

Управление образования администрации
Татарского муниципального района
муниципальное бюджетное общеобразовательное
учреждение – лицей г. Татарска
структурное подразделения - Центр цифрового образования детей «IT-куб»

Рассмотрена на заседании
педагогического совета
«27» сентября 2026г.,
протокол № 1

УТВЕРЖДАЮ
Директор МБОУ - ЛИЦЕЯ
И.А. Чернова
Приказ № 62/1/ОД от 27.08.2026



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

технической направленности
«Программирование на Java»

Возраст обучающихся: 13-17 лет

Срок реализации программы: 1 год

Автор-составитель программы:
Моисеенко Никита Сергеевич,
педагог дополнительного образования

Татарск, 2026

Раздел 1. Комплекс основных характеристик программы

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Java» имеет техническую направленность.

Актуальность программы

Актуальность программы «Программирование на Java» заключается в том, что язык Java является фундаментом мировой корпоративной разработки и экосистемы Android, и благодаря этому владение им открывает перед подростками широкие перспективы для будущего профессионального самоопределения в IT-сфере. В настоящее время цифровая экономика испытывает острую потребность в квалифицированных инженерах-программистах, а школа, в силу ограниченности базовой программы, не дает углубленных знаний по объектно-ориентированному программированию. Кроме того, наблюдается разрыв между школьными знаниями по информатике и реальными требованиями IT-рынка, а также недостаток практико-ориентированных курсов, формирующих навыки написания промышленного кода у школьников.

Отличительные особенности программы

Данная программа разработана с акцентом на глубокое изучение объектно-ориентированного программирования (ООП) с первых занятий, в то время как большинство курсов для школьников ограничиваются процедурным программированием на Python или Scratch.

Отличительные черты программы: модульность построения тем, позволяющая гибко перестраивать траекторию обучения в зависимости от уровня группы; введение в программирование через визуализацию работы классов и сравнение Java с языками C++/Python для расширения кругозора. Основные идеи, которые придают программе своеобразие: плавный переход от консольных приложений к полноценным графическим интерфейсам (Swing/JavaFX) и ознакомление с системой контроля версий Git на уровне командной разработки итогового проекта.

Целевая группа

Программа адресована обучающимся среднего и старшего школьного возраста — 13–17 лет. Данный возрастной период характеризуется активным формированием абстрактно-логического мышления, стремлением к самоопределению, профессиональной ориентации и поиску своего места в социуме. Половозрастной состав группы смешанный, без ограничений по полу. Состояние здоровья: дети с I и II группой здоровья, допущенные к работе с компьютерной техникой; при наличии медицинских показаний предусмотрена адаптация рабочего места согласно СанПиН.

Потребности и интересы целевой группы: познавательный интерес к

устройству цифровых технологий и созданию программных продуктов; потребность в интеллектуальной самореализации и решении сложных алгоритмических задач; интерес к командным формам работы и соревновательной деятельности; стремление получить практический навык, который можно применить уже сейчас (создание игры, приложения, веб-сервиса).

Ожидаемые эффекты от освоения программы: повышение уровня цифровой грамотности; сформированность навыков объектно-ориентированного программирования как базы для дальнейшего профессионального обучения; умение работать в команде; пополнение портфолио проектами для поступления в профильные вузы; готовность к участию в олимпиадах по информатике и конкурсах технической направленности.

Объём и срок освоения программы

Объём программы - 144 часа в год.

Программа рассчитана на 1 год обучения.

Форма обучения: очная.

Уровень программы – базовый.

Особенности организации образовательного процесса

Общая оценка соответствия планируемого содержания программы имеющимся в ОО условиям ее реализации является положительной.

Программа обеспечена необходимыми кадровыми ресурсами: педагог имеет профильное образование и практические навыки программирования на Java. Материально-техническая база включает учебный класс на 10–12 мест с доступом к сети Интернет, оснащенный необходимым ПО (IntelliJ IDEA, JDK, Git) и мультимедийным оборудованием. Организационно занятия проводятся во второй половине дня по расписанию, в очной форме, с возможностью применения дистанционных технологий. Набор в группы осуществляется на добровольной основе.

Имеющиеся условия полностью соответствуют целям и задачам программы и позволяют достичь планируемых результатов в полном объеме.

Режим занятий, периодичность и продолжительность занятий

Занятия проводятся два раза в неделю продолжительностью 2 академических часа (1 академический час равен 45 минутам). Перерыв между учебными занятиями 10-15 минут.

Количество обучающихся в одной группе: 10-12 человек.

1.2. Цель и задачи программы

Цель: формирование устойчивых навыков объектно-ориентированного программирования на языке Java и развитие алгоритмического мышления у обучающихся для успешного применения полученных знаний в проектной деятельности, олимпиадах и профессиональном самоопределении.

Задачи программы:

Обучающие: сформировать знания основ синтаксиса языка Java (переменные, типы данных, операторы, условные конструкции, циклы, массивы); познакомить с принципами объектно-ориентированного программирования (инкапсуляция, наследование, полиморфизм); обучить работе в среде разработки IntelliJ IDEA и использованию средств отладки кода; сформировать навыки работы с основными структурами данных (ArrayList, HashMap) и обработки исключений; познакомить с основами создания графических интерфейсов с использованием библиотек Swing/JavaFX.

Развивающие: развивать алгоритмическое и логическое мышление через решение практических и олимпиадных задач; формировать навыки самостоятельной работы с технической документацией и поиска информации в сети Интернет; формировать навыки командной работы, распределения ролей и коллективной разработки проектов.

Воспитательные: формировать ответственное отношение к результатам своего труда и качеству выполняемой работы; воспитывать культуру оформления программного кода и соблюдение стандартов разработки; способствовать развитию коммуникативных навыков через участие в обсуждении решений и публичной защите проектов; воспитывать навыки здорового образа жизни, соблюдение санитарно-гигиенических норм при работе за компьютером; ориентировать обучающихся на осознанный выбор будущей профессии в сфере информационных технологий.

1.3. Содержание программы **Учебно-тематический план**

№ п/п	Название раздела	Количество часов			Формы промежуточной аттестации/ контроля
		Всего	Из них		
			Теория	Практика	
1	Введение в программирование. Установка окружения.	6	4	2	Практическая работа
2	Основы синтаксиса Java. Алгоритмизация.	24	10	14	Тестирование, самостоятельная работа
3	Объектно- ориентированное программирование (ООП).	26	10	16	Практическая работа, зачётное занятие
4	Стандартные библиотеки. Коллекции.	16	6	10	Самостоятельная работа, тестирование

	Исключения.				
5	Графические интерфейсы (Swing/JavaFX).	24	8	16	Практическая работа, защита творческой работы
6	Основы алгоритмов и структур данных.	20	8	12	Тестирование, самостоятельная работа
7	Итоговый проектный модуль.	28	6	22	Защита проекта
	Итого:	144	52	92	

Содержание программы

1. Введение в программирование. Установка окружения.

Теория: Техника безопасности. Понятие алгоритма и программы. Классификация языков программирования. Платформа Java: история, особенности, JVM. Установка JDK, настройка переменных окружения. Знакомство со средой IntelliJ IDEA: интерфейс, создание проекта. Структура первой программы. Метод `main()`. Система вывода `System.out.println()`. Компиляция и запуск.

Практика: Установка и настройка JDK и IntelliJ IDEA. Создание первого проекта. Написание и запуск Hello, World!. Настройка рабочего пространства (тема, шрифт, автодополнение).

2. Основы синтаксиса Java. Алгоритмизация.

Теория: Структура программы. Комментарии. Переменные: объявление, инициализация. Примитивные типы данных. Правила именования. Арифметические, логические операторы, операторы сравнения. Приоритет операторов. Преобразование типов. Условные конструкции: `if-else`, `switch-case`. Тернарный оператор. Циклы: `while`, `do-while`, `for`. Операторы `break`, `continue`. Массивы: одномерные и двумерные. Методы: объявление, параметры, возвращаемое значение. Перегрузка методов. Область видимости переменных.

Практика: Решение задач на арифметические вычисления. Разработка программ с ветвлением (четность, сравнение, квадратное уравнение). Программы с циклами (таблица умножения, факториал, сумма ряда). Работа с массивами (заполнение, вывод, поиск максимума, транспонирование). Написание методов. Решение задач: простые числа, числа Фибоначчи, алгоритм Евклида.

3. Объектно-ориентированное программирование (ООП).

Теория: Парадигма ООП. Класс и объект. Поля и методы.

Конструкторы: по умолчанию, параметризованный, перегрузка. Ключевое слово `this`. Модификаторы доступа (`public`, `private`, `protected`). Инкапсуляция: геттеры и сеттеры. Статические поля и методы (`static`). Константы (`final`). Наследование (`extends`), `super()`. Переопределение методов (`@Override`). Полиморфизм. Абстрактные классы и методы. Интерфейсы (`implements`). Пакеты и импорт.

Практика: Создание класса «Прямоугольник». Иерархия «Животные» (базовый класс, подклассы, переопределение методов). Иерархия «Транспорт». Иерархия «Сотрудники» (расчет зарплаты). Реализация интерфейсов `Comparable`, `Comparator`. Консольное приложение «Библиотека».

4. Коллекции. Исключения. Ввод-вывод.

Теория: Пакет `java.util`. Интерфейс `Collection`. Иерархия коллекций: `List` (`ArrayList`, `LinkedList`), `Set` (`HashSet`, `TreeSet`), `Map` (`HashMap`, `TreeMap`). Итераторы. Цикл `for-each`. Иерархия исключений. Обработка исключений: `try-catch-finally`. `try-with-resources`. Генерация исключений (`throw`, `throws`). Создание собственных исключений. Пакет `java.io`. `File`, `FileReader`, `FileWriter`, `BufferedReader`, `Scanner`.

Практика: Приложение «Список дел» (`ArrayList`). Телефонный справочник (`HashMap`). Обработка ошибок ввода (деление на ноль, ввод строки). Чтение и запись текстовых файлов. Консольное приложение «Записная книжка» с сохранением в файл.

5. Графические интерфейсы (Swing/JavaFX).

Теория: Графические библиотеки Java: `Swing` и `JavaFX`. Компоненты: `JFrame`, `JPanel`. Менеджеры компоновки (`FlowLayout`, `BorderLayout`, `GridLayout`). Обработка событий: `ActionListener`, `MouseListener`, `KeyListener`. Анонимные классы и лямбда-выражения. Элементы управления: `JButton`, `JLabel`, `JTextField`, `JTextArea`, `JList`, `JComboBox`. Меню (`JMenuBar`, `JMenu`, `JMenuItem`). Диалоги (`JOptionPane`, `JFileChooser`). Рисование (`paintComponent()`). Таймеры (`Timer`). Обзор `JavaFX`.

Практика: Приложение «Калькулятор». Игра «Угадай число». Графический редактор «Рисовалка» (цвет, кисть, очистка). Игра «Сапер» (упрощенная версия). Приложение «Конвертер валют». Анимация с `Timer`.

6. Алгоритмы и структуры данных.

Теория: Понятие алгоритма. Оценка сложности (`Big O`). Сортировки: пузырьковая, вставками, выбором, быстрая, слиянием. Поиск: линейный и бинарный. Рекурсия: стек вызовов. Жадные алгоритмы. Структуры данных: `Stack`, `Queue`. Обзор бинарных деревьев.

Практика: Реализация сортировок. Сравнение времени работы. Бинарный поиск. Рекурсивные задачи (факториал, Фибоначчи, обход файлов).

Жадные алгоритмы. Решение задач на соревновательных платформах.

7. Итоговый проектный модуль.

Теория: Жизненный цикл ПО. Этапы разработки. Техническое задание. Проектирование. Распределение ролей в команде. Основы Git (init, add, commit, push, pull, clone). Работа с удаленным репозиторием. Ветки (branch, merge). Java Code Conventions. Оформление документации и презентации.

Практика: Настройка репозитория. Разработка командного проекта («Банковское приложение», «Справочник», «Тетрис», «Интернет-магазин», «Библиотека», «Телефонный справочник»). Распределение задач. Написание кода. Тестирование и отладка. Оформление документации. Подготовка презентации. Защита проекта.

1.4. Планируемые результаты

Обучающие результаты:

- Знание основных понятий и принципов ООП (класс, объект, инкапсуляция, наследование, полиморфизм).
- Знание синтаксиса Java: типы данных, операторы, условные конструкции, циклы, массивы, методы.
- Владение навыками написания и отладки программ в IntelliJ IDEA.
- Умение создавать консольные приложения и программы с графическим интерфейсом (Swing/JavaFX).
- Владение навыками работы с коллекциями (ArrayList, HashMap) и обработкой исключений.
- Знание базовых алгоритмов сортировки и поиска.
- Владение основами работы с Git.

Развивающие результаты:

- Развитие критического и творческого мышления, умения анализировать, сравнивать, обобщать.
- Овладение универсальными учебными действиями: целеполагание, планирование, самоконтроль, рефлексия.
- Развитие коммуникативных навыков: умение вести диалог, распределять роли в совместной деятельности, эффективно взаимодействовать со сверстниками.
- Развитие регулятивных навыков: умение ставить цели, преодолевать трудности, оценивать свои действия.

Воспитательные результаты:

- Сформированность мотивации к познанию и творчеству, ценностное отношение к знаниям.
- Формирование культуры общения и уважительного отношения к мнению других.

- Развитие таких качеств, как трудолюбие, ответственность, самостоятельность, умение работать в команде.

Раздел 2. Комплекс организационно-педагогических условий

2.1. Календарный учебный график

№ п/п занятия	Тема занятия	Форма занятия	Количество часов	Форма контроля
Раздел 1. Введение в программирование. Установка окружения – 6 ч				
1	Техника безопасности. Понятие алгоритма и программы. Классификация языков программирования.	Вводное занятие	2	Проблемная беседа
2	Платформа Java: история, особенности, JVM. Установка JDK, настройка переменных окружения.	Лекция	2	Опрос
3	Знакомство со средой IntelliJ IDEA. Структура первой программы. Компиляция и запуск. Написание и запуск Hello, World!.	Практикум	2	Практическая работа
Раздел 2. Основы синтаксиса Java. Алгоритмизация – 24 ч				
	Структура программы. Комментарии. Переменные и примитивные типы данных. Правила именования.	Лекция	2	Проблемная беседа
	Арифметические, логические операторы, операторы сравнения. Преобразование типов. Решение задач на вычисления.	КЗ	2	Самостоятельная работа
	Условные конструкции: if-else, switch-case. Тернарный оператор. Решение задач с ветвлением.	КЗ	2	Практическая работа
	Циклы while, do-while, for. Операторы break,	КЗ	4	Самостоятельная работа

	continue. Решение задач на циклические алгоритмы.			
	Массивы: одномерные. Заполнение, вывод, поиск элементов. Решение задач.	КЗ	2	Практическая работа
	Двумерные массивы. Транспонирование матрицы. Решение задач.	КЗ	4	Самостоятельная работа
	Методы: объявление, параметры, возвращаемое значение. Решение задач.	КЗ	2	Практическая работа
	Перегрузка методов. Область видимости переменных. Решение задач.	КЗ	2	Самостоятельная работа
	Решение задач на простые числа, числа Фибоначчи, алгоритм Евклида.	Практикум	2	Самостоятельная работа
	Обобщающее занятие по разделу «Основы синтаксиса». Решение комплексных задач.	П	2	Тестирование

Раздел 3. Объектно-ориентированное программирование (ООП) – 26 ч

	Парадигма ООП. Класс и объект. Поля и методы. Конструкторы. Ключевое слово this.	Лекция	2	Проблемная беседа
	Модификаторы доступа. Инкапсуляция: геттеры и сеттеры. Создание класса «Прямоугольник».	КЗ	2	Практическая работа
	Статические поля и методы (static). Константы (final). Решение задач.	КЗ	2	Самостоятельная работа
	Наследование (extends), super(). Переопределение методов (@Override). Иерархия «Животные».	КЗ	4	Практическая работа

	Полиморфизм. Абстрактные классы и методы. Иерархия «Транспорт».	КЗ	2	Практическая работа
	Интерфейсы (implements). Реализация Comparable, Comparator.	КЗ	4	Самостоятельная работа
	Пакеты и импорт. Разработка консольного приложения «Библиотека» .	Практикум	2	Практическая работа
	Разработка консольного приложения «Библиотека».	Практикум	2	Практическая работа
	Иерархия «Сотрудники» (расчет зарплаты с переопределением методов).	КЗ	2	Самостоятельная работа
	Решение задач на ООП. Подготовка к зачётному занятию.	Практикум	2	Самостоятельная работа
	Зачётное занятие по разделу «ООП»	П	2	Зачётное занятие
Раздел 4. Коллекции. Исключения. Ввод-вывод– 16 ч				
	Пакет java.util. Иерархия коллекций: List (ArrayList, LinkedList).	Лекция	2	Проблемная беседа
	Приложение «Список дел» с использованием ArrayList.	Практикум	2	Практическая работа
	Интерфейсы Set (HashSet, TreeSet), Map (HashMap, TreeMap). Телефонный справочник.	КЗ	2	Практическая работа
	Итераторы. Цикл for-each. Решение задач с коллекциями.	КЗ	2	Самостоятельная работа
	Иерархия исключений. Обработка исключений: try-catch-finally	Лекция	2	Проблемная беседа
	Генерация и проброс исключений (throw, throws). Создание	КЗ	2	Самостоятельная работа

	собственных исключений.			
	Пакет java.io. Чтение и запись текстовых файлов (FileReader, FileWriter, Scanner).	КЗ	2	Практическая работа
	Консольное приложение «Записная книжка» с сохранением в файл.	Практикум	2	Самостоятельная работа
Раздел 5. Графические интерфейсы (Swing/JavaFX) – 24 ч				
	Введение в графические библиотеки Java. Компоненты JFrame, JPanel. Менеджеры компоновки.	Лекция	2	Проблемная беседа
	Обработка событий: ActionListener, MouseListener, KeyListener. Анонимные классы и лямбда-выражения.	Лекция	2	Опрос
	Элементы управления: JButton, JLabel, JTextField, JTextArea. Создание окна.	КЗ	2	Практическая работа
	Приложение «Калькулятор» (интерфейс).	Практикум	2	Практическая работа
	Приложение «Калькулятор» (логика).	Практикум	2	Практическая работа
	Списки (JList) и выпадающие списки (JComboBox). Приложение «Конвертер валют».	КЗ	2	Практическая работа
	Меню (JMenuBar, JMenu, JMenuItem). Диалоги (JOptionPane, JFileChooser).	КЗ	2	Самостоятельная работа
	Рисование (paintComponent()). Игра «Угадай число» с графическим	КЗ	2	Практическая работа

	интерфейсом.			
	Таймеры (Timer). Анимация. Создание движущегося объекта.	КЗ	2	Самостоятельная работа
	Графический редактор «Рисовалка» (цвет, кисть, очистка).	Практикум	2	Практическая работа
	Игра «Сапер» (упрощенная версия)	Практикум	2	Практическая работа
	Игра «Сапер» (упрощенная версия). Защита творческой работы.	П	2	Защита творческой работы
Раздел 6. Алгоритмы и структуры данных – 20 ч				
	Понятие алгоритма. Оценка сложности (Big O).	Лекция	2	Проблемная беседа
	Сортировки: пузырьковая, вставками, выбором. Реализация и сравнение.	КЗ	2	Самостоятельная работа
	Эффективные сортировки: быстрая и слиянием. Сравнение времени работы.	КЗ	2	Самостоятельная работа
	Поиск: линейный и бинарный. Реализация.	КЗ	2	Практическая работа
	Рекурсия: стек вызовов. Рекурсивные задачи (факториал, Фибоначчи, обход файлов).	КЗ	2	Самостоятельная работа
	Жадные алгоритмы. Решение задач.	КЗ	2	Практическая работа
	Структуры данных: Stack, Queue. Обзор бинарных деревьев.	Лекция	2	Опрос
	Решение задач на соревновательных платформах.	Практикум	4	Самостоятельная работа
	Обобщающее занятие по разделу «Алгоритмы и структуры данных».	П	2	Тестирование

Раздел 7. Итоговый проектный модуль – 28 ч				
	Жизненный цикл ПО. Этапы разработки. Техническое задание.	Лекция	2	Проблемная беседа
	Основы Git: init, add, commit, push, pull, clone. Настройка репозитория.	КЗ	2	Практическая работа
	Ветки (branch, merge). Java Code Conventions. Оформление документации.	Лекция	2	Опрос
	Выбор темы проекта. Распределение ролей в команде. Разработка ТЗ.	ПР	2	Проектная деятельность
	Проектирование архитектуры. Создание UML-схем.	ПР	2	Проектная деятельность
	Разработка проекта	ПР	10	Проектная деятельность
	Оформление документации. Подготовка презентации проекта.	ПР	2	Проектная деятельность
	Репетиция защиты. Ответы на вопросы.	ПР	2	Проектная деятельность
	Защита проекта	Защита проекта	2	Защита проекта
	Анализ выполненных проектов. Итоги курса.	Итоговое занятие	2	Проблемная беседа

Календарный учебный график заполнен с помощью условных обозначений:

КЗ – комбинированные занятия, сочетающие элементы теории и практики;

П – проверочное занятие;

ПР – проектная работа (работа над кейсами).

2.2. Условия реализации программы

Материально-техническое обеспечение

– персональные компьютеры (на каждого обучающегося) с программным обеспечением, с минимальными системными требованиями (процессор Intel

Core i3 или аналог AMD, 8 Гб оперативной памяти, 50 Гб свободного дискового пространства);

- экран, проектор или интерактивная доска для демонстрации учебного материала;

- доступ к сети Интернет (скорость не менее 10 Мбит/с) для работы с репозиториями и соревновательными платформами;

- установленная на каждый компьютер среда разработки IntelliJ IDEA (Community Edition — бесплатная версия) или Eclipse IDE;

- установленный на каждый компьютер Java Development Kit (JDK) версии 17 или выше;

- установленная на каждый компьютер система контроля версий Git;

- зарегистрированные аккаунты обучающихся и педагога на платформе GitHub или GitLab для хранения проектов;

- браузер последней версии для работы с документацией, соревновательными платформами и онлайн-ресурсами;

- доступ к официальной документации Java (Java API Documentation) в формате офлайн или онлайн;

- текстовый редактор для быстрого просмотра файлов конфигураций.

Информационное обеспечение

Специальные компьютерные программы:

- среда разработки IntelliJ IDEA (Community Edition);

- Java Development Kit (JDK) версии 17+;

- система контроля версий Git;

- клиент для работы с репозиториями (GitHub Desktop или встроенный в IDEA);

- браузер для работы с онлайн-ресурсами;

- офисный пакет для работы с документацией (LibreOffice).

Информационные технологии:

- использование облачных сервисов для хранения и обмена файлами (Яндекс Диск);

- применение системы дистанционного обучения (при необходимости) для размещения материалов и обратной связи;

- использование онлайн-платформ для решения задач (соревновательные платформы);

- использование официальной документации Java API в онлайн и офлайн-форматах;

- использование сервисов для совместной работы над кодом (GitHub/GitLab).

Кадровое обеспечение

1. Требования к кадровому обеспечению деятельности Центра «IT-куб» определяются образовательной организацией самостоятельно с учетом действующего трудового законодательства.

2. Образовательную деятельность по дополнительным

общеобразовательным программам на базе Центра «ИТ-куб» осуществляют педагоги дополнительного образования. В соответствии с пунктом 4 статьи 46 Федерального закона от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» к занятию педагогической деятельностью по дополнительным общеобразовательным программам допускаются лица, обучающиеся по образовательным программам высшего образования по специальностям и направлениям подготовки, соответствующим направленности дополнительных общеобразовательных программ, и успешно прошедшие промежуточную аттестацию не менее чем за два года обучения. Соответствие образовательной программы высшего образования направленности дополнительной общеобразовательной программы определяется образовательной организацией.

2.3. Формы аттестации

Формы оценки уровня достижений обучающегося

Для контроля и самоконтроля за эффективностью обучения применяются методы:

- предварительные (наблюдение, входной тест);
- текущие (наблюдение, решение задач на занятии, проверка практических работ, самостоятельные работы);
- промежуточные (тестирование, практическая работа, зачётное занятие);
- итоговые (защита проекта).

Формы фиксации образовательных результатов

Для фиксации образовательных результатов в рамках курса используются:

- портфолио работ обучающихся;
- отзывы обучающихся по итогам занятий и итогам обучения.

Формы предъявления и демонстрации образовательных результатов:

- презентация проекта.

Формы подведения итогов реализации программы:

- педагогическое наблюдение;
- педагогический анализ выполнения обучающимися учебных заданий;
- активность обучающихся на занятиях.

2.4. Оценочные материалы

Входной контроль – начальная диагностика в форме опроса «Основы алгоритмики» с целью определения уровня по изучаемой программе (Приложение 1). Анализ результатов входного контроля проводится с помощью следующей шкалы:

<i>Оценка параметров</i>	<i>Уровень по сумме баллов</i>
Начальный уровень	7-14 баллов
Средний уровень	14-21 баллов
Высокий уровень	21-31 баллов

Текущий контроль (оценка усвоения изучаемого материала) осуществляется педагогом на протяжении всей программы в форме педагогического наблюдения, анализа опросов, выполненных практических заданий.

Итоговый контроль осуществляется в виде оценки продукта обучающегося, разработанного в рамках выполнения итогового проекта с применением ранее полученных знаний и навыков на основе сформулированных критериев (Приложение 2).

3. Список литературы

1. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации (с изменениями и дополнениями, вступившими в силу с 01.01.2021)
2. Концепция развития дополнительного образования детей до 2030 года (утверждена распоряжением Правительства РФ от 31 марта 2022 г. № 678-р)
3. Приказ Министерства просвещения Российской Федерации от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
4. Постановление Главного государственного санитарного врача РФ от 28.09.2020 №28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи.
5. Барабанов, А. В. Java. Полное руководство. — М.: Издательство "Вильямс", 2021.
6. Шульц, Г. Основы Java. Практическое руководство. — М.: Издательство "Наука", 2020.
7. Станкевич, И. О. Java. Лекции и практические задания. — М.: Физматлит, 2021.
8. Юрьев, В. И. Java. Как это работает. — М.: Альпина Пабlishер, 2021.
9. Яковлев, П. Н. Java: от простого к сложному. — М.: Аспект Пресс, 2020.
10. Носов, С. В. Java для начинающих. — М.: Бином. Лаборатория Знаний, 2021.
11. Руководство по Java FX / [Электронный ресурс] // Metanit: [сайт]. Режим доступа: <https://metanit.com/java/javafx/> (дата обращения: 10.08.2026).
12. Java SE / [Электронный ресурс] // Metanit: [сайт]. Режим доступа: <https://metanit.com/java/> (дата обращения: 04.08.2026).

13. Java с нуля / [Электронный ресурс] // Proglang: [сайт]. Режим доступа: <https://proglang.su/java> (дата обращения: 10.08.2026).

Входной тест по теме: «Основы алгоритмики».

1. Алгоритм - это
 - а) правила выполнения определенных действий;
 - б) предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;
 - в) набор команд для компьютера.
2. Какой из документов является алгоритмом?
 - а) Правила техники безопасности.
 - б) Инструкция по получению денег в банкомате.
 - в) Расписание уроков.
3. Какой из объектов может являться исполнителем?
 - а) Луна. б) Карта. в) Принтер. г) Книга
4. Дискретность – свойство алгоритма, означающее...
 - а) однозначность правил выполнения алгоритма
 - б) правильность результатов выполнения алгоритма
 - в) деление алгоритма на отдельные шаги
5. Свойством алгоритма является:
 - а) конечность;
 - б) цикличность;
 - в) возможность изменения последовательности команд;
 - г) возможность выполнения алгоритма в обратном порядке.
6. Алгоритм называется линейным, если:
 - а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
 - б) ход его выполнения зависит от истинности тех или иных условий;
 - в) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий.
7. Алгоритм структуры «ветвление» предусматривает
 - а) выбор условий, б) выбор алгоритмов, в) выбор команд (действий)
8. Алгоритм называется циклическим, если:
 - а) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
 - б) ход его выполнения зависит от истинности тех или иных условий;
 - в) его команды выполняются в порядке их естественного следования

друг за другом независимо от каких-либо условий.

9. Алгоритм называется вспомогательным, если

- а) он предполагает выбор действий
- б) повторяет действия до выполнения какого – либо условия;
- в) решает часть задачи и вызывается из основной программы.

10. Цикл со счётчиком

- а) зависит от некоторого условия; б) зависит от известного числа повторений.

11. Какой тип алгоритмической структуры необходимо применить, если последовательность команд выполняется или не выполняется в зависимости от условия

- а) цикл б) ветвление в) линейный.

12. Ромб — графический объект, используемый в блок-схеме для записи:

- а) ввода, вывода данных; б) вычислительных действий;
- в) конца выполнения задачи; г) условия выполнения действий

13. Когда необходимо составлять блок-схему программы?

- а) До начала составления самой программы
- б) После составления программы
- в) В процессе составления программы

14. Наиболее наглядной формой описания алгоритма является структурно-стилизированный метод

- а) представление алгоритма в виде схемы
- б) язык программирования высокого уровня
- в) словесное описание алгоритма

15. Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа -

- а) ассемблер
- б) паскаль
- в) компилятор

16. Разработка алгоритма решения задачи – это

- а) сведение задачи к математической модели, для которой известен метод решения
- б) точное описание данных, условий задачи и ее целого решения
- в) определение последовательности действий, ведущих к получению результатов

17. Разработке алгоритма предшествует

- а) постановка задачи, разработка математической модели, проектирование программ
- б) постановка задачи, разработка математической модели
- в) постановка задачи, выбор метода решения, проектирование программ

18. Раздел переменных определяется служебным словом:

- а) LABEL
- б) TYPE
- в) VAR

19. Для вычисления квадратного корня применяется функция:

- а) SQR(X)
- б) SQRT(X)
- в) ORD(X)

20. Укажите неправильно записанный оператор:

- а) if a<b then a:=a*a else b:=b*b;
- б) if x and y then s:=s+1; else s:=s-1;
- в) if k<>m then k:=m;

21. В Паскале различие в высоте букв (прописные или строчные)

- а) имеет значение при написании текстовых констант
- б) имеет значение для написания служебных слов
- в) имеет значение для написания имен переменных

22. Какое выражение неверно?

- а) Структура подпрограммы аналогична структуре основной программы
- б) Подпрограмма предназначена для выполнения какой-то последовательности

действий

- в) В Паскале имеется две разновидности подпрограмм – процедуры и функции

23. Каким символом разделяются перечень разнотипных формальных параметров в процедуре?

- а) запятой
- б) точкой с запятой
- в) двоеточием

24. Из предложенных заголовков программы правильным является:

- а) Program сумма;
- б) programm сумма;
- в) programm summa

25. Конечное значение переменной Y в результате выполнения следующих действий: $Y:=5$; $X:=2$; $Y:=(Y-2*X)/2+X$ будет равно:

- а) 0,5;
- б) 2;
- в) 2,5; г) 5.

26. Примером разветвленного алгоритма является:

- а) переход улицы по сигналу светофора
- б) жизнь растения
- в) заваривание чая

27. К свойствам алгоритма относится?

- а) результативность, дискретность, массовость
- б) дискретность, массовость, абсолютность
- в) массовость, дискретность, результативность, детерминированность

28. Что не является способом записи алгоритма?

- а) Словесное описание
- б) Программа
- в) Блок-схема

29. Алгоритмическая структура какого типа записана ниже?

IF Условие THEN

Серия 1

ELSE

Серия 2

END IF

- а) Циклическая
- б) Ветвление
- в) Линейная

30. Вставьте пропущенное слово, выбрав его из списка. Блок-схема - это форма записи алгоритма, при которой для обозначения различных шагов алгоритма используются....

- а) формулы
- б) списки
- в) геометрические фигуры

Критерии оценивания итогового проекта

№ п/п	Критерий	Показатели	Балл/ Макс.балл
1	Работоспособность	Программа компилируется без ошибок и успешно запускается. Основной функционал работает в соответствии с техническим заданием. Программа корректно обрабатывает основные сценарии использования.	___/25
2	Качество кода	Код написан в соответствии со стандартами Java Code Conventions (названия переменных, классов, методов — CamelCase; правильные отступы; наличие комментариев). Отсутствие "магических чисел", дублирования кода, "захардкоженных" значений. Код читаемый и структурированный.	___/20
3	Архитектура и ООП	Использование принципов объектно-ориентированного программирования: инкапсуляция (геттеры/сеттеры, модификаторы доступа), наследование, полиморфизм. Логичное разделение на классы и методы. Отсутствие "божественных" классов.	___/15
4	Сложность и объём проекта	Объём кода (не менее 300–500 строк), наличие нескольких классов/интерфейсов. Использование изученных структур	___/15

		данных (коллекции), обработка исключений, работа с файлами или графическим интерфейсом.	
5	Наличие графического интерфейса или интерактивности	Программа имеет удобный интерфейс (консольный с меню или графический Swing/JavaFX). Обработка пользовательского ввода, понятные подсказки и обратная связь для пользователя.	___/10
6	Оформление проекта	Наличие документации: README-файл с описанием проекта, инструкцией по установке и запуску. Наличие комментариев в коде. Аккуратное оформление репозитория (структура папок, файлы).	___/5
7	Защита проекта	Умение презентовать свою работу: четко объяснить цель, функционал, архитектуру. Способность отвечать на вопросы по коду и принятым решениям. Владение терминологией.	___/10
Итого:	Оценка: _____ Проверил: _____		___/100

Шкала перевода баллов:

Уровень	Баллы	Оценка
Высокий	85–100	«отлично»
Выше среднего	70–84	«хорошо»
Средний	50–69	«удовлетворительно»
Низкий	менее 50	«неудовлетворительно»
Уровень	Баллы	Оценка