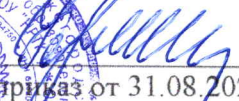


МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ НИЖЕГОРОДСКОЙ ОБЛАСТИ
Государственное бюджетное профессиональное образовательное учреждение
«НИЖЕГОРОДСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

СОГЛАСОВАНО:
Директор МАОУ № 186
«Авторская академическая школа»

М.В.Буров



УТВЕРЖДАЮ:
Директор ГБПОУ «НРТК»

И.А.Кормщицова
приказ от 31.08.2023г. № 96-ИТ/О



СОГЛАСОВАНО:
Педагогический совет
Протокол № 1 от 30.08.2023г.

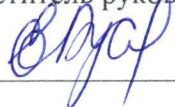
**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ
ПРОГРАММА
«Основы программирования на Java»
(базовый уровень)**

Направленность – техническая
Возраст обучающихся: 12-14 лет
Срок реализации программы: 68 часов

г.Нижний Новгород
2023г.

Внутренняя экспертиза проведена. Программа рекомендована к рассмотрению на педагогическом совете учреждения.

Заместитель руководителя по ИМР

 /Русинова Е.Г.

«31» августа 2023 г.

Дополнительная общеразвивающая программа соответствует действующим федеральным, региональным нормативным документам Российской Федерации и локальным нормативным актам ГБПОУ «НРТК»

Информационная карта

Полное наименование программы	Дополнительная общеразвивающая программа «Основы программирования на Java» (базовый уровень)
Направленность	техническая
Общий объем программы	68 часов
Целевая категория обучающихся	обучающиеся 12 – 14 лет
Аннотация программы	Программа направлена на развитие интереса обучающихся к программированию, овладение первоначальными навыками программирования на языке Java и проектной деятельности в рамках работы над итоговыми проектами. Обучение по программе позволяет получить практические навыки и знания, необходимые для написания правильного и чистого кода.
Планируемые результаты реализации программы	Обучающиеся изучат особенности языка программирования Java, научатся корректно использовать структуру и принципы его программного кода для создания собственных приложений. Также обучающиеся смогут сформировать навыки критического и творческого мышления при решении практических кейсов, а также навыки проектной деятельности в процессе работы над собственным итоговым проектом.
Авторы-составители	– педагог дополнительного образования Григорян Н.М.

1. Комплекс основных характеристик программы

1.1. Пояснительная записка

В современном мире язык программирования Java остается одним из самых популярных языков среди разработчиков благодаря его особенностям. Большинство крупных компаний используют этот язык для создания серверных и пользовательских приложений, в том числе в области мобильной разработки и в работе с большими данными. Java-разработчики остаются востребованными на рынке труда, поэтому подготовка специалистов этого профиля является важной задачей современного образования. Программа «Основы программирования на Java» (базовый уровень) направлена на ознакомление обучающихся с этим языком, получение первичных навыков программирования, что поможет им в будущем профессиональном определении.

Дополнительная общеразвивающая программа «Основы программирования на Java» (базовый уровень) реализуется посредством сетевого взаимодействия между ГБПОУ «Нижегородский радиотехнический колледж» и МАОУ № 186 «Авторская академическая школа».

Актуальность программы обосновывается необходимостью популяризации языка программирования среди обучающихся, развитием мотивации к изучению IT-технологий с целью их будущего профессионального определения в сфере информационных технологий и искусственного интеллекта.

Направленность программы. Программа обладает технической направленностью и предназначена для использования в системе дополнительного образования детей. Обучающиеся в ходе занятий приобщаются к знаниям в области информационных технологий, формируют логическое мышление, формируют навыки программирования на языке Java.

Адресат программы. Дополнительная программа «Основы программирования на Java» (базовый уровень) предназначена для детей в возрасте 12 – 14 лет, без ограничений возможности здоровья. Программа ориентирована на обучающихся, которые имеют базовый уровень подготовки по математике и информатике и проявляют интерес к программированию. Содержание программы составлено с учётом возрастных и психологических особенностей детей данного возраста.

Форма проведения занятий. По форме организации деятельности предполагаются групповые формы занятий, количество обучающихся в группе 7-17 человек.

Место проведения занятий: г. Нижний Новгород, ул. Генкиной, 84.

Режим занятий. Занятия проводятся 2 раза в неделю по 2 академических часа с перерывом на 10 минут для проветривания помещения и отдыха обучающихся. Продолжительность одного занятия – 40 минут.

Условия реализации программы. Зачисление детей производится без предварительного отбора (свободный набор).

Срок освоения общеразвивающей программы определяется содержанием программы и составляет 4 месяца (68 часов).

Формы обучения: очная, с применением дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

Виды занятий: беседы, мастер-классы, кейсы, деловые игры, практикумы, групповые занятия, самостоятельная работа, проект, консультация.

Уровень освоения программы. По уровню освоения программа общеразвивающая, *базовая*. Это обеспечивает возможность обучения с любым уровнем подготовки.

1.2. Цели и задачи программы

Цель программы – формирование базовых знаний и практических навыков программирования на языке Java посредством создания реально действующих кроссплатформенных программ на основе практических кейсов, а также развитие мотивации к дальнейшему изучению языка программирования Java.

Для успешной реализации поставленной цели необходимо решить ряд поставленных задач:

Обучающие (предметные):

- ознакомить обучающихся с особенностями языка программирования Java, принципами написания и структурой программного кода;
- научить задавать переменные, присваивать им значения, а также использовать условные операторы if, else, switch – case, тернарный оператор и булевы операторы;

- научить применять бесконечные циклы и циклы с заданными параметрами, изучить операторы break и continue;
- научить создавать и использовать классы, методы в Java;
- познакомить с принципом инкапсуляции, понятием имутабельности и копировании объектов;
- научить создавать константы и целесообразно применять их в написании программного кода;
- научить отличать примитивы от объектов;
- научить работать с различными операциями над числами: арифметическими, с присвоением, инкрементом, декрементом, сравнением;
- научить работать со строками: задавать строки, сравнивать их, объединять.

Развивающие (метапредметные):

- сформировать навыки критического и творческого мышления при решении задач, практических кейсов, выполнении индивидуальных и/или групповых заданий;
- развить познавательную активность, мотивацию к самостоятельному изучению дополнительных материалов по темам модулей;
- развить внимание, память, логику и концентрацию при работе над проектами;
- сформировать навыки проектной деятельности, в том числе коммуникативные навыки в процессе работы в команде;
- сформировать навык анализа промежуточных результатов деятельности, работы над ошибками, подбора разных способов улучшения эффективности своей деятельности.

Воспитательные (личностные):

- сформировать дисциплину и ответственность за свои действия и результаты деятельности;
- развить трудолюбие и уважение к результатам своего труда и труда окружающих;
- сформировать навык самоорганизации, самоконтроля и рефлексии;
- воспитать мотивацию к творчеству, созданию продуктов интеллектуальной собственности, инновационных разработок;
- воспитать чувство коллективизма, взаимопомощи и толерантности.

2. Содержание дополнительной общеразвивающей программы

2.1. Учебный план

№ п/п	Наименование раздела/модуля	Количество часов			Формы промежуточного контроля
		Всего	Теория	Практика	
1	Введение в программу	2	1	1	-
2	Модуль 1. Синтаксис языка.	16	4	12	Тестирование
3	Модуль 2. Введение в объектно-ориентированное программирование на Java.	30	11	19	Выполнение практической работы
4	Модуль 3. Строки, числа и даты	16	6	10	Тестирование
5	Модуль 4. Проектная деятельность.	10	0	10	-
6	Итоговая аттестация	2	0	2	Защита проектов
Итого:		76	22	54	

2.2. Содержание учебного плана

№ п/п	Наименование разделов и тем	Содержание учебного материала, лабораторные и практические работы, самостоятельная работа обучающихся, проект	Количество часов			Основные виды деятельности обучающихся
			Всего	Теория	Практика	
1.	Вводное занятие. Ознакомление с программой.	<p>Теория</p> <p>Инструктаж по технике безопасности. Цели и области применения Java. Основные термины. Как выглядит программный код на языке Java. Какие программы можно разработать на языке Java. Frontend и backend. Особенности языка Java.</p>	1	1	0	Беседа, ответы на контрольные вопросы.
2.	Среда разработки IntelliJ IDEA Community Edition.	<p>Практика</p> <p>Установка среды разработки IntelliJ IDEA Community Edition, JDK. Изучение основных инструментов среды разработки. Этапы создания и запуска приложения. Структура программы на языке Java. Создание нового проекта в среде разработки. Написание простой программы сообщаемой дату и время. Запуск кода.</p>	1	0	1	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
Модуль 1. Синтаксис языка						
3.	Переменные. Условные операторы.	<p>Теория</p> <p>Переменные int, double, boolean, String, var. Статическая типизация и явная типизация. Условный оператор if - else. Булевы операции. Операторы &&, , !. Приоритеты и скобки в условиях. Вложенные условия. Тернарный оператор. Комментарии в коде.</p> <p>Практика</p> <p>Написание кода с использованием условных операторов if и else. Написание кода с использованием булевых операторов. Упрощение кода с помощью тернарного оператора и объединения условий.</p>	6	2	4	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.

4.	Циклы.	Теория	8	2	6	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Циклы и их параметры. Понятие итерации. Циклы for, while, do while. Основные методы ввода данных с компьютера. Бесконечные циклы. Операторы break и continue. Условный оператор switch - case.				
		Практика				
		Создание циклов с заданными параметрами, создание бесконечных циклов. Использование операторов break, continue, условного методов ввода с клавиатуры в создании цикла. Упрощение кода с помощью условного оператора switch - case.				
Модуль 2. Введение в объектно-ориентированное программирование на Java						
5.	Методы и классы.	Теория	8	3	5	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Метод: модификаторы доступа, параметры метода, тело метода. Возвращаемый тип метода, ключевое слово return, ключевое слово void, стек вызовов. Класс, ключевое слово new, ключевое слово static, объект. Ключевое слово this, конструктор, конструктор без параметров, конструктор с параметрами. Перегрузка метода, сигнатура метода. Области видимости.				
		Практика				
		Создание метода, который будет выводить в консоль текущую дату и текущее время. Создание метода, вызывающего самого себя. Написание методов с одним и несколькими параметрами. Создание и вызывание методов, возвращающих значение. Создание классов с методами и переменными и объектами на их основе. Доработка существующих классов. Создание и использование классов с конструкторами. Создание классов с переопределенными методами.				
6.	Инкапсуляция.	Теория	6	3	3	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Определение инкапсуляции. POJO-классы, геттеры и сеттеры. Иммутабельность. Управление изменяемостью полей классов. Передача по ссылке или по значению. Garbage collector. Копирование объектов.				
		Практика				
		Написание кода класса с применением принципа инкапсуляции. Создание POJO-классов, геттеров и сеттеров. Создание иммутабельных классов.				

7.	Статические переменные и методы.	Теория	6	2	4	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Статическая переменная. Статические и нестатические методы. Константы. Перечисляемые типы - enum. Блок статической инициализации.				
		Практика				
		Создание класса с двумя статическими переменными и методами. Создание класса, использующего enum-переменную. Создание нескольких связанных классов и enum.				
8.	Примитивы.	Теория	4	2	2	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Что такое примитивы. Сравнение примитивов и объектов. Виды примитивов. Диапазоны значений и значения по умолчанию. Системы счисления. Единицы измерения объема информации. Биты и байты. Числа, способы их записи. Примитив char. Специальные символы. Преобразование символов в код. Классы-обёртки, их особенности. Boxing, unboxing.				
		Практика				
		Поиски и исправление ошибок в коде. Написание кода с использованием преобразования символов в код.				
Модуль 3. Строки, числа и даты						
9.	Операции с числами.	Теория	8	3	5	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Арифметические операции. Операции с присвоением. Операции сравнения. Класс Math и его методы для выполнения сложных математических операций. Инкремент и декремент. Пост- и преинкремент. Пост- и предекремент. Преобразование чисел разных типов. Явное преобразование. Неявное преобразование. Точность вычислений. Работа с большими числами и точными числами. Классы BigInteger и BigDecimal. Дата и время: их сравнение, вычисление разницы. Парсинг. Метка времени: timestamp в Java.				
		Практика				
		Работа с различными операциями, которые можно производить над числами: арифметическими операциями, операциями с присвоением, инкрементом и декрементом. Работа с большими числами. Преобразование метки времени в дату. Получение метки времени из даты.				

10.	Работа со строками.	Теория	8	3	5	Наблюдение за работой педагога, самостоятельная работа с инструментами среды.
		Строки, конкатенация и сравнение. Преобразование чисел в строки и обратно. Символы и кодировки. Методы работы с подстроками. String Pool. Регулярные выражения.				
		Практика				
		Создание консольного приложения, которое будет сравнивать строки. Создание консольного приложения, которое будет вычислять значения простых арифметических выражений. Написание консольного приложения, разделяющего строку с фамилией, именем и отчеством на компоненты. Написание приложения, извлекающего из строки заработные платы людей и суммирующее их. Написание консольного приложения с использованием регулярного выражения.				
Модуль 4. Проектная деятельность						
11.	Создание собственного проекта.	Практика	10	0	10	Самостоятельная работа с инструментами среды по созданию собственного приложения, консультирование обучающихся.
		Создание собственного консольного приложения с использованием полученных знаний и выработанных навыков программирования на языке Java.				
12.	Итоговая аттестация.	Демонстрация работы созданного приложения.	2	0	2	Выступление, рефлексия.
Итого			68	21	47	

2.3. Календарный учебный график

Номер учебной недели/ группа	сентябрь				октябрь				ноябрь				декабрь				
	04.09.2023-10.09.2023	11.09.2023-17.09.2023	18.09.2023-24.09.2023	25.09.2023-01.10.2023	02.10.2023-08.10.2023	09.10.2023-15.10.2023	16.10.2023-22.10.2023	23.10.2023-29.10.2023	30.10.2023-05.11.2023	06.11.2023-12.11.2023	13.11.2023-19.11.2023	20.11.2023-26.11.2023	27.11.2023-03.12.2023	04.12.2023-10.12.2023	11.12.2023-17.12.2023	18.12.2023-24.12.2023	25.12.2023-31.12.2023
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ПД-3	У	У	У	У	У	У	У	У	У	У	У	У	У	У	У	У	И

Считать нерабочими праздничными днями: 4 ноября.

Условные обозначения:

- У – учебные занятия,
- П – промежуточный контроль,
- И – итоговая аттестация.

2.4. Планируемые результаты обучения

Обучающие (предметные):

- обучающиеся понимают особенности языка программирования Java, корректно используют структуру и принципы написания программного кода;
- обучающиеся умеют задавать переменные, присваивать им значения, а также использовать условные операторы if, else, switch – case, тернарный оператор и булевы операторы;
- обучающиеся научились создавать бесконечные циклы и циклы с заданными параметрами, корректно используют операторы break и continue;
- обучающиеся умеют создавать и использовать классы, методы в Java;
- обучающиеся понимают принципы инкапсуляции, понятия иммутабельности и копирования объектов;
- обучающиеся умеют создавать константы и целесообразно применять их в написании программного кода;
- обучающиеся понимают сходства и отличия примитивов от объектов;
- обучающиеся научились работать с различными операциями над числами: арифметическими, с присвоением, инкрементом, декрементом, сравнением;
- обучающиеся умеют работать со строками: задавать строки, сравнивать их, объединять.

Развивающие (метапредметные):

- сформированы навыки критического и творческого мышления при решении задач, практических кейсов, выполнении индивидуальных и/или групповых заданий;
- развита познавательная активность, обучающиеся самостоятельно изучают дополнительные материалы по темам модулей;
- развиты внимание, память, логика и концентрация при работе над проектами;
- сформированы навыки проектной деятельности, в том числе коммуникативные навыки в процессе работы в команде;
- сформирован навык анализа промежуточных результатов деятельности, работы над ошибками, подбора разных способов улучшения эффективности своей деятельности.

Воспитательные (личностные):

- сформирована дисциплина и ответственность за свои действия и результаты деятельности;
- развиты трудолюбие и уважение к результатам своего труда и труда окружающих;
- сформирован навык самоорганизации, самоконтроля и рефлексии;
- сформирован интерес к творчеству, созданию продуктов интеллектуальной собственности, инновационных разработок;
- сформированы чувство коллективизма, взаимопомощи и толерантности.

2.5. Определение результативности реализации программы Формы аттестации

Система отслеживания результатов обучающихся выстроена следующим образом:

- текущий контроль;
- итоговый контроль.

Текущий контроль осуществляется путём наблюдения, определения качества выполнения практических работ, отслеживания динамики развития обучающегося в рамках учебных занятий. Способы проверки уровня освоения тем: опрос, выполнение упражнений, наблюдение, оценка выполненных самостоятельных работ.

Итоговый контроль проводится в виде общей рефлексии, анализа результатов прохождения обучающимися программы и итоговой аттестации. Итоговая аттестация осуществляется в форме защиты проекта, демонстрации работы созданного приложения и оценивается по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы (Таблица 1).

Таблица 1

Критерии оценивания уровня усвоения программы

Баллы, набранные обучающимся	Уровень освоения	Результат
0–30 баллов	низкий	Обучающиеся демонстрируют низкий уровень заинтересованности в учебной, проектной деятельности, составляющих содержание программы. Итоговый проект не соответствует требованиям, содержит ошибки в коде, что делает код написанного приложения нефункциональным.
31–70 баллов	средний	Обучающиеся демонстрируют достаточную заинтересованность в учебной, проектной деятельности, составляющих содержание программы. Итоговый проект соответствует требованиям, код написанного приложения работает, но требует незначительной доработки.
71–100 баллов	высокий	Обучающиеся демонстрируют высокую заинтересованность в учебной, проектной деятельности, составляющих содержание программы. Итоговый проект соответствует требованиям, код написанного приложения работает и не требует доработки. Итоговый проект содержит реализацию собственной разработки, творческой идеи.

Итоговый проект оценивается специально формируемой комиссией. Состав комиссии (не менее 3-х человек): педагог (в обязательном порядке), представитель администрации учебной организации, приветствуется привлечение IT-профессионалов, представителей высших и других учебных заведений.

Критерии оценивания итоговых работ разрабатываются педагогом в зависимости от содержания программы, ее целей и задач. Критерии оценивания итоговых работ обучающихся по программе «Основы программирования на Java» (базовый уровень) представлены в таблице 2:

Таблица 2

№ п/п	Критерий	Баллы
Оценка итогового проекта		
1.	Функциональность приложения	0 – приложение не работает, или работает некорректно; 5 – приложение работает, но требуются значительные доработки; 10 – приложение работает, требуются незначительные доработки, не влияющие на функциональность кода; 15 – приложение работает, написан функциональный код.

2.	Аккуратность и читабельность программного кода (Приложение 1)	0 – программный код нечитабельный, присутствует более 7 нарушений правил написания; 5 – программный код неаккуратный, присутствует от 4 до 6 нарушений правил написания; 10 – программный код читабельный, но присутствуют незначительные нарушения правил написания (1-3 нарушения); 15 – программный код написан аккуратно, правила написания читабельного кода не нарушены.
3.	Именованые проектов, классов, методов и переменных (Приложение 2)	0 – присутствует более 4 различных нарушений правил именования проектов, классов, методов и переменных; 5 – присутствует 2-3 различных нарушения правил именования проектов, классов, методов и переменных; 10 – присутствуют один тип нарушения правил именования проектов, классов, методов и переменных; 15 – полностью соблюдены правила именования проектов, классов, методов и переменных.
4.	Комментарии в программном коде	0 – большое количество излишних комментариев, которые мешают чтению кода; в программе есть закоментированные фрагменты кода, которые не используются; комментарии дублируют содержание кода; 5 – комментарии написаны четко и кратко; комментарии содержат важную для понимания кода дополнительную информацию.
Оценка демонстрации итогового проекта		
1.	Целеполагание, планирование проектной деятельности	0 – цель не сформулирована; 3 – цель сформулирована нечетко, отсутствует план ее достижения; 7 – цель сформулирована ясно, отсутствует план ее достижения; 10 – цель сформулирована ясно, присутствует подробный план ее достижения.
2.	Глубина раскрытия содержания темы проекта	0 – тема проекта не раскрыта 3 – тема проекта раскрыта фрагментарно, присутствует отклонение от выбранной темы; 7 – тема проекта раскрыта, автор показал достойный уровень знаний по выбранной теме; 10 – тема проекта раскрыта, автор продемонстрировал глубокие знания по выбранной теме.
3.	Презентация итогового проекта	0 – отсутствует презентация; 3 – презентация есть, но отсутствует четкая структура, допущены ошибки в оформлении; 7 – презентация обладает четкой структурой, присутствуют незначительные ошибки в оформлении; 10 – презентация обладает четкой структурой, отличается грамотным оформлением.
4.	Оценка выступления	0 – автор читает текст, слабо владеет материалом, не умеет вести дискуссию и не может ответить на вопросы членов комиссии; 3 – автор самостоятельно представляет свою итоговую работу, но присутствуют речевые ошибки, слабо владеет материалом, испытывает затруднения в умении вести дискуссию и отвечать на вопросы членов комиссии; 7 – автор самостоятельно представляет свою итоговую работу, но присутствуют незначительные речевые ошибки, владеет материалом на достойном уровне, умеет

		вести дискуссию и отвечать на вопросы членов комиссии; 10 - автор самостоятельно представляет свою итоговую работу, отсутствуют речевые ошибки, владеет материалом на достойном уровне, умеет вести дискуссию, доказательно и корректно защищать свою идею и отвечать на вопросы членов комиссии;
5.	Рефлексия	0 – отсутствуют выводы по результатам проделанной работы; 3 – выводы по работе представлены неполно; 7 – выводы по работе представлены полно, присутствует анализ сложностей в ходе работы над проектом; 10 – выводы по работе представлены полно, присутствует анализ сложностей в ходе работы над проектом и предложен хотя бы один способ их преодоления в рамках работы над будущими проектами.

3. Комплекс организационно-педагогических условий реализации дополнительной общеразвивающей программы

3.1. Материально-техническое обеспечение

Требования к помещению:

- помещение для занятий, отвечающее требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочее место для педагога.

Общее оборудование:

- ноутбук тип 1 – 1 шт.;
- ноутбук тип 2 – 14 шт.

Презентационное оборудование:

- моноблочное интерактивное устройство SMART MX265-V2 – 1 шт.;
- доска магнитно-маркерная настенная с полимерным покрытием (ДО-17б) – 1 шт.;
- доска-флипчарт магнитно-маркерная 70 x 100 см BRAUBERG Стандарт – 1 шт.;
- напольная мобильная стойка для интерактивных досок DIGIS DSM-P1060CL – 1 шт.

Дополнительное оборудование:

- роутер TP-Link Archer C6U – 1 шт.;
- web-камера A4Tech PK-940HA – 1 шт.;
- ультрафиолетовый рециркулятор Milerd DZR-1 – 1 шт.;
- тележка для хранения и зарядки 20 ноутбуков – 1 шт.;
- лазерный цветной принтер HP Color LaserJet Pro M255dw (7KW64A) – 1 шт.;
- огнетушитель ОУ-3 (ВСЕ)

Материалы:

- подставка под огнетушитель П-15 – 1 шт.;
- комплект переходников (Набор Rexant 18-1203 USB 6 переходников+удлинитель) – 1 шт.;
- компьютерная мышь Logitech Mouse M100USB, проводная, черная – 15 шт.;
- наушники SVEN AP-860V – 14 шт.;
- кресло для ученика Нота (экокожа, цвет бежевый, крестовина «Паук», глайдеры) – 14 шт.

Мебель:

- кресло для учителя СН-883 (экокожа, цвет Слоновая кость, крестовина хром) – 1 шт.;
- шкаф (Размер(Ш*Г*В) 500мм*415мм*2000мм.Корпус и полки ЛДСП-16мм цвет: серый шиншилла – 1 шт.;
- стол преподавателя с тумбой, левый – 1 шт.;
- стол ученический тип 1 – 14 шт.

Программное обеспечение:

- среда разработки IntelliJ IDEA.

3.2. Методические материалы

Основной тип занятий — **комбинированный**, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств. Единицей учебного процесса является раздел, каждый из них охватывает отдельную информационную технологию или её часть. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами. Каждая тема курса начинается с описание конкретных задач, определяющих содержание материал, которое предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения. Основные задания являются обязательными для выполнения всеми обучающимися в группе. Задания выполняются на компьютере с использованием интегрированной среды разработки IntelliJ IDEA Community Edition. При этом обучающиеся не только формируют теоретические практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на **индивидуальный** подход. Для того чтобы каждый обучающийся получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы. Такая форма организации обучения стимулирует интерес обучающегося к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

В рамках учебных занятий больше часов отводится на самостоятельную работу обучающихся, реализуемую посредством выполнения практических заданий, решения кейсов. Это позволяет укрепить знания по пройденной теме и развить навык программирования и работы в среде разработки. Более того, это способствует реализации индивидуального подхода к обучению, поскольку каждый обучающийся решает задачу или кейс в своем темпе. Для обучающихся, которые быстро справляются с решением практической работы, у педагога всегда есть дополнительный материал повышенного уровня сложности для дополнительной практики.

В ходе обучения проводится проверочные задания, которые позволяют оценить уровень остаточных знаний обучающихся, а также способствует активизации учебно-познавательной деятельности, что служит индикатором успешности образовательного процесса.

Реализация программы предполагает использование **здоровьесберегающих** технологий. Специальных медицинских противопоказаний к занятиям не существует, но при выборе данного объединения родителям и педагогу необходимо с особым вниманием отнестись к обучающимся, относящимся к группе риска по зрению, так как на занятиях значительное время они пользуются компьютером. Педагогами проводится предварительная беседа с родителями, в которой акцентируется внимание на обозначенную проблему.

Здоровьесберегающая деятельность реализуется:

- созданием безопасных материально-технических условий;
- включением в занятие динамических пауз, периодической смены деятельности обучающихся;
- выполнение комплекса упражнений для релаксации глаз, а также увеличения остроты зрения (Приложение 3);
- контролем педагога за соблюдением обучающимися правил работы за персональным компьютером;
- созданием благоприятного психологического климата в учебной группе в целом.

Список использованной литературы и интернет-ресурсы

Основная литература

1. Блинов И.Н., Романчик В.С.. Java 2 Практическое руководство. // Мн.: Универсал Пресс, 2005 – 400 с.
2. Блох Д. Java. Эффективное программирование. // М.: Лори, 2002 – 224 с.
3. Макконнелл С. Совершенный код.// СПб: Питер, 2005 – 868 с.

4. Седжвик Роберт, Уэйн Кевин. Алгоритмы на Java., пер. с англ. – 4-е изд. - М.: Вильямс, 2013 – 848 с.
5. Скиена С.С., Ревилла М.А. Олимпиадные задачи по программированию. Руководство по подготовке к соревнованиям. Пер. с англ. – М.: КУДИЦ-ОБРАЗ, 2005 – 416 с.
6. Сухов С. А. Учебное пособие Основы программирования на Java. Ульяновск: Ул ГТУ, 2006 – 88 с.
7. Сьерра К., Бейтс Б. Изучаем Java. изд. Эксмо, 2020 – 720 с.
8. Харди Б., Филлипс Б. Android. Программирование для профессионалов. 2-е изд. – СПб.: Питер, 2016 – 640 с.
9. Шилдт Герберт. Полный справочник по Java. пер. с англ. – 7-е изд. – М.: Вильямс, 2007 – 1040 с.
10. Эккель Б. Философия Java. 4-е изд. СПб.: Питер, 2009 – 637с.

Интернет-источники

1. Интегрированная среда разработки с использованием Java «Online Java Compiler»: <https://www.jdoodle.com/online-java-compiler/>
2. Интерактивная среда разработки на языке Java «Browxy»: <https://www.browxy.com/>

Литература, рекомендованная для учащихся и родителей

1. Арсак Ж. Программирование игр и головоломок. — М.: Наука, 1990 —224 с.
2. Бентли Д. Жемчужины творчества программистов: пер. с англ. — М.: Радио и связь, 1990 — 224 с.
3. Окулов С.М., Пестов А.А. 100 задач по информатике. Киров: Изд-во ВГПУ, 2000 — 272с.
4. Уэзерелл Ч. Этюды для программистов. — М.: Мир, 1982 — 288 с.

Правила написания аккуратного и читабельного программного кода:

1. Определение переменных нужно располагать в начале блока, а не «ждать» первого использования переменной. Инициализация должна производиться, по возможности, сразу.
2. Между именем метода и скобками для списка параметров нет пробела.
3. Параметры разделяются пробелом.
4. Пробелы окружают любой оператор.
5. Ключевое слово и следующая за ним скобка (должны разделяться пробелом.
6. Открывающаяся фигурная скобка располагается на той же строке, что и сигнатура метода/заголовок if, while-блока и т.д.
7. Закрывающаяся фигурная скобка выровнена по строке начала данного блока.
8. Методы разделяются пустой строкой, объявления свойств класса располагаются по одному на строку.
9. На строке располагается только один оператор.

Правила именования проектов, классов, методов и переменных:

1. Имена проектам, классам, методам и переменным даются на английском языке.
2. Имена классов должны быть существительными, первые буквы всех слов — заглавные.
3. Названия методов должны быть глаголами, отражать свойства метода, первая буква должна быть строчной, первые буквы внутренних слов — заглавные.
4. Имена переменных должны начинаться со строчной буквы, внутренние слова — с заглавной.
5. Имена проектов, классов, методов и переменных должны отражать их содержание (их названия должны быть понятны для любого разработчика, который будет читать код).

