

**Муниципальное бюджетное общеобразовательное учреждение
«Центр образования № 4» г. Тулы**

ПРИНЯТА:
на заседании педагогического совета
МБОУ ЦО № 4
Протокол № 1 от 31.08.2020 г.

УТВЕРЖДАЮ:
Директор МБОУ ЦО № 4
_____ Е.Ю. Степанов
«31» августа 2020 г.

ПРОГРАММА

Внеурочной деятельности «Программирование – это просто»

для 10 классов

Составил: Углова М.А.

Квалификация: высшая

Составлена на основе федерального государственного образовательного стандарта
основного общего образования

Программа рассмотрена
на заседании методического объединения
учителей математики, информатики и технологии
МБОУ ЦО № 4
Протокол № 1 от «24» августа 2020 г.
Руководитель: Галыгина Г.В.

_____ *подпись*

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа по информатике составлена в соответствии с требованиями Федерального государственного образовательного стандарта основного общего образования (ФГОС ООО); примерной основной образовательной программы основного общего образования.

Рабочая программа реализуется в течение одного года в 10 классе за 1 ч. в неделю.

1. Планируемые результаты освоения учебного предмета.

– Личностные результаты в сфере отношений обучающихся к себе, к своему здоровью, к познанию себя:

– ориентация обучающихся на достижение личного счастья, реализацию позитивных жизненных перспектив, инициативность, креативность, готовность и способность к личностному самоопределению, способность ставить цели и строить жизненные планы;

– принятие и реализация ценностей здорового и безопасного образа жизни, бережное, ответственное и компетентное отношение к собственному физическому и психологическому здоровью;

– Личностные результаты в сфере отношений обучающихся с окружающими людьми:

– развитие компетенций сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности.

– Личностные результаты в сфере отношений обучающихся к окружающему миру, живой природе, художественной культуре:

– мировоззрение, соответствующее современному уровню развития науки, значимости науки, готовность к научно-техническому творчеству, владение достоверной информацией о передовых достижениях и открытиях мировой и отечественной науки, заинтересованность в научных знаниях об устройстве мира и общества;

– 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;

– Личностные результаты в сфере отношения обучающихся к труду, в сфере социально-экономических отношений:

- осознанный выбор будущей профессии как путь и способ реализации собственных жизненных планов;
- готовность обучающихся к трудовой профессиональной деятельности как к возможности участия в решении личных, общественных, государственных, общенациональных проблем;
- потребность трудиться, уважение к труду и людям труда, трудовым достижениям, добросовестное, ответственное и творческое отношение к разным видам трудовой деятельности;
- готовность к самообслуживанию, включая обучение и выполнение домашних обязанностей.
- Планируемые метапредметные результаты освоения программы:
- Регулятивные универсальные учебные действия
- Выпускник научится:
- самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута;
- оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали;
- ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;
- оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели;
- выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты;
- организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели;
- сопоставлять полученный результат деятельности с поставленной заранее целью.
- 2. Познавательные универсальные учебные действия
- Выпускник научится:

- искать и находить обобщенные способы решения задач, в том числе, осуществлять развернутый информационный поиск и ставить на его основе новые (учебные и познавательные) задачи;
- критически оценивать и интерпретировать информацию с разных позиций, распознавать и фиксировать противоречия в информационных источниках;
- использовать различные модельно-схематические средства для представления существенных связей и отношений, а также противоречий, выявленных в информационных источниках;
- находить и приводить критические аргументы в отношении действий и суждений другого; спокойно и разумно относиться к критическим замечаниям в отношении собственного суждения, рассматривать их как ресурс собственного развития;
- выходить за рамки учебного предмета и осуществлять целенаправленный поиск возможностей для широкого переноса средств и способов действия;
- выстраивать индивидуальную образовательную траекторию, учитывая ограничения со стороны других участников и ресурсные ограничения;
- менять и удерживать разные позиции в познавательной деятельности.
- Коммуникативные универсальные учебные действия
- Выпускник научится:
 - осуществлять деловую коммуникацию как со сверстниками, так и со взрослыми (как внутри образовательной организации, так и за ее пределами), подбирать партнеров для деловой коммуникации исходя из соображений результативности взаимодействия, а не личных симпатий;
 - при осуществлении групповой работы быть как руководителем, так и членом команды в разных ролях (генератор идей, критик, исполнитель, выступающий, эксперт и т.д.);
 - координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия;
 - развернуто, логично и точно излагать свою точку зрения с использованием адекватных (устных и письменных) языковых средств;

– распознавать конфликтогенные ситуации и предотвращать конфликты до их активной фазы, выстраивать деловую и образовательную коммуникацию, избегая личностных оценочных суждений.

Предметные результаты

– формализовать понятие «алгоритм» с помощью одной из универсальных моделей вычислений (машина Тьюринга, машина Поста и др.); понимать содержание тезиса Черча–Тьюринга;

– понимать и использовать основные понятия, связанные со сложностью вычислений (время работы и размер используемой памяти при заданных исходных данных; асимптотическая сложность алгоритма в зависимости от размера исходных данных); определять сложность изучаемых в курсе базовых алгоритмов;

– анализировать предложенный алгоритм, например определять, какие результаты возможны при заданном множестве исходных значений и при каких исходных значениях возможно получение указанных результатов;

– создавать, анализировать и реализовывать в виде программ базовые алгоритмы, связанные с анализом элементарных функций (в том числе приближенных вычислений), записью чисел в позиционной системе счисления, делимостью целых чисел; линейной обработкой последовательностей и массивов чисел (в том числе алгоритмы сортировки), анализом строк, а также рекурсивные алгоритмы;

– применять метод сохранения промежуточных результатов (метод динамического программирования) для создания полиномиальных (не переборных) алгоритмов решения различных задач; примеры: поиск минимального пути в ориентированном ациклическом графе, подсчет количества путей;

– создавать собственные алгоритмы для решения прикладных задач на основе изученных алгоритмов и методов;

– применять при решении задач структуры данных: списки, словари, деревья, очереди; применять при составлении алгоритмов базовые операции со структурами данных;

– использовать основные понятия, конструкции и структуры данных последовательного программирования, а также правила записи этих конструкций и структур в выбранном для изучения языке программирования;

– использовать в программах данные различных типов; применять стандартные и собственные подпрограммы для обработки символьных строк; выполнять обработку данных, хранящихся в виде массивов различной размерности; выбирать тип цикла в зависимости от решаемой подзадачи; составлять циклы с использованием заранее определенного инварианта цикла; выполнять базовые операции с текстовыми и двоичными файлами; выделять подзадачи, решение которых необходимо для решения поставленной задачи в полном объеме; реализовывать решения подзадач в виде подпрограмм, связывать подпрограммы в единую программу; использовать модульный принцип построения программ; использовать библиотеки стандартных подпрограмм;

– применять алгоритмы поиска и сортировки при решении типовых задач;

– выполнять объектно-ориентированный анализ задачи: выделять объекты, описывать на формальном языке их свойства и методы; реализовывать объектно-ориентированный подход для решения задач средней сложности на выбранном языке программирования;

– выполнять отладку и тестирование программ в выбранной среде программирования; использовать при разработке программ стандартные библиотеки языка программирования и внешние библиотеки программ; создавать многокомпонентные программные продукты в среде программирования;

– устанавливать и деинсталлировать программные средства, необходимые для решения учебных задач по выбранной специализации;

– пользоваться навыками формализации задачи; создавать описания программ, инструкции по их использованию и отчеты по выполненным проектным работам;

– разрабатывать и использовать компьютерно-математические модели; анализировать соответствие модели реальному объекту или процессу; проводить эксперименты и статистическую обработку данных с помощью компьютера; интерпретировать результаты, получаемые в ходе моделирования реальных процессов; оценивать числовые параметры моделируемых объектов и процессов;

Выпускник получит возможность:

2. *использовать знания о методе «разделяй и властвуй»;*
3. *приводить примеры различных алгоритмов решения одной задачи, которые имеют различную сложность; использовать понятие переборного алгоритма;*
4. *использовать понятие универсального алгоритма и приводить примеры алгоритмически неразрешимых проблем;*
5. *использовать второй язык программирования; сравнивать преимущества и недостатки двух языков программирования;*
6. *создавать программы для учебных или проектных задач средней сложности..*

7. Содержание учебного предмета, курса.

Алгоритмы и элементы программирования

Алгоритмы и структуры данных

– Алгоритмы исследования элементарных функций, в частности – точного и приближенного решения квадратного уравнения с целыми и вещественными коэффициентами, определения экстремумов квадратичной функции на отрезке.

– Алгоритмы анализа и преобразования записей чисел в позиционной системе счисления.

– 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.). Обработка элементов последовательности, удовлетворяющих определенному 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.*

– Рекурсивные алгоритмы, в частности: нахождение натуральной и целой степени заданного ненулевого вещественного числа; вычисление факториалов; вычисление n-го элемента рекуррентной последовательности (например, последовательности Фибоначчи). Построение и анализ дерева рекурсивных вызовов. Возможность записи рекурсивных алгоритмов без явного использования рекурсии.

– 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.

– Алгоритмы анализа символьных строк, в том числе: подсчет количества появлений символа в строке; разбиение строки на слова по пробельным символам; поиск подстроки внутри данной строки; замена найденной подстроки на другую строку.

– Построение графика функции, заданной формулой, программой или таблицей значений.

– Алгоритмы приближенного решения уравнений на данном отрезке, например, методом деления отрезка пополам. Алгоритмы приближенного вычисления длин и площадей, в том числе: приближенное вычисление длины плоской кривой путем аппроксимации ее ломаной; приближенный подсчет методом трапеций площади под графиком функции, заданной формулой, программой или таблицей значений. *Приближенное вычисление площади фигуры методом Монте-Карло. Построение траекторий, заданных разностными схемами. Решение задач оптимизации. Алгоритмы вычислительной геометрии. Вероятностные алгоритмы.*

– Сохранение и использование промежуточных результатов. Метод динамического программирования.

– Представление о структурах данных. Примеры: списки, словари, деревья, очереди. *Хэш-таблицы.*

Языки программирования

– Подпрограммы (процедуры, функции). Параметры подпрограмм. Рекурсивные процедуры и функции.

- Логические переменные. Символьные и строковые переменные. Операции над строками.
- Двумерные массивы (матрицы). *Многомерные массивы.*
- Средства работы с данными во внешней памяти. Файлы.
- Подробное знакомство с одним из универсальных процедурных языков программирования. Запись алгоритмических конструкций и структур данных в выбранном языке программирования. Обзор процедурных языков программирования.
- *Представление о синтаксисе и семантике языка программирования.*
- *Понятие о непроцедурных языках программирования и парадигмах программирования. Изучение второго языка программирования.*

Разработка программ

- Этапы решения задач на компьютере.
- 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.
- Интегрированная среда разработки программы на выбранном языке программирования. Пользовательский интерфейс интегрированной среды разработки программ.
- Понятие об объектно-ориентированном программировании. Объекты и классы. *Инкапсуляция, наследование, полиморфизм.*
- Среды быстрой разработки программ. Графическое проектирование интерфейса пользователя. Использование модулей (компонентов) при разработке программ.

Элементы теории алгоритмов

- Формализация понятия алгоритма. Машина Тьюринга – пример абстрактной универсальной вычислительной модели. Тезис Чёрча–Тьюринга.

– Другие универсальные вычислительные модели (пример: машина Поста).
Универсальный алгоритм. Вычислимые и невычислимые функции. Проблема остановки и ее неразрешимость.

– Абстрактные универсальные порождающие модели (пример: грамматики).

– Сложность вычисления: количество выполненных операций, размер используемой памяти; их зависимость от размера исходных данных. Сложность алгоритма сортировки слиянием (MergeSort).

– Примеры задач анализа алгоритмов: определение входных данных, при которых алгоритм дает указанный результат; определение результата алгоритма без его полного пошагового выполнения.

– Доказательство правильности программ.

3. Тематическое планирование и содержание учебного курса по темам (учебник К.Ю. Поляков. Программирование)

№	Тема	10 кл.
	Алгоритмы и элементы программирования	34

Календарно-тематическое планирование

Описание раздела	Тема урока	Кол-во часов	Дата проведения
Алгоритмы и элементы программирования	Первые программы. Диалоговые программы	1	1. неделя
	Компьютерная графика	1	2. неделя
	Процедуры	1	3. неделя
	Обработка целых чисел	1	4. неделя
	Обработка вещественных чисел	1	5. неделя
	Случайные и псевдослучайные числа	1	6. неделя
	Ветвления	1	7. неделя
	Сложные условия	1	8. неделя
	Циклы с условием	1	9. неделя
	Анимация	1	10. неделя
	Циклы по переменной	1	11. неделя
	Циклы в компьютерной графике	1	12. неделя
	Проектирование программ	1	13. неделя
	Процедуры	1	14. неделя
	Рекурсия	1	15. неделя
	Функции	1	16. неделя
	Символьные строки	1	17. неделя
	Массивы (списки)	1	18. неделя
	Алгоритмы обработки массивов	1	19. неделя
	Поиск в массивах	1	20. неделя
	Используем массивы	1	21. неделя
	Матрицы	1	22. неделя
	Сложность алгоритмов	1	23. неделя
	Простые алгоритмы сортировки	1	24. неделя
	Быстрые алгоритмы сортировки	1	25. неделя

	Двоичный поиск	1	26. неделя
	Обработка файлов	1	27. неделя
	Целочисленные алгоритмы	1	28. неделя
	Словари	1	29. неделя
	Стек, очередь, дек	1	30. неделя
	Деревья	1	31. неделя
	Графы	1	32. неделя
	Динамическое программирование	1	33. неделя
	Игровые модели	1	34. неделя