортировка в массивах. Методы сортировки: метод пузырька, метод выбора, сортировка вставками.
2.	Метод выбора.	Часть 3. § 13. Простые		ПР № 2. Метод	1	Массивы в ₁

	Сортировка вставками.	алгоритмы сортировки.		выбора. Сортировка вставками.		подпрограмма х. Ключевые слова: сортировка, метод пузырька, сортировка вставками,
3.	Массивы в подпрограммах.	Часть 3. § 13. Простые алгоритмы сортировки.	Тест № 1. Массивы в подпрограммах.		1	
						перестановка элементов, сложность алгоритма.
4.	Сортировка слиянием.	Часть 3. § 14. Быстрые алгоритмы сортировки и поиска.		ПР № 3. Массивы в подпрограммах.	1	Сортировка в массивах. Методы сортировки: сортировка слиянием,
5.	Быстрая сортировка.	Часть 3. § 14. Быстрые алгоритмы сортировки и поиска.		ПР № 4. Быстрая сортировка.	1	быстрая сортировка. Стандартная сортировка на языке
6.	Стандартная сортировка на языке PYTHON.	Часть 3. § 14. Быстрые алгоритмы сортировки и поиска.		ПР № 5. Стандартная сортировка на языке PYTHON.	1	PYTHON. Двоичный поиск. Ключевые слова:
7.	Двоичный поиск.	Часть 3. § 14. Быстрые алгоритмы сортировки и поиска.		ПР № 6. Двоичный поиск.	1	сортировка слиянием, быстрая сортировка, двоичный поиск, лямбда-функция.
8.	Принцип сэндвича. Файловые потоки.	Часть 3. § 15. Обработка файлов.	Тест № 2. Принцип сэндвича. Файловые потоки.		1	Понятие файла. Типы файлов. Этапы работы с файлами:

9.	Неизвестное количество данных. Обработка массивов. Чтение файлов по словам.	Часть 3. § 15. Обработка файлов.	Тест № 3. Обработка массивов.	ПР № 7. Чтение файлов по словам.	1	открытие файла, запись в файл, удаление записей из файла, чтение из файла, закрытие файла.
10.	Построчная обработка файлов. Передача имени файла программе.	Часть 3. § 15. Обработка файлов.		ПР № 8. Построчная обработка файлов. Передача имени файла программе.	1	Обработка массивов, записанных в файле. Обработка строк, записанных в файле. Обработка смешанных данных, записанных в файле.
						Ключевые слова: файл, файловый поток, открытие файла, закрытие файла, чтение из файла, запись в файл, конец файла, аргументы командной строки.
11.	Решето Эратосфена.	Часть 3. § 16. Целочисленные алгоритмы.		ПР № 9. Решето Эратосфена.	1	Целочисленные алгоритмы. Решето Эратосфена.
12.	«Длинные» числа в PYTHON.	Часть 3. § 16. Целочисленные алгоритмы.		ПР № 10. «Длинные» числа в PYTHON.	1	Длинные числа. Ключевые слова: решето Эратосфена, длинные числа, разряды, перенос.

13.	Динамические массивы в PYTHON.	Часть 3. § 17. Динамические массивы и словари.	Тест № 4. Динамические массивы в PYTHON.		1	Понятие динамического массива. Объявление и заполнение динамического массива. Использование динамических массивов в подпрограммах. Расширение массива. Принципы работы динамического массива. Тип vector из библиотеки STL. Итераторы. Словари. Перебор элементов словаря.
14.	Тип vector из библиотеки STL.	Часть 3. § 17. Динамические массивы и словари.		ПР № 11. Тип vector из библиотеки STL.	1	
15.	Итераторы. Применение.	Часть 3. § 17. Динамические массивы и словари.		ПР № 12. Итераторы. Применение.	1	
16.	Словари. Перебор элементов словаря.	Часть 3. § 17. Динамические массивы и словари.		ПР № 13. Алфавитно-частотный словарь.	1	
						словаря. Ключевые слова: динамический массив, указатель, контейнер, вектор, словарь, ключ, значение, перебор элементов, сортировка по ключу, итератор.
17.	Структуры в PYTHON. Обращение к полям структуры.	Часть 3. § 18. Структуры.	Тест № 5. Обращение к полям структуры.	ПР № 14. Ввод и вывод структур.	1	Понятие структуры. Объявление структур. Обращение к полю структуры. Работа с файлами. Сортировка.
18.	Работа с файлами.	Часть 3. § 18. Структуры.	Тест № 6. Работа с файлами.	ПР № 15. Чтение структур из файла.	1	

19.	Сортировка.	Часть 3. § 18. Структуры.		ПР № 16. Сортировка структур.	1	Ключевые слова: структура, поле, точечная запись, сортировка, ключ.
20.	Стек.	Часть 3. § 19. Стек, очередь, дек.		ПР № 17. Функции для работы со стеком из библиотеки STL.	1	Понятие стека. Использование динамического массива для создания стека. Вычисление арифметических выражений. Скобочные выражения. Очередь и дека как разновидности и линейного списка. Ключевые слова: стек, очередь, дек.
21.	Очередь. Хранение очереди в массиве.	Часть 3. § 19. Стек, очередь, дек.		ПР № 18. Тип моделирования очереди queue.	1	
22.	Дек.	Часть 3. § 19. Стек, очередь, дек.		ПР № 19. Тип моделирования дека queue.	1	
23.	Деревья в PYTHON.	Часть 3. § 20. Деревья.	Тест № 7. Деревья в PYTHON.		1	Дерево. Двоичное дерево. Дерево поиска. Обход двоичного дерева. Вычисление арифметических выражений. Использование связанных структур. Хранение двоичного
24.	Обходы дерева.	Часть 3. § 20. Деревья.	Тест № 8. Обходы дерева.	ПР № 20. Обходы дерева.	1	
25.	Деревья поиска.	Часть 3. § 20. Деревья.		ПР № 21. Деревья поиска.	1	
26.	Вычисления арифметических выражений.	Часть 3. § 20. Деревья.		ПР № 22. Вычисления арифметических выражений.	1	

27.	Дерево в массиве. Модульность.	Часть 3. § 20. Деревья.		ПР № 23. Дерево в массиве. Модульность.	1	дерева в массиве. Ключевые слова: дерево, двоичное дерево, обход в глубину, обход в ширину, дерево поиска, ключ, хранение в массиве, модуль, проект, заголовочный файл.
28.	Графы в PYTHON.	Часть 3. § 21. Графы.	Тест № 9. Графы.		1	Понятие графа.
29.	Задача коммивояжера.	Часть 3. § 21. Графы.		ПР № 24. Задача коммивояжера.	1	«Жадные» алгоритмы. Алгоритм Прима-Крускала.
30.	Задача коммивояжера: жадный алгоритм.	Часть 3. § 21. Графы.		ПР № 25. Задача коммивояжера: жадный алгоритм.	1	Поиск кратчайших путей в графе. Алгоритм Дейкстры.
31.	Задача коммивояжера: случайные перестановки.	Часть 3. § 21. Графы.		ПР № 26. Задача коммивояжера: случайные перестановки.	1	Алгоритм Флойда-Уоршелла. Ключевые слова: жадный алгоритм,
32.	Как избавиться	Часть 3. § 21. Графы.		ПР № 27. Передача	1	остовное дерево, задача

	от глобальных переменных Передача данных по ссылке.			данных по ссылке.		коммиво яжёра , Гамиль тонов цикл, полны й перебо р, рек урс ия, хво стов ая рек урс ия, слу чай ный пои ск, глобал ьные переме нные, структ уры.
33, 34	Итоговый урок				1	