



АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ  
**НАЦИОНАЛЬНЫЙ ЦЕНТР  
ИННОВАЦИЙ В ОБРАЗОВАНИИ**  
ИНН 7708241976, КПП 770801001, ОГРН 1147799018696

107045, Россия, Москва, ул. Сретенка, д. 24/2, стр. 1, Тел: +7(495)114-56-28, www.ncio.ru, E-mail: info@ncio.ru



**Программа  
дополнительного профессионального образования  
(повышение квалификации)**

**«Основы объектно-ориентированного  
программирования и планирования в JavaScript»**

Автор: А.А. Герасимова,

инженер-разработчик ООО «Экзамен-Технолаб»

Москва, 2023 г.

## Тема: «Основы объектно-ориентированного программирования и планирования в JavaScript»

### Раздел 1. «Характеристика программы»

#### 1.1. Цель реализации программы

**Цель:** совершенствование профессиональных компетенций слушателей в области изучения основ объектно-ориентированное программирование (ООП) и планирования в JavaScript для использования в образовательном процессе.

#### 1.2. Совершенствуемые компетенции

№ п/п	Компетенции	Направление подготовки Педагогическое образование 44.03.01 Бакалавриат
		Код компетенции
1.	Способен осуществлять педагогическую деятельность на основе специальных научных знаний	ОПК-8

#### 1.3. Планируемые результаты

№ п/п	Уметь-Знать	Направление подготовки Педагогическое образование 44.03.01 Бакалавриат
		Код компетенции
1.	<b>Уметь:</b> писать программы с использованием методологии ООП; создавать объекты и классы; применять наследование; использовать планирование (таймеры и интервалы). <b>Знать:</b> основы методологии объектно-ориентированного программирования; принципы и особенности создания объектов и классов, а также	ОПК-8

	наследования классов; синтаксис и особенности использования таймеров и интервалов.	
2.	<b>Уметь:</b> разрабатывать учебные занятия по изучению основ ООП и планирования в JavaScript. <b>Знать:</b> алгоритм разработки учебных занятий по изучению основ ООП и планирования в JavaScript.	ОПК-8

#### 1.4. Категория слушателей

*Уровень образования* – высшее образование;

*Направление подготовки* – педагогическое образование, преподаватели с базовым знанием языка программирования JavaScript или с опытом программирования на другом языке;

*Область профессиональной деятельности* – обучение на уровне общего образования, дополнительного образования.

1.5. **Форма обучения:** Заочная с ДОТ и ЭО.

1.6. **Режим занятий:** 3 раза на протяжении 6 недель.

1.7. **Трудоёмкость программы:** 29 часов.

## Раздел 2. «Содержание программы»

### 2.1. Учебный план

№ п/п	Наименование разделов/тем	Всего, внеаудит. часов	Внеаудиторные учебные занятия		Форма контроля	Трудоёмкость
			Онлайн-лекция	Практ. занятия		
1.	<b>Раздел 1. Повторение базы JavaScript</b>	4	3	1		4
1.1.	Примитивы, базовые и условные операторы.	1	1			1
1.2.	Циклы, массивы и их методы.	1	1			1
1.3.	Функции.	2	1	1	Тест № 1	2

<b>2.</b>	<b>Раздел 2. ООП в JavaScript.</b>	<b>13</b>	<b>4</b>	<b>9</b>		<b>13</b>
2.1.	Что такое ООП?	1	1			1
2.2.	Объекты JavaScript. В	3	1	2	Тест № 2	3
2.3.	Классы JavaScript. В	3	1	2	Тест № 3	3
2.4.	Наследование.	6	1	5	Тест № 4 Проект № 1	6
<b>3.</b>	<b>Раздел 3. Планирование.</b>	<b>8</b>	<b>4</b>	<b>4</b>		<b>8</b>
3.1.	Понятие асинхронности. Колбэк.	1	1			1
3.2.	Интервал.	1	1			1
3.3.	Таймер.	5	1	4	Тест № 5 Проект № 2	5
3.4.	Обзор на асинхронность в JavaScript.	1	1			1
<b>4.</b>	<b>Раздел 4. Разработка учебных занятий по основам ООП и планирования в JavaScript</b>	<b>3</b>	<b>1</b>	<b>2</b>		<b>3</b>
4.1.	Построение учебных занятий по основам ООП	1	1			1

	и планирования в JavaScript					
4.2.	Разработка учебного занятия по основам ООП и планирования в JavaScript	2		2	Проект № 3	2
5.	<b>Итоговая аттестация</b>	<b>1</b>		<b>1</b>	Зачёт по совокупности выполненных работ и итогового тестирования	<b>1</b>
<b>Итого:</b>		<b>29</b>	<b>12</b>	<b>17</b>		<b>29</b>

## 2.2. Календарный учебный график

Программа повышения квалификации реализуется в течение двух месяцев по мере комплектования группы в соответствии с календарным учебным графиком. Продолжительность реализации программы по графику – шесть недель с момента укомплектованности группы.

Наименование разделов/тем	Объём нагрузки, часов	Учебные недели					
		1 неделя	2 неделя	3 неделя	4 неделя	5 неделя	6 неделя
<b>Раздел 1. Повторение базы JavaScript</b>	<b>4</b>	3 часа (онлайн-лекция), 1 час (практика)	-	-	-	-	-
1.1. Примитивы, базовые и условные операторы.	1	1 час (онлайн-лекция)					
1.2. Циклы, массивы и их методы.	1	1 час (онлайн-лекция)					
1.3. Функции.	2	1 час (онлайн-лекция), 1 час (практика)					
<b>Раздел 2. ООП в JavaScript.</b>	<b>13</b>	-	2 часа (онлайн-)	2 часа (онлайн-)	3 часа (практика)	-	-

			лекция), 2 часа (практика)	лекция), 4 часа (практика)			
2.1. Что такое ООП?	1		1 час (онлайн-лекция)				
2.2. Объекты в JavaScript.	3		1 час (онлайн-лекция), 2 часа (практика)				
2.3. Классы в JavaScript.	3			1 час (онлайн-лекция), 2 часа (практика)			
2.4. Наследование.	6			1 час (онлайн-лекция), 2 часа (практика)	3 часа (практика)		
<b>Раздел 3. Планирование</b>	<b>8</b>	-	-	-	2 часа (онлайн-лекция)	2 часа (онлайн-лекция), 4 часа (практика)	-
3.1. Понятие асинхронности. Колбэк.	1				1 час (онлайн-лекция)		
3.2. Интервал.	1				1 час (онлайн-лекция)		
3.3. Таймер.	5					1 час (онлайн-лекция), 4 часа (практика)	
3.4. Обзор на асинхронность в JavaScript.	1					1 час (онлайн-лекция)	
<b>Раздел 4. Разработка учебных занятий по основам ООП и планирования в JavaScript</b>	<b>3</b>	-	-	-	-	-	1 час (онлайн-лекция), 2 часа (практика)
4.1. Построение учебных	1						1 час (онлайн-лекция)

занятий по основам ООП и планирования в JavaScript							
4.2. Разработка учебного занятия по основам ООП и планирования в JavaScript	2						2 часа (практика)
<b>5. Итоговая аттестация.</b>	<b>1</b>	-	-	-	-	-	1 час практика

### 2.3. Рабочая программа

№ п/п	Наименование разделов, тем	Виды учебных занятий, час	Содержание
<b>Раздел 1. Повторение базы JavaScript</b>			
1.1.	Примитивы, базовые и условные операторы.	Онлайн-лекция – 1 час	Переменные. Математические и логические операторы. Типы данных. Типы условных операторов. Ветвления. Операторы с дополнительным условием. Тернарный оператор. Среда разработки.
1.2.	Циклы, массивы и их методы.	Онлайн-лекция – 1 час	Циклы с постусловием и предусловием. Цикл с шагом. Понятие массива. Массивы и строки. Методы перебора. Сортировка, отбор и сведение.
1.3.	Функции.	Онлайн-лекция – 1 час	Функция и ее область видимости. Способы создания функции. Стрелочные функции.
		Практическое занятие – 1 час	Тест № 1. Выполнение тестирования по базе JavaScript.
<b>Раздел 2. ООП в JavaScript.</b>			
2.1.	Что такое ООП?	Онлайн-лекция – 1 час	Методология объектно-ориентированного программирования. Особенности и основные принципы.

2.2.	Объекты в JavaScript.	Онлайн-лекция – 1 час	Понятие объекта. Способы создания объектов. Запись свойств. Перебор свойств. Сравнение объектов. Передача по ссылке.
		Практическое занятие – 2 час	Практика на представленных в лекции примерах по объектам. Тест № 2. Выполнение тестирования по объектам.
2.3.	Классы в JavaScript.	Онлайн-лекция – 1 час	Понятие класса. Создание класса. Конструктор класса. Создание экземпляров класса.
		Практическое занятие – 2 часа	Практика на представленных в лекции примерах по классам. Тест № 3. Выполнение тестирования по классам.
2.4.	Наследование.	Онлайн-лекция – 1 час	Принципы наследования. Переопределение методов. Расширение методов.
		Практическое занятие – 5 часов	Практика на представленных в лекции примерах по наследованию. Тест № 4. Выполнение тестирования по наследованию и полиморфизму. Проект №1. Написание программы для решения задачи с использованием принципов ООП.
<b>Раздел 3. Планирование.</b>			
3.1.	Понятие асинхронности. Колбэк.	Онлайн-лекция – 1 час	Разница между синхронным и асинхронным кодом. Колбэк.
3.2.	Интервал.	Онлайн-лекция – 1 час	Понятие интервала. Синтаксис, особенности использования. Практика на представленных в лекции примерах по интервалам.



3.3.	Таймер.	Онлайн-лекция – 1 час	Таймер. Его синтаксис и особенности использования.
		Практическое занятие – 4 час	Практика на представленных в лекции примерах по таймерам.  Тест № 5. Выполнение тестирования по планированию в JS.  Проект №2. Написание программы для решения задачи с использованием планирования.
3.4.	Обзор на асинхронность в JavaScript.	Онлайн-лекция – 1 час	Как выполняется асинхронный код в JavaScript. Событийный цикл. Промисы.
<b>Раздел 4. Разработка учебных занятий по основам ООП и планирования в JavaScript</b>			
4.1.	Построение учебных занятий по основам ООП и планирования в JavaScript	Онлайн-лекция – 1 час	Особенности построения учебных занятий по изучению основам ООП и планирования в JavaScript
4.2.	Разработка учебного занятия по основам ООП и планирования в JavaScript	Практическое занятие – 2 часа	Проект №3.  Разработка учебного занятия по изучению основ ООП и планирования в JavaScript
5.	<b>Итоговая аттестация</b>	Практическое занятие – 1 час	Зачёт по совокупности выполненных работ и итогового тестирования.

### **Раздел 3. «Формы аттестации и оценочные материалы»**

Программой предусмотрены: промежуточный контроль – тесты и проекты; итоговая аттестация – итоговое тестирование, зачёты промежуточного контроля.

#### **3.1 Промежуточный контроль**

##### **Проекты:**

**Проект №1:** «Написание программы для решения задачи с использованием принципов ООП»

**Требования к выполнению проекта:** написать программный код на языке JavaScript, который будет соответствовать решению задачи, предлагаемой преподавателем. Ответ должен быть представлен в виде ссылки на решение.

**Критерии оценивания:**

1. Ответ представлен в требуемом виде.
2. Алгоритм оптимален и использует принципы ООП.
3. Задача решена в полной мере.

**Оценивание:**

Зачтено	Не зачтено
Выполнены все пункты оценочных материалов.	Не выполнен хотя бы один из оценочных пунктов.

**Проект №2:** «Написание программы для решения задачи с использованием планирования»

**Требования к выполнению проекта:** написать программный код на языке JavaScript, который будет соответствовать решению задачи, предлагаемой преподавателем. Ответ должен быть представлен в виде ссылки на решение.

**Критерии оценивания:**

1. Ответ представлен в требуемом виде.
2. Алгоритм оптимален и соответствует принципам ООП.
3. Правильно используется планирование.
4. Задача решена в полной мере.

**Оценивание:**

Зачтено	Не зачтено
Выполнены все пункты оценочных материалов.	Не выполнен хотя бы один из оценочных пунктов.

**Проект №3:** «Разработка учебного занятия по изучению основ ООП и планирования в JavaScript»

**Требования к выполнению проекта:** составить учебное занятие (тему выбирает слушатель) по изучению основ ООП и планирования в JavaScript, в которое должны быть включены, изученные технологии. Ответ должен быть представлен в виде файла в формате doc, docx или pdf.

**Критерии оценивания:**

1. Ответ представлен в требуемом виде.

2. Представлено учебное занятие по изучению основ ООП и планирования в JavaScript.

3. Содержание занятия соответствует одной или нескольким технологиям, изученным на занятиях пройденного учебного курса.

### Оценивание:

Зачтено	Не зачтено
Выполнены все пункты оценочных материалов.	Не выполнен хотя бы один из оценочных пунктов.

### Тесты:

#### Тест № 1 (образец);

1. Соотнесите обозначение оператора и его тип:

- |                             |                |
|-----------------------------|----------------|
| <input type="checkbox"/>    |                |
| <input type="checkbox"/> &  | Математический |
| <input type="checkbox"/> !! |                |
| <input type="checkbox"/> =  | Логический     |
| <input type="checkbox"/> == |                |
| <input type="checkbox"/> != | Присваивание   |
| <input type="checkbox"/> >= |                |
| <input type="checkbox"/> => |                |

2. Выберите из списка условные конструкции, существующие в языке JavaScript:

- if
- ?
- If...else
- Switch

3. Где верный вариант?

- let b = (a === 5) ? a\*2 ; a;
- let b = (a === 5) ? a\*2 : a;
- let b = (a === 5) ? a\*2 , a;
- let b = (a === 5) ? (a\*2 , a);

4. Выберите из списка циклы, существующие в языке JavaScript:

- for
- repeat
- do...while
- while
- loop

5. Выберите верный вариант:

- for (let i; i<10; i++) console.log(i);
- for (let i=0; i<10; i++){  
    console.log(i);  
}
- for (let i=0, i<10, i++){  
    console.log(i);  
}

6. Выберите метод, который запускает функцию для каждого элемента массива:

- map
- forEach
- filter
- reduce
- splice

7. Выберите метод, который вычисляет значение на основе всех элементов массива:

- map
- forEach
- filter
- reduce
- splice

8. Где объявлен массив?

- const a=[];
- let a = array(4, 5, 6)

`let a = new Array();`

9. Выберите неверный вариант объявления функции:

`function func(){  
    console.log('function')  
}`

`let func(param) => {  
    console.log('function')  
}`

`let func = (param) => {  
    console.log('function')  
}`

10. Что делает данная функция?

```

1  function func(n, result=n)
2  {
3      if(n>1)
4      {
5          result=result*(--n);
6          return func(n,result);
7      }
8      return result;
9  }
10

```

Возвращает факториал числа, переданного в параметрах

Выводит факториал числа, переданного в параметрах

Возвращает количество разрядов числа, переданного в параметрах

Выводит количество разрядов числа, переданного в параметрах

### Критерии оценивания:

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

### Оценивание:

Зачтено	Не зачтено
Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.	Правильные ответы слушателя составляют менее 75% вопросов компьютерного проверочного теста.

**Тест № 2 (образец);**

1. Что является ключом (ключами) в данном объекте?

```
1 let user = {  
2   name: "John",  
3   age: 30  
4 };
```

- user
- name
- age
- John
- 30

2. Выберите неверное утверждение для данного объекта:

```
1 let user = {  
2   name: "John",  
3   age: 30,  
4   "likes birds": true  
5 };
```

- Объект user имеет 3 свойства
- Значение свойства name - John
- В 4 строке ошибка: "likes birds" не может быть ключом

3. Введите команду для обращения к значению свойства age объекта user.

Укажите ответ без пробелов там, где их можно не ставить.

4. Введите команду для добавления в объект user свойства с ключом isTeacher и значением true. Укажите ответ без пробелов там, где их можно не ставить.

5. Что будет выведено в результате работы программы?

```
1 let user = {  
2   name: "Kate",  
3   age: 20  
4 };  
5  
6 console.log( age in user );
```

- true
- false
- 20

- Ошибка

**Критерии оценивания:**

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

**Оценивание:**

Зачтено	Не зачтено
Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.	Правильные ответы слушателя составляют менее 75% вопросов компьютерного проверочного теста.

**Тест № 3 (образец);**

1. Выберите верный вариант определения класса:

- class User {  
     constructor(name, age) {  
         this.name = name;  
         this.age = age;  
     }  
 }
- class User(name, age) {  
     this.name = name;  
     this.age = age;  
 }
- class User(name, age) {  
     name: name;  
     age: age;  
 }

2. Что такое класс?

- Расширяемый шаблон для написания функций
- Шаблон для создания методов объектов
- Расширяемый шаблон для создания объектов

3. Как создать новый экземпляр класса User с именем Kate?

```

01 class User {
02   constructor(name) {
03     this.name = name;
04   }
05 }

```

- new User({name: "Kate"})
- new User("Kate")
- new User(name = "Kate")

4. Что из этого метод класса?

```

class User {
  constructor(name) {
    this.name = name;
  }
  printName(){
    console.log(this.name)
  }
}

```

- user
- constructor
- name
- printName

5. Выберите верный вариант вызова метода:

```

1 class User {
2   constructor(name) {
3     this.name = name;
4   }
5   printName(){
6     console.log(this.name)
7   }
8 }
9
10 let user = new User('Kate');

```

- User.printName
- user.printName()
- user.printName

### Критерии оценивания:

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

### Оценивание:

Зачтено	Не зачтено
---------	------------



Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.

Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.

### Тест № 4 (образец);

1. С помощью какого слова осуществляется наследование?

- class
- extends
- constructor

2. «class Rabbit extends Animal» означает:

- Класс Animal наследует все свойства и методы класса Rabbit
- Класс Rabbit наследует все свойства и методы класса Animal

3. Мы создаем игру, где есть враги и герой. Они все имеют здоровье, скорость, силу атаки, могут перемещаться и атаковать. Но враги, в отличие от героя, имеют особый тип перемещения. Также существуют враги с особым типом атаки. Какие классы необходимо создать?

- Класс врага и класс героя
- Класс героя и наследующий его класс врага
- Класс персонажа и наследующие его классы врага и героя
- Класс врага и наследующий его класс особого врага

4. Что произойдет в результате работы данной программы?

```

1  class User {
2    constructor(name) {
3      this.name = name;
4    }
5    printName(){
6      console.log(this.name)
7    }
8  }
9
10 class Teacher extends User{
11   constructor(params){
12     super(params);
13     this.isTeacher = true;
14   }
15 }
16
17 let user = new Teacher('Kate');
18
19 user.printName()
```

- Ничего, так как у класса Teacher нет метода printName

- Будет выведено «Kate», так как класс Teacher наследует метод от класса User
- Ошибка, так как у класса Teacher нет метода printName
- Будет выведено «true», так как переопределили конструктор класса Teacher

5. Что будет выведено в результате работы программы?

```

1  class User {
2    constructor(name) {
3      this.name = name;
4    }
5    printName(){
6      console.log(this.name)
7    }
8  }
9
10 class Teacher extends User{
11  constructor(params){
12    super(params);
13    this.isTeacher = true;
14  }
15
16  printName(){
17    console.log('Учитель ' + this.name);
18  }
19 }
20
21 let user = new Teacher('Kate');
22
23 user.printName()

```

- Kate
- True
- Учитель
- Учитель Kate
- Учитель true

### Критерии оценивания:

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

### Оценивание:

Зачтено	Не зачтено
Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.	Правильные ответы слушателя составляют менее 75% вопросов компьютерного проверочного теста.

### Тест № 5 (образец);

1. Когда команды выполняются одна за другой, это:
  - Синхронный код
  - Асинхронный код

2. Если необходимо выполнить действие один раз с задержкой, необходимо использовать:

- setTimeout
- setInterval
- event loop

3. Если необходимо выполнять действие регулярно с определенной частотой, необходимо использовать:

- setTimeout
- setInterval
- event loop

4. Что первое будет выведено в результате работы программы?

```
1 function sayHi() {
2   console.log('Привет');
3 }
4
5 setTimeout(sayHi, 1000);
6 console.log('Пока');
```

- Привет
- Пока
- 1000
- Ошибка

5. Сколько раз будет выведено сообщение?

```
1 const message = 'Hello world'
2
3 function sayHi(){
4   console.log(message);
5 }
6
7 let timer = setInterval(sayHi, 2000);
8
9 setTimeout(() => { clearInterval(timer); console.log('stop'); }, 5000);
```

- 2
- 3
- 5

- Бесконечное число раз
- 0

### Критерии оценивания:

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

### Оценивание:

Зачтено	Не зачтено
Правильные ответы слушателя составляют не менее 75% вопросов компьютерного проверочного теста.	Правильные ответы слушателя составляют менее 75% вопросов компьютерного проверочного теста.

## 3.2. Итоговая аттестация

Зачёт по совокупности выполненных работ и итогового тестирования.

### Итоговое тестирование (образец):

1. Что выведет этот код?

```

1 let obj = {
2   "0": 1,
3   0: 2
4 };
5
6 alert( obj["0"] + obj[0] );

```

- 2
- 3
- 4
- 12
- Ошибка

2. Сколько раз будет выведено сообщение?

```

1 const message = 'Hello world'
2
3 function sayHi(){
4   console.log(message);
5 }
6
7 let timer = setInterval(sayHi, 2000);
8
9 setTimeout(() => { clearInterval(timer); console.log('stop'); }, 5000);

```

- 2

- 3
- 5
- Бесконечное число раз
- 0

3. Что будет выведено в результате работы программы?

```
1 class User {
2   constructor(name) {
3     this.name = name;
4   }
5   printName(){
6     console.log(this.name)
7   }
8 }
9
10 class Teacher extends User{
11   constructor(params){
12     super(params);
13     this.isTeacher = true;
14   }
15
16   printName(){
17     console.log('Учитель ' + this.name);
18   }
19 }
20
21 let user = new Teacher('Kate');
22
23 user.printName()
```

- Kate
- True
- Учитель
- Учитель Kate
- Учитель true

4. Что произойдет в результате работы данной программы?

```

1  class User {
2    constructor(name) {
3      this.name = name;
4    }
5    printName(){
6      console.log(this.name)
7    }
8  }
9
10 class Teacher extends User{
11   constructor(params){
12     super(params);
13     this.isTeacher = true;
14   }
15 }
16
17 let user = new Teacher('Kate');
18
19 user.printName()

```

- Ничего, так как у класса Teacher нет метода printName
  - Будет выведено «Kate», так как класс Teacher наследует метод от класса User
  - Ошибка, так как у класса Teacher нет метода printName
  - Будет выведено «true», так как переопределили конструктор класса Teacher
5. Введите команду для обращения к значению свойства age объекта user. Укажите ответ без пробелов там, где их можно не ставить.
6. Введите команду для добавления в объект user свойства с ключом isTeacher и значением true. Укажите ответ без пробелов там, где их можно не ставить.
7. Мы создаём игру, где есть враги и герой. Они все имеют здоровье, скорость, силу атаки, могут перемещаться и атаковать. Но враги, в отличие от героя, имеют особый тип перемещения. Также существуют враги с особым типом атаки. Какие классы необходимо создать?
- Класс врага и класс героя
  - Класс героя и наследующий его класс врага
  - Класс персонажа и наследующие его классы врага и героя
  - Класс врага и наследующий его класс особого врага
8. Что первое будет выведено в результате работы программы?

```

1  function sayHi() {
2      console.log('Привет');
3  }
4
5  setTimeout(sayHi, 1000);
6  console.log('Пока');

```

- Привет
- Пока
- 1000
- Ошибка

### Критерии оценивания:

Тест оценивается по системе зачтено/не зачтено. Ответ на вопрос теста должен соответствовать действительности.

### Оценивание:

Зачтено	Не зачтено
Все пункты задания выполнены.	Хотя бы один из пунктов не выполнен.

Обучающийся считается аттестованным:

**Критерии оценивания:** если обучающимся получены «зачёты» по итогам промежуточного контроля и итогового тестирования.

### Оценивание:

«**Зачтено**»: выполнены тесты № 1, № 2, № 3, № 4 и № 5, Проекты № 1, № 2, № 3, итоговое тестирование.

«**Не зачтено**»: не выполнен хотя бы один из пяти тестов или не сделан хотя бы один из трёх проектов, или итоговое тестирование.

## Раздел 4. Организационно-педагогические условия реализации программы

### 4.1. Учебно-методическое обеспечение и информационное обеспечение программы

#### Нормативно-правовые документы

1. Российская Федерация. Закон. Об образовании в Российской Федерации : Федеральный закон № 273-ФЗ : [принят Государственной думой 21 декабря 2012 года : одобрен Советом Федерации 26 декабря 2012 года ] : [Электронный ресурс] / Минпросвещения России (официальный интернет-ресурс) : <http://pravo.gov.ru/proxy/ips/?docbody=&nd=102162745> (дата обращения 01.09.2023). – Текст : электронный.
2. Российская Федерация. Приказ. Об утверждении и введении в действие федерального государственного образовательного стандарта среднего общего образования : приказ Министерства образования и науки РФ от 6 октября 2009 г. № 413 : [приложения утверждены приказом Министерства образования и науки РФ от 17 мая 2012 г. № 413] : [Электронный ресурс] / Реестр примерных основных общеобразовательных программ : <https://fgosreestr.ru/uploads/files/cc49b8b607ab29a7ea856f3a8cfd17d9.pdf> (дата обращения 01.09.2023). – Текст : электронный.
3. Российская Федерация. Приказ. Об утверждении федерального государственного образовательного стандарта основного общего образования : приказ Министерства образования и науки РФ от 31 мая 2021г. № 287 : [утверждён приказом Министерства образования и науки РФ от 31 мая 2021 г. № 287] : [Электронный ресурс] / Реестр примерных основных общеобразовательных программ : <https://fgosreestr.ru/uploads/files/238eb2e61e443460b65a83a2242abd57.pdf> (дата обращения 01.09.2023). – Текст : электронный.
4. Российская Федерация. Приказ. О внесении изменений в федеральный государственный образовательный стандарт среднего общего образования, утверждённый приказом Министерства образования и науки Российской Федерации от 17 мая 20 12 г. № 413 : приказ Министерства просвещения Российской Федерации от 12 августа 2022 г. № 732 : [приложения утверждены приказом Министерства образования и науки РФ от 17 мая 2012 г. № 413] : [Электронный ресурс] / Реестр примерных основных общеобразовательных программ : <https://fgosreestr.ru/uploads/files/3ecd094e3813dce94559978a8a95fc4e.pdf> (дата обращения 01.09.2023). – Текст : электронный.

### **Основная литература**

1. Флэнаган, Дэвид. Ф73 JavaScript. Полное руководство: пер. с англ. – СПб. : ООО «Диалектика», 2021. – 720 с.: ил. – парал. тит. англ.



## Дополнительная литература

1. Минник Крис, Холланд, Ева. М62 JavaScript для чайников. : пер. с англ. – М. : ООО «И.Д. Вильямс», 2017 - 321с. : ил. – парал. тит. англ.
2. Фримен Э., Робсон Э. Изучаем программирование на JavaScript. – СПб. : Питер, 2015. – 640 с. : ил. – (серия «Head First O'Reilly»)

### 4.2. Материально-технические условия реализации программы

Наименование аудиторий	Вид занятий	Наименование оборудования, программного обеспечения
Рабочие места, оборудованные для проведения онлайн-лекций.	Онлайн-лекция / практические занятия	Компьютеры (ноутбуки) с доступом в интернет, 1 компьютер на каждого слушателя

### 4.3. Кадровые условия реализации программы

Количество преподавателей, привлечённых для реализации дополнительной профессиональной программы повышения квалификации, – не менее одного человека: может быть штатным сотрудником или внешним совместителем, обладающим компетенциями в области программирования и знанием языка программирования JavaScript.