

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Казанский (Приволжский) федеральный университет»
Высшая школа педагогического мастерства

УТВЕРЖДАЮ

Проректор



_____ И.А. Хайруллин
(подпись) (инициалы, фамилия)

« _____ » _____ 20__ г.

**Дополнительная профессиональная программа повышения квалификации
«Программирование на Python: от основ до подготовки к ЕГЭ»**

Руководитель
образовательной программы

Л.Ф. Осипова

Казань, 2026

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации программы

Целью реализации программы является совершенствование профессиональных компетенций учителей информатики в области преподавания программирования и получение новой компетенции, связанной с использованием языка Python, современных педагогических методик, подготовки обучающихся к выполнению заданий ЕГЭ по информатике, а также применения технологий искусственного интеллекта в образовательном процессе.

Программа ориентирована на педагогов, не имеющих предварительной подготовки в программировании, и предполагает обучение «с нуля», в доступной и пошаговой форме.

1.2. Планируемые результаты обучения

1.2.1. Перечень профессиональных компетенций:

В результате освоения программы у слушателей формируются и совершенствуются следующие профессиональные компетенции:

1. Способность преподавать основы программирования на языке Python в основной и средней школе.
2. Способность разрабатывать и реализовывать учебные занятия по программированию с учетом возрастных и когнитивных особенностей обучающихся.
3. Способность подготавливать обучающихся к выполнению заданий ЕГЭ по информатике (№ 17, 24, 25, 26, 27).
4. Способность использовать цифровые и ИИ-инструменты (включая чат-ботов) в образовательной деятельности.
5. Способность применять методику обучения программированию «Юный кодер».

1.2.2. Планируемые знания и умения, обеспечивающие формирование и совершенствование компетенций:

Слушатель, успешно освоивший программу, должен знать:

- базовые понятия алгоритмизации и программирования;
- синтаксис и основные конструкции языка Python;
- принципы работы со списками, словарями, функциями;
- требования и типовые модели заданий ЕГЭ по информатике;
- методические подходы к обучению программированию с нуля;
- основы Vibe-кодинга и применения ИИ в обучении.

Слушатель, успешно освоивший программу, должен уметь:

- устанавливать и настраивать Python и среду разработки VS Code;
- писать простые программы на Python;
- объяснять программирование «с нуля», без перегрузки терминологией;
- разбирать и объяснять задания ЕГЭ по программированию;
- использовать программу «Юный кодер» в учебном процессе;
- создавать простых ботов с использованием ИИ-платформ.

1.3. Требования к уровню подготовки поступающего на обучение

К освоению дополнительных профессиональных программ допускаются лица, имеющие среднее профессиональное и (или) высшее педагогическое образование. Специальных знаний в области программирования не требуется.

1.4. Программа разработана на основе:

Программа разработана в соответствии с приказом Министерства образования и науки Российской Федерации от 24.03.2025 года № 266 «Об утверждении Порядка организации осуществления образовательной деятельности по дополнительным профессиональным программам» и на основе профессионального стандарта «Педагог», утвержденного приказом Министерства труда и социальной защита Российской Федерации от 18 октября 2013 года № 544н.

1.5. Форма обучения – заочная

1.6. Формат обучения – онлайн-асинхронный

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный план

Наименование раздела	Трудоемкость, час	Дистанционные занятия			Промежуточная аттестация	
		Всего, час.	в том числе		Зачет	Экзамен
			лекции	прак. занятия, семинары		
1	2	3	4	5	7	8
Модуль 1. Основы программирования на Python и методика преподавания	24	24	12	12	+	
Введение в программирование	2	2	1	1		
Установка и настройка среды разработки	4	4	2	2		
Переменные и типы данных	4	4	2	2		
Линейные алгоритмы. Условные операторы	4	4	2	2		
Циклы	2	2	1	1		
Списки, Словари	4	4	2	2		
Функции	2	2	1	1		
Методика преподавания Python	2	2	1	1		
Модуль 2. Методика обучения программированию «Юный кодер»	24	24	12	12	+	
Введение в программу «Юный кодер»	3	3	2	1		

Наименование раздела	Трудоемкость, час	Дистанционные занятия			Промежуточная аттестация	
		Всего, час.	в том числе		Зачет	Экзамен
			лекции	прак. занятия, семинары		
1	2	3	4	5	7	8
Методические основы обучения детей	2	2	1	1		
Структура курса «Юный кодер»	4	4	2	2		
Игровые методы обучения	4	4	2	2		
Практика преподавания	4	4	1	3		
Контроль и оценка результатов	4	4	2	2		
Интеграция в школьный курс	3	3	2	1		
Модуль 3. Подготовка к ЕГЭ по информатике. ИИ и Vibe-кодинг	22	22	11	11	+	
Структура ЕГЭ по информатике	2	2	1	1		
Задание №24	3	3	2	1		
Задание №26	3	3	2	1		
Задание №27	4	4	2	2		
Vibe-кодинг	3	3	1	2		
Искусственный интеллект в образовании	4	4	2	2		
Создание бота с ИИ	3	3	1	2		
Всего	70	70	35	35		
Итоговая аттестация	2					2
Итого	72	70	35	35		2

2.2. Календарный учебный график

Период обучения (дни, недели) ¹⁾	Наименование раздела
1 неделя	Модуль 1. Основы программирования на Python и методика преподавания
2 неделя	Модуль 2. Методика обучения программированию «Юный кодер»
3 неделя	Модуль 3. Подготовка к ЕГЭ по информатике. ИИ и Vibe-кодинг

¹⁾Даты обучения будут определены в расписании занятий при наборе группы на обучение

2.3. Рабочие программы разделов

№, наименование темы	Содержание лекций (количество часов)	Наименование практических занятий или семинаров (количество часов)
1	2	3
МОДУЛЬ 1. Основы программирования на языке Python и методика преподавания (24 часа)		
Введение в программирование	Понятие алгоритма. Исполнитель. Программа и язык программирования. Роль программирования в школьном курсе информатики. Психологические трудности начинающих (1 ч)	Обсуждение типичных ошибок начинающих. Анализ школьных заданий без программирования (1 ч)
Установка и настройка среды разработки	Установка Python. Назначение среды VS Code. Структура проекта. Запуск первой программы (1 ч)	Установка Python и VS Code. Создание и запуск первой программы print() (2 ч)
Переменные и типы данных	Понятие переменной. Типы данных: int, float, str. Операции присваивания. Ввод и вывод данных (2 ч)	Решение простых задач на ввод, вывод и вычисления (2 ч)
Линейные алгоритмы. Условные операторы	Понятие линейного алгоритма. Арифметические операции. Порядок выполнения команд. Логические выражения. Оператор if. Ветвление алгоритмов (2 ч)	Решение задач на линейные алгоритмы в формате школьных заданий. Решение задач с условиями. Разбор типовых ошибок учащихся (2 ч)
Циклы	Циклы for и while. Назначение циклов. Типичные ошибки при работе с циклами (1 ч)	Решение задач на циклы. (1 ч)
Списки, Словари	Понятие списка. Индексация. Перебор элементов. Основные методы. Понятие словаря. Ключи и значения. Применение словарей в задачах (2 ч)	Практика работы со списками. Задачи базового уровня. Решение практических задач со словарями. (2 ч.)
Функции	Понятие функции. Аргументы и возвращаемые значения. Назначение функций в учебных программах (1 ч)	Написание простых функций. Разбор типовых заданий (1 ч)
Методика преподавания Python	Принципы объяснения программирования «с нуля». Методика поэтапного обучения. Типичные ошибки учеников (1 ч)	Семинар: моделирование урока по Python (1 ч)
МОДУЛЬ 2. Методика обучения программированию «Юный кодер» (24 часа)		

Введение в программу «Юный кодер»	Цели и задачи программы. Целевая аудитория. Место в школьном курсе информатики (2 ч)	Обсуждение опыта преподавания в начальной и основной школе (1 ч)
Методические основы обучения детей	Возрастные особенности. Формирование алгоритмического мышления (1 ч)	Анализ типичных ошибок обучающихся (1 ч)
Структура курса «Юный кодер»	Логика построения тем. Переход от игры к коду (2 ч)	Разбор учебных материалов программы (2 ч)
Игровые методы обучения	Геймификация. Мотивация обучающихся (1 ч)	Проектирование игровых заданий (2 ч)
Практика преподавания	Организация урока. Работа с разным уровнем учеников (1 ч)	Моделирование фрагментов уроков (3 ч)
Контроль и оценка результатов	Формы контроля знаний. Обратная связь (2 ч)	Разработка проверочных заданий (2 ч)
Интеграция в школьный курс	Связь с основной программой информатики (2 ч)	Внедрение программы в собственный учебный план (1 ч)
МОДУЛЬ 3. Подготовка к ЕГЭ. Vibe-кодинг и ИИ (24 часа)		
Структура ЕГЭ по информатике	Анализ заданий с программированием. Типичные сложности учащихся (1 ч)	Разбор реальных вариантов ЕГЭ (1 ч)
Задание №24	Алгоритмы обработки строк и последовательностей (2 ч)	Решение заданий №24 (1 ч)
Задание №26	Работа с сортировками (2 ч)	Решение заданий №26 (1 ч)
Задание №27	Оптимизационные задачи. Особенности объяснения (2 ч)	Пошаговый разбор задания №27 (2 ч)
Vibe-кодинг	Понятие Vibe-кодинга. Современные подходы к обучению программированию (1 ч)	Применение Vibe-кодинга на уроках и в жизни (2 ч)
Искусственный интеллект в образовании	Возможности ИИ. Ограничения и риски (2 ч)	Использование ИИ для подготовки заданий (1 ч)
Создание бота с ИИ	Принципы работы чат-ботов. Применение в обучении (1 ч)	Практика: создание простого бота с ИИ (2 ч)

2.4. Оценка качества освоения программы (формы аттестации, оценочные и методические материалы)

Оценка качества освоения программы включает промежуточную и итоговую аттестацию.

2.4.1. Форма(ы) промежуточной и итоговой аттестации

Промежуточная аттестация проводится в форме автоматизированного тестирования на платформе электронного обучения КФУ <https://edu.kpfu.ru/course/view.php?id=7496>

Итоговая аттестация проводится в виде экзамена, в форме автоматизированного тестирования на платформе электронного обучения КФУ <https://edu.kpfu.ru/course/view.php?id=7496>

2.4.2. Оценочные материалы

Промежуточная аттестация проводится по итогам освоения каждого модуля образовательной программы. Аттестационное испытание включает 10 тестовых заданий, к каждому из которых предлагается несколько вариантов ответов. Слушатель должен выбрать один правильный ответ.

Примеры вопросов для промежуточной аттестации

Модуль 1

Вопрос	Ответ 1	Ответ 2	Ответ 3	Ответ 4
1. Что такое алгоритм?	Программа на Python	Последовательность команд для решения задачи	Графический рисунок с командами	Текстовое описание инструментов работы
2. Какая программа необходима для написания и запуска кода на Python?	Microsoft Word	Графический редактор Paint	Интерпретатор Python	Процессор электронных таблиц Excel
3. Какая команда используется для вывода текста на экран в Python?	input()	write()	echo()	print()
4. Какой тип данных используется для хранения целых чисел?	float	str	int	bool
5. Что произойдет при выполнении кода: <code>x = 5; print(x)</code> ?	Будет ошибка	Ничего не произойдет	Выведется слово x	Выведется число 5
6. Какой оператор используется для проверки условия в Python?	for	while	if	def
7. Какой цикл используется, когда известно количество повторений?	while	if	for	def
8. Как обратиться к первому элементу списка <code>a = [10, 20, 30]</code> ?	<code>a[1]</code>	<code>a(0)</code>	<code>a[0]</code>	<code>a{0}</code>
9. Для чего используются функции в Python?	Для создания переменных	Для хранения списков	Для запуска программы	Для выделения повторяющихся действий в отдельный блок кода
10. Какой из принципов важен при	Использование сложной терминологии	Максимум теории без практики	Быстрый переход к олимпиадным задачам	Пошаговое объяснение с опорой на практику

объяснении программирования «с нуля»?				
---------------------------------------	--	--	--	--

Модуль 2

Вопрос	Ответ 1	Ответ 2	Ответ 3	Ответ 4
1. Какова основная цель программы «Юный кодер»?	Подготовка к олимпиадам по математике	Изучение офисных программ	Обучение школьников основам программирования на Python	Создание сайтов
2. Как организован экран в программе «Юный кодер» для ученика?	Игровое поле слева и поле кода справа	Только игровое поле	Только поле ввода текста	Таблица и график
3. Какая кнопка позволяет выполнить код построчно?	Запуск кода	Сброс	Подсказки	Пошаговый запуск
4. Что делает кнопка «Сброс робота»?	Удаляет код	Возвращает робота в исходную позицию	Очищает статистику	Закрывает программу
5. Сколько практических заданий включает каждый модуль курса?	3 задания	10 заданий	5 заданий	1 итоговое задание
6. Какой педагогический эффект достигается через управление виртуальным исполнителем?	Ускорение печати	Формирование алгоритмического мышления через визуализацию	Заучивание команд	Изучение дизайна интерфейса
7. Что обязан сделать учитель перед началом работы с классом в программе?	Установить антивирус	Создать презентацию	Создать учебный класс в системе	Подключить интернет
8. Какая функция доступна учителю для массового добавления учеников?	Импорт данных из файла	Копирование списка из Word	Ручной ввод каждого ученика	Отправка приглашения по SMS
9. Для чего используется комбинация клавиш SHIFT + CTRL + ALT + T?	Закрытие программы	Открытие справки	Ускоренное прохождение уровней (режим проверки)	Перезагрузка системы
10. Какая методическая особенность отличает курс «Юный кодер»?	Постепенное усложнение материала с мини-тестами	Только теоретические лекции	Отсутствие практики	Использование платных библиотек

Модуль 3

Вопрос	Ответ 1	Ответ 2	Ответ 3	Ответ 4
--------	---------	---------	---------	---------

1. Какой номер задания ЕГЭ по информатике связан с обработкой строк?	17	24	26	27
2. В задании №26 ЕГЭ чаще всего требуется работа с:	Обработкой таблиц	Базами данных	Сортировкой и обработкой массивов	Созданием сайтов
3. Какой подход наиболее эффективен при обучении решению задания №27?	Сразу показывать готовое решение	Постепенно усложнять задачи и анализировать оптимизацию	Использовать только теорию без кода	Ограничиваться разбором примеров из оригинальных ЕГЭ
4. Основная трудность задания №27 заключается в:	Написании цикла for	Использовании print()	Необходимости оптимизации алгоритма	Подключении библиотек
5. Что важно при объяснении задания №24-27 ученикам?	Заучивание готового кода	Использование только списков	Пошаговый разбор логики обработки данных	Максимально короткое решение
6. Что такое Vibe-кодирование?	Новый язык программирования	Интуитивное и творческое освоение кода	Онлайн-платформа для ЕГЭ	Метод шифрования данных
7. Какую роль ИИ может выполнять в обучении программированию?	Помогать генерировать примеры и объяснения	Проверять только орфографию	Полностью заменять учителя	Использоваться только для развлечения
8. Какое преимущество использования ИИ при подготовке к ЕГЭ?	Увеличение количества терминов	Исключение самостоятельной работы	Быстрая генерация тренировочных заданий	Отмена необходимости учить теорию
9. Что важно учитывать при использовании ИИ в образовании?	Только скорость ответа	Только стоимость сервиса	Ограничения, риски и корректность информации	Использование только офлайн-режима
10. Что лежит в основе создания простого чат-бота с ИИ?	Работа с Excel	Использование API	Создание HTML-страницы	Установка антивируса

Оценка результатов:

Максимальная оценка – 10 баллов

Форма контроля	Зачтено	Незачтено
Зачет	7-10 баллов	Менее 7 баллов

Итоговая аттестация проводится по завершении освоения программы и включает 30 тестовых заданий. К каждому заданию предлагается несколько вариантов ответов, из которых слушателю необходимо выбрать один правильный.

Примеры вопросов для итогового контроля

Вопрос	Ответ1	Ответ2	Ответ3	Ответ4
1) Что выведет код? <code>print('2' + '3')</code>	23	5	Ошибка	2 3
2) Как правильно считать число и сохранить в переменную x?	<code>x = input(int())</code>	<code>x = int(input())</code>	<code>x = input()</code>	<code>x = float(int())</code>
3) В чём ключевое отличие = от ==?	== присваивает	= присваивает, == сравнивает	= — только для строк	отличий нет
4) Что произойдёт при выполнении кода? <code>a = [1,2,3]; print(a[3])</code>	Выведет 3	Ошибка (выход за границы списка)	Выведет 0	Выведет None
5) Как корректно проверить, что x лежит в диапазоне от 1 до 10 включительно?	<code>if x > 1 or x < 10:</code>	<code>if 1 < x < 10:</code>	<code>if x >= 1 or x <= 10:</code>	<code>if 1 <= x <= 10:</code>
6) Что выведет код? <code>for i in range(3): print(i)</code>	1 2 3	0 1 2	0 1 2 3	3 2 1
7) Какая строка создаёт пустой словарь?	<code>d = []</code>	<code>d = ()</code>	<code>d = "</code>	<code>d = {}</code>
8) Что вернёт вызов <code>len('abc')</code> ?	0	3	2	Ошибка
9) Какая запись корректно определяет функцию, возвращающую сумму a и b?	<code>def sum(a,b): print(a+b)</code>	<code>def sum(a,b): return a, b</code>	<code>def sum(a,b): return a + b</code>	<code>sum(a,b) = a+b</code>

10) Что важнее всего при первом объяснении циклов новичкам?	Сразу дать сложные задачи	Доказывать формулы	Показать «повтор действия» на понятных примерах и пошаговой трассировке	Дать только определения
11) Какой приём лучше всего снижает страх у новичков в «Юном кодере» на старте?	Дать 20 терминов на урок	Показать интерфейс и выполнить 1 задачу вместе пошагово	Сразу контрольная	Запретить подсказки
12) Почему пошаговый запуск особенно важен в обучении?	Ускоряет прохождение уровней	Помогает увидеть, на какой строке возникла ошибка и что делает робот	Убирает необходимость объяснять	Делает код короче
13) Что является обязательным шагом для учителя при организации обучения в системе?	Создать презентацию	Создать учебный класс и распределить учеников	Купить платный IDE	Установить игру
14) Какая функция упрощает первичное добавление большого списка учеников?	Экспорт данных	Ручной ввод	Импорт данных из файла	Сброс лицензии
15) Какой методический акцент правильнее при первых задачах на линейные алгоритмы?	Сразу требовать «самое короткое решение» и штрафовать	Сначала добиться понимания порядка команд, потом обсуждать оптимизацию	Вообще не обсуждать ошибки	Только переписывать готовый код
16) Что такое «команда» в контексте исполнителя?	Любая строка кода	Только input()	Только print()	Функция, написанная специально для исполнителя
17) Какая типичная	Использование range()	Перепутать направление в	Неправильный импорт библиотек	Слишком много функций

ошибка учеников при переносе предметов (get/set)?		<code>get_from_cell() / set_to_cell()</code>		
18) Какой подход лучше при «скучных повторях» в линейных заданиях (до циклов)?	Сказать: «терпите»	Сразу перейти к циклам без объяснений	Дать цель: найти более удобный маршрут/структуру действий и сравнить варианты	Убрать задания
19) Что корректнее сделать учителю, если ученик застрял на уровне?	Дать готовый код и идти дальше	Задать наводящие вопросы про цель и последовательность, предложить пошаговый запуск	Снизить оценку	Запретить подсказки
20) В чём ключевое педагогическое преимущество визуального исполнителя (робота)?	Ускоряет набор кода	Делает задания проще	Позволяет отказаться от теории	Позволяет увидеть результат алгоритма и связать код с действием
21) Какой вариант корректно сортирует список словарей students по ключу 'score' по убыванию?	<code>students.sort('score', reverse=True)</code>	<code>students.sort(key=lambda s: s['score'], reverse=True)</code>	<code>sorted(students, 'score')</code>	<code>students = students.sort(reverse=True)</code>
22) Как найти элемент с максимальным 'score' в списке словарей students?	<code>max(students)</code>	<code>students.max('score')</code>	<code>max(students, key=lambda s: s['score'])</code>	<code>max(key='score', students)</code>
23) Что делает параметр key= в sorted()/max()?	Меняет тип списка	Определяет, по какому значению сравнивать элементы	Ускоряет Python	Сортирует только числа

24) Задание №27 чаще всего требует от ученика:	Красивого вывода	Только знания синтаксиса	Оптимизации алгоритма по времени/памяти	Использование графики
25) Типичный признак «неоптимального» решения №27:	Использование функций	Вложенные циклы по большим данным без сокращения перебора	Наличие комментариев	Использование if
26) В задании №26 часто встречается работа с данными, где важно:	Оформление кода	Аккуратная обработка массива/набора значений и корректная сортировка/выбор	Только вывод строк	Использование turtle
27) Какой подход наиболее правильный при объяснении №24 (строки/последовательности)?	Разбирать пошагово: что считаем, по каким правилам, где обновляем ответ	Давать готовый шаблон без объяснений	Учить наизусть 1 решение	Сразу давать №27
28) Что является корректным и безопасным использованием ИИ на уроке?	ИИ решает за ученика, ученик переписывает	ИИ помогает генерировать похожие тренировочные задания и объяснения, а учитель проверяет корректность	ИИ ставит оценки без проверки	ИИ заменяет уроки
29) Что важнее всего проговорить при «Создании бота с использованием ИИ»?	Цвет интерфейса	Логика диалога + ограничения + проверку фактов + правила безопасности данных	Только как установить Python	Только как сделать мем
30) Что лучше всего описывает Vibe-кодинг в обучении?	Запрет на ошибки	Только «красивый код»	Подход, где важны вовлечённость, быстрый результат, эксперименты и рефлексия	Замена программирования презентациями

Оценка результатов:

Максимальное оценка – 30 баллов

Форма контроля	Зачтено	Незачтено
Зачет	22-30 баллов	Менее 22 баллов

3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

3.1. Материально-технические условия

Освоение программы предполагает использование слушателями персональных компьютеров или мобильных устройств с доступом в сеть Интернет и стандартным программным обеспечением (веб-браузер).

Наименование специализированных учебных помещений	Вид занятий	Наименование оборудования, программного обеспечения
-	-	-

3.2. Учебно-методическое и информационное обеспечение

1. Лутц М. Программирование на Python. Подробный толковый справочник / М. Лутц. — 5-е изд. — СПб.: Вильямс, 2019. — 1664 с. ISBN 978-5-97060-733-2
2. Дж. Зельцер (авт.-сост.) Изучаем Python / Дж. Зельцер, А. Свигарт. — М.: Питер, 2021. — 752 с. ISBN 978-5-4461-1358-9
3. Э. Маттис, Т. ДеИтани Python. Курс молодого бойца / Э. Маттис, Т. ДеИтани. — М.: Мир, 2020. — 864 с. ISBN 978-5-03-009798-0
4. Русаков А. В. Python — От новичка к профи / А. В. Русаков. — М.: ДМК Пресс, 2022. — 912 с. ISBN 978-5-97060-920-6
5. Макконнелл С. Совершенный код. Мастерство разработки / С. Макконнелл. — СПб.: Питер, 2020. — 960 с. ISBN 978-5-4461-1166-0
6. Python для ЕГЭ, Stepik - Бесплатный курс по Python [интернет-портал]. – Режим доступа: <https://stepik.org/course/67400/promo> (дата обращения: 12.02.2026).
7. Методическое пособие «Программирование на языке Python с использованием программы «Юный кодер» для школьников».

Электронные ресурсы

1. Visual Studio Code : IDE для работы с языками программирования : [интернет-портал]. – Режим доступа: <https://code.visualstudio.com/> (дата обращения: 12.02.2026).
2. Python.org : Интерпретатор языка программирования Python : [сайт]. – Режим доступа: <https://www.python.org/> (дата обращения: 12.02.2026).
3. ИТЗК.ру : Пакет программного обеспечения для обучения языкам программирования и основ алгоритмизации «Юный кодер» : [сайт]. – Режим доступа: <https://itzk.ru/> (дата обращения: 12.02.2026).

3.3. Кадровые условия

Кадровое обеспечение программы осуществляют высококвалифицированные специалисты, обладающие компетенциями в области информационных технологий, в том числе в области современной педагогики и частных методик преподавания информатики.

3.4. Условия для функционирования электронной информационно-образовательной среды (при реализации программ с использованием дистанционных образовательных технологий)

Электронные	Вид	Наименование оборудования,
-------------	-----	----------------------------

информационные ресурсы	занятий	программного обеспечения
https://edu.kpfu.ru/course/view.php?id=7496	Все виды (лекции, практические занятия, консультации, контроль)	Компьютер, подключенный к сети Интернет, интернет-браузер; Стабильное Интернет-соединение, актуальный веб-браузер.