УТВЕРЖДЕНА
решением Попечительского совета
УДО «Школа им. Хинксона»
Протокол № 47 от «20» августа 2026 г.

Председатель Попечительского совета _____ /_____/

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Информатика, алгоритмизация и программирование на английском языке. Повышенный уровень»

Направленность: техническая

Разработано УДО «Школа им. Хинксона»

Москва

1. Паспорт программы

Параметр	Характеристика
Наименование программы	«Информатика, алгоритмизация и программирование на английском языке. Повышенный уровень»
Вид программы	дополнительная общеобразовательная общеразвивающая программа
Направленность	техническая
Уровень	повышенный
Язык реализации	английский
Форма обучения	очная
Организация занятий	индивидуально и/или в группе в соответствии с расписанием школы
Срок освоения	один учебный год
Объем	290 академических часов
Контактная работа	145 академических часов
Самостоятельная работа	145 академических часов
Недельная нагрузка	10 академических часов: 5 контактных + 5 самостоятельных
Язык(и) программирования и программные средства	определяются педагогом в соответствии с целями, содержанием и уровнем программы
Итоговый контроль	комплексная практическая работа и/или программный проект с защитой результата

2. Нормативная и методическая основа

Программа разработана как дополнительная общеобразовательная общеразвивающая программа с учетом действующего законодательства Российской Федерации и локальных актов УДО «Школа им. Хинксона».

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» — в применимой части;
- устав, правила внутреннего распорядка обучающихся, положения об организации образовательного процесса, формах контроля, охране здоровья, обработке персональных данных и иные локальные акты УДО «Школа им. Хинксона».

3. Пояснительная записка

3.1. Направленность и назначение

Программа имеет техническую направленность и предназначена для углубленного освоения основ информатики, алгоритмизации и программирования на английском языке. Основное

внимание уделяется не пользовательской работе с компьютером, а формированию алгоритмического мышления, навыков разработки программ, анализа данных, тестирования, отладки и проектирования небольших программных решений.

3.2. Актуальность

Алгоритмическое мышление и программирование являются универсальными инструментами решения задач в науке, инженерии, экономике и цифровой среде. Обучение на английском языке дополнительно формирует способность использовать профессиональную терминологию, читать технические материалы и аргументированно объяснять разработанные решения.

3.3. Цель

Сформировать у обучающихся устойчивые навыки алгоритмического решения задач и практического программирования, обеспечивающие самостоятельную разработку, тестирование, анализ и представление небольших программных продуктов на английском языке.

3.4. Задачи

- сформировать представление об информации, способах ее кодирования и представления;
- освоить системы счисления, основы математической логики и логические выражения;
- научить анализировать условие задачи и строить алгоритм решения;
- освоить основные конструкции используемого языка программирования: переменные, типы данных, ветвления, циклы, функции и базовые структуры данных;
- сформировать навыки декомпозиции, повторного использования кода и выбора подходящей структуры данных;
- освоить базовые алгоритмы поиска, сортировки и анализа эффективности решений;
- сформировать навыки работы со строками, списками, словарями, файлами и небольшими наборами данных;
- развить навыки тестирования, отладки, чтения и анализа программного кода;
- познакомить с базовыми принципами объектно-ориентированного подхода;
- сформировать представление о корректности, безопасности и ответственном использовании цифровых технологий;
- развить умение планировать и реализовывать индивидуальные программные проекты и объяснять принятые решения на английском языке.

4. Категория обучающихся и условия приема

Программа рассчитана на обучающихся школьного возраста, как правило 14–18 лет, заинтересованных в углубленном изучении информатики и программирования и способных работать с учебными материалами на английском языке.

Рекомендуемый исходный уровень: базовое понимание работы с цифровыми устройствами, элементарные математические навыки, владение английским языком на уровне, достаточном для понимания инструкций преподавателя и учебных материалов. Предварительный опыт программирования желателен, но не является обязательным при достаточной готовности к интенсивной учебной нагрузке.

До начала обучения может проводиться входная диагностика для определения исходного уровня и планирования индивидуальной траектории. Результаты диагностики используются прежде всего для методического планирования и не являются самостоятельным основанием для отказа в обучении, если иное не предусмотрено локальными актами и условиями договора.

5. Срок освоения, объем и режим занятий

Срок освоения: один учебный год.

Объем программы: 290 академических часов, в том числе 145 часов контактной работы и 145 часов самостоятельной работы.

Базовая недельная нагрузка: 10 академических часов, в том числе 5 контактных и 5 самостоятельных часов.

Конкретные сроки реализации программы определяются учебным календарем и расписанием УДО «Школа им. Хинксона».

Продолжительность и чередование занятий, перерывы, распределение нагрузки в течение недели и иные организационные параметры устанавливаются расписанием школы с учетом возраста обучающихся, санитарных требований и локальных актов.

6. Форма обучения, язык и организация образовательного процесса

Форма обучения — очная. Занятия проводятся индивидуально и/или в группе. Язык реализации программы — английский. Русский язык может использоваться эпизодически для организационных разъяснений, если это необходимо для безопасности и понимания задания, но основная учебная коммуникация, терминология, инструкции и представление результатов осуществляются на английском языке.

При реализации программы используются один или несколько языков программирования и программные средства, соответствующие содержанию курса, возрасту обучающихся и учебным целям. Конкретный язык программирования и среда разработки определяются педагогом и могут уточняться без изменения основных параметров программы.

7. Планируемые результаты

По завершении программы обучающийся способен:

- анализировать условие задачи, выделять входные данные, ограничения и ожидаемый результат;
- разрабатывать алгоритм решения и представлять его в виде описания, псевдокода, блок-схемы или программного кода;
- использовать переменные, типы данных, арифметические и логические выражения;
- применять условные конструкции и циклы для реализации алгоритмов;
- создавать и использовать функции, параметры и возвращаемые значения;
- применять строки, списки/массивы, словари и другие базовые структуры данных;
- читать, анализировать и объяснять программный код соответствующего уровня;

- выявлять синтаксические, логические и граничные ошибки и исправлять их;
- составлять тестовые случаи и проверять корректность программы на типичных и граничных данных;
- выбирать подходящий алгоритмический подход для поиска, сортировки и обработки данных;
- объяснять на доступном уровне различия в эффективности простых алгоритмов и использовать базовые представления об алгоритмической сложности;
- читать и записывать данные в файлы и выполнять обработку небольших наборов данных;
- применять базовые принципы объектно-ориентированного подхода при решении подходящих задач;
- декомпозировать задачу на подзадачи и организовывать программный код в понятную структуру;
- создавать законченные небольшие программы и мини-проекты по заданным требованиям;
- соблюдать требования корректности, безопасности, академической добросовестности и ответственного использования цифровых технологий;
- объяснять алгоритм, программное решение, результаты тестирования и принятые решения на английском языке.

8. Учебный план

Раздел	Контактные	Самостоятельные	Всего	Форма контроля
1. Информация и представление данных	9	9	18	Практические задания, краткий контроль
2. Системы счисления и кодирование данных	10	10	20	Задачи, практическая работа
3. Логические выражения и математическая логика	9	9	18	Алгоритмические задачи, тест
4. Алгоритмы и способы их представления	9	9	18	Практическая работа
5. Основы программирования: переменные, типы данных и выражения	14	14	28	Программные задания
6. Условные конструкции и циклы	14	14	28	Программные задания, мини-контроль

Раздел	Контактные	Самостоятельные	Всего	Форма контроля
7. Функции, процедуры и декомпозиция	13	13	26	Практическая работа
8. Строки, списки, словари и базовые структуры данных	14	14	28	Программные задания
9. Файлы и обработка данных	9	9	18	Практическая работа с данными
10. Поиск, сортировка и основы алгоритмической сложности	11	11	22	Алгоритмические задачи, практическая работа
11. Основы объектно-ориентированного подхода	8	8	16	Мини-проект
12. Тестирование, отладка, корректность и цифровая безопасность	8	8	16	Отладочная практическая работа
13. Итоговый программный проект / комплексная практическая работа	17	17	34	Итоговый контроль и защита
ИТОГО	145	145	290	

9. Содержание программы

Раздел 1. Информация и представление данных

Понятие информации и данных. Формы представления информации. Двоичное представление данных. Единицы измерения информации. Текстовые, числовые и графические данные на концептуальном уровне. Точность, полнота и корректность данных. Практика: преобразование представлений, анализ примеров кодирования, задания на оценку объема данных.

Раздел 2. Системы счисления и кодирование данных

Позиционные системы счисления. Двоичная, десятичная и шестнадцатеричная системы. Перевод чисел между системами. Двоичное представление целых чисел на доступном уровне.

Символьное кодирование, Unicode как практический пример. Практика: преобразования и задачи на кодирование.

Раздел 3. Логические выражения и математическая логика

Логические значения, операции AND, OR, NOT, сравнения. Таблицы истинности. Составные условия. Приоритет операций. Применение логики при построении алгоритмов и проверке условий. Практика: преобразование условий, анализ логических выражений, задачи на истинность.

Раздел 4. Алгоритмы и способы их представления

Понятие алгоритма, свойства алгоритмов, исполнитель, вход и выход. Последовательность, ветвление, повторение. Псевдокод, блок-схемы и программный код. Декомпозиция задачи. Практика: построение алгоритмов до программной реализации, анализ корректности шагов.

Раздел 5. Основы программирования: переменные, типы данных и выражения

Синтаксис используемого языка программирования. Переменные, присваивание, числовые и строковые типы, логические значения. Ввод и вывод, арифметические операции, преобразование типов, выражения. Практика: короткие программы, вычислительные задачи, чтение и исправление кода.

Раздел 6. Условные конструкции и циклы

if/elif/else. Вложенные условия. Циклы for и while. Счетчики и накопители. Диапазоны, вложенные циклы. Условия завершения и типичные ошибки циклов. Практика: алгоритмические задачи, генерация последовательностей, обработка наборов значений.

Раздел 7. Функции, процедуры и декомпозиция

Назначение функций. Определение и вызов функции, параметры, аргументы, возвращаемые значения, область видимости на базовом уровне. Декомпозиция и повторное использование кода. Документирование функций. Практика: переработка монолитных программ в набор функций, модульные тестовые примеры.

Раздел 8. Строки, списки, словари и базовые структуры данных

Индексация и срезы. Методы строк. Списки, добавление, удаление, перебор, вложенные списки. Словари и пары ключ-значение. Выбор структуры данных в зависимости от задачи. Практика: обработка текстов, таблиц значений, частот, небольших наборов записей.

Раздел 9. Файлы и обработка данных

Открытие, чтение и запись текстовых файлов. Структурированные данные на базовом уровне. Разделители, строки и поля. Проверка качества входных данных. Практика: чтение набора данных из файла, вычисление показателей, фильтрация и формирование результата.

Раздел 10. Поиск, сортировка и основы алгоритмической сложности

Линейный поиск, бинарный поиск при соблюдении условий применимости. Простые алгоритмы сортировки и использование встроенных средств языка. Сравнение количества операций и времени выполнения на доступном уровне. Понятие алгоритмической сложности без

формального углубления. Практика: реализация и сравнение вариантов алгоритма, обоснование выбора.

Раздел 11. Основы объектно-ориентированного подхода

Объект, класс, атрибут, метод. Создание простых классов, экземпляры и взаимодействие объектов. Когда объектное представление делает программу понятнее. Практика: моделирование небольшой предметной области, мини-проект.

Раздел 12. Тестирование, отладка, корректность и цифровая безопасность

Синтаксические и логические ошибки. Трассировка и диагностический вывод. Тестовые случаи, граничные значения, ожидаемый и фактический результат. Проверка входных данных. Читаемость кода и базовые правила безопасной работы с файлами и пользовательскими данными. Ответственное использование цифровых инструментов и внешних источников. Практика: поиск дефектов в готовом коде, исправление и документирование тестов.

Раздел 13. Итоговый программный проект / комплексная практическая работа

Выбор или получение задачи. Анализ требований. Планирование решения, декомпозиция, разработка алгоритма, реализация программы, тестирование, исправление ошибок и подготовка представления результата. Итоговый продукт должен демонстрировать самостоятельное применение основных конструкций языка, структур данных и приемов разработки. Защита проводится на английском языке в объеме, соответствующем уровню обучающегося.

10. Самостоятельная работа обучающихся

Самостоятельная работа является обязательной содержательной частью программы и по объему равна контактной работе. Она направлена преимущественно на практическое закрепление и расширение навыков, а не на пассивное изучение теории.

- написание и доработка программ по заданным требованиям;
- решение алгоритмических задач различного уровня сложности;
- анализ, исправление и улучшение готового программного кода;
- составление тестовых данных и проверка программ;
- работа с небольшими наборами данных и файлами;
- подготовка кратких объяснений алгоритмов и решений на английском языке;
- выполнение индивидуальных практических заданий и мини-проектов;
- поэтапная работа над итоговым программным проектом или комплексной практической работой;
- подготовка к текущему и итоговому практическому контролю.

Объем и сложность самостоятельных заданий дифференцируются с учетом возраста, исходной подготовки и индивидуального темпа освоения материала.

11. Контроль и оценивание

11.1. Текущий контроль

Текущий контроль проводится в практических формах и может включать:

- короткие программные задания;
- алгоритмические задачи;
- анализ программного кода;
- поиск и исправление ошибок;
- тестирование программ;
- практические работы с данными;
- мини-проекты;
- устное или письменное объяснение выбранного решения на английском языке.

Текущий контроль используется для обратной связи и корректировки обучения. Конкретная шкала фиксации результатов определяется локальными актами школы и может предусматривать балльную, процентную, уровневую или иную систему.

11.2. Итоговый контроль

Итоговый контроль проводится в форме комплексной практической работы и/или программного проекта. Обучающийся должен продемонстрировать полный цикл решения задачи:

1. анализирует поставленную задачу и требования;
2. разрабатывает алгоритм и структуру решения;
3. реализует решение программными средствами;
4. подготавливает и выполняет тестирование;
5. выявляет и исправляет обнаруженные ошибки;
6. представляет результат и объясняет принятые решения на английском языке.

11.3. Критерии итоговой оценки

Критерий	Показатели	Вес
Анализ задачи и постановка решения	корректно выделены входные данные, ограничения, требуемый результат; выбран обоснованный способ решения	15%
Алгоритм и декомпозиция	решение логично разбито на этапы/функции; алгоритм соответствует задаче и не содержит существенных логических противоречий	20%
Программная реализация	код выполняет требуемые функции, корректно использует основные конструкции языка и подходящие структуры данных	25%

Критерий	Показатели	Вес
Тестирование и отладка	подготовлены содержательные тесты, включая граничные случаи; найденные ошибки исправлены или объяснены	15%
Качество кода и работа с данными	код читаем, именование и структура понятны; входные данные обрабатываются корректно; соблюдены базовые требования безопасности	10%
Объяснение и защита на английском языке	обучающийся способен объяснить алгоритм, ключевые фрагменты кода, результаты тестирования и ограничения решения	15%
ИТОГО		**100%**

Рекомендуемая интерпретация результата при использовании 100-балльной шкалы: 85–100 — высокий уровень; 70–84 — повышенный уровень; 55–69 — достаточный уровень; менее 55 — требуется доработка. Школа вправе установить иную шкалу локальным актом при сохранении указанных критериев содержательной оценки.

12. Организационно-педагогические условия реализации

12.1. Кадровые условия

Реализацию программы осуществляет педагог, обладающий компетенциями в области информатики, алгоритмизации и программирования, владеющий английским языком на уровне, достаточном для ведения занятий и оценивания учебных работ, и соответствующий установленным законодательством и локальными актами требованиям к педагогическим работникам.

12.2. Материально-техническое обеспечение

- учебное помещение, соответствующее применимым санитарным и противопожарным требованиям;
- компьютер или ноутбук для каждого обучающегося либо иная организация рабочего места, обеспечивающая индивидуальную практику;
- рабочее место преподавателя и средство демонстрации материалов;
- доступ к локальной сети и/или сети Интернет при необходимости и в соответствии с правилами информационной безопасности школы;
- установленная среда программирования, соответствующая используемому языку и задачам программы;
- средства хранения учебных файлов и резервного копирования в соответствии с правилами школы;

- при необходимости — средства ограничения доступа к нежелательному контенту и иные технические меры защиты.

12.3. Требования к безопасной организации работы

При организации занятий учитываются требования охраны здоровья обучающихся, эргономики рабочего места, допустимой продолжительности непрерывной работы с экранными устройствами, регламентированных перерывов и смены видов деятельности. Обработка персональных данных, использование сетевых сервисов и публикация учебных проектов осуществляются в соответствии с законодательством и локальными актами школы.

13. Методическое обеспечение

Обучение строится на сочетании кратких объяснений, демонстрации решений, управляемой практики, самостоятельного программирования, разбора ошибок, парного/группового обсуждения и проектной работы. Не менее половины контактного времени целесообразно использовать для непосредственной практической деятельности обучающихся.

- problem-based learning: освоение понятий через решение содержательных задач;
- пошаговое усложнение заданий от отдельных конструкций к интегрированным программам;
- worked examples с последующим изменением и самостоятельным восстановлением решения;
- code tracing и чтение кода до написания собственного решения;
- debugging practice как самостоятельный вид учебной деятельности;
- формирующее оценивание и короткая обратная связь по коду;
- мини-проекты и итоговый проект с промежуточными контрольными точками;
- дифференциация сложности задач и объема подсказок в зависимости от исходного уровня.

14. Учебные и информационные ресурсы

При реализации программы используются учебные материалы, подготовленные педагогом и школой, документация используемых языков программирования и программных средств, справочные материалы по алгоритмам и структурам данных, наборы практических задач, примеры программного кода и учебные наборы данных.

Рекомендуемые категории ресурсов:

- официальная документация используемых языков программирования и программных средств, а также встроенные справочные системы;
- учебные пособия по основам программирования и алгоритмизации, соответствующие возрасту обучающихся;
- локальные методические материалы школы: инструкции, карточки задач, шаблоны проектов, критерии оценивания;
- безопасные учебные среды и репозитории примеров кода, одобренные школой;
- небольшие открытые или специально подготовленные наборы данных, не содержащие избыточных персональных данных.

Конкретный перечень цифровых платформ, внешних сервисов и учебных ресурсов утверждается или допускается к использованию школой с учетом условий лицензирования, защиты персональных данных, информационной безопасности и применимых ограничений.

15. Календарная организация реализации

Программа разработана для многократного применения и не содержит привязки к конкретному учебному году. Последовательность разделов может уточняться педагогом в пределах общего содержания программы с учетом уровня группы, при условии сохранения общего объема 290 академических часов и достижения заявленных планируемых результатов.

Конкретные сроки реализации программы определяются учебным календарем и расписанием УДО «Школа им. Хинксона».

16. Итоговые положения

Настоящая программа является самостоятельной дополнительной общеобразовательной общеразвивающей программой УДО «Школа им. Хинксона». Изменение учебного календаря, праздничных дней и конкретных дат начала или окончания занятий само по себе не требует изменения содержания или повторного утверждения программы, если сохраняются срок освоения, общий объем, заявленные результаты и иные существенные параметры.

Изменения в общий объем, срок освоения, направленность, существенные планируемые результаты или форму итогового контроля оформляются в порядке, установленном локальными актами школы.

Приложение 1. Сводные параметры программы

Показатель	Значение
Срок освоения	один учебный год
Общий объем	290 академических часов
Контактные часы	145 академических часов
Самостоятельные часы	145 академических часов
Недельная нагрузка	10 академических часов (5 + 5)
Язык реализации	английский
Форма обучения	очная
Организация занятий	индивидуально и/или в группе
Язык программирования	определяется педагогом в соответствии с содержанием программы
Среда программирования	локальная IDE/редактор или иная среда, одобренная школой
Итоговый контроль	комплексная практическая работа и/или программный проект с защитой