

Министерство образования и науки Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования «Казанский национальный исследовательский  
технический университет им. А.Н. Туполева-КАИ»  
Институт Компьютерных технологий и защиты информации  
Кафедра Компьютерных систем

УТВЕРЖДАЮ

Ответственный за ОП

Верш И.С.Вершинин

«31» 08 2017г.

Регистрационный номер 4010-17/5-052

**ФОНДОЦЕНОЧНЫХ СРЕДСТВ**

для проведения промежуточной аттестации обучающихся по дисциплине

**Программирование на языках высокого уровня**  
(наименование дисциплины, практики)

Индекс по учебному плану: **Б1.Б.14**

Направление подготовки: **09.03.01 «Информатика и вычислительная техника»**

Квалификация: **бакалавр**

Профиль подготовки: **Вычислительные машины, комплексы, системы и сети, Автоматизированные системы обработки информации и управления, Программное обеспечение средств вычислительной техники и автоматизированных систем, Системы автоматизированного проектирования (электронные средства), Системы автоматизированного проектирования машиностроения**

Вид(ы) профессиональной деятельности: **научно-исследовательская; проектно-конструкторская**

Заведующий кафедрой С.С.Зайдуллин

Разработчик: доцент кафедры ПМИ П.И.Тутубалин

Казань 2017 г.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

**Программирование на языках высокого уровня**  
(наименование дисциплины, практики)

Содержание фонда оценочных средств (ФОС) соответствует требованиям федерального государственного стандарта высшего образования (ФГОС ВО) по направлению 09.03.01 «Информатика и вычислительная техника», учебному плану направления 09.03.01 «Информатика и вычислительная техника».

Разработанные ФОС обладают необходимой полнотой и являются актуальными для оценки компетенций, осваиваемых обучающимися при изучении дисциплины «Введение в теорию принятия решений». Разработанные ФОС полностью соответствуют задачам будущей профессиональной деятельности обучающихся, установленных ФГОС ВО по направлению 09.03.01 «Информатика и вычислительная техника». В составе ФОС присутствуют оценочные средства в виде тестовых заданий и контрольных вопросов различного уровня сложности, которые позволяют провести оценку порогового, продвинутого и превосходного уровней освоения компетенций по дисциплине.

ФОС обладают необходимой степенью приближенности к задачам будущей профессиональной деятельности обучающихся, связанным со способностью обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности (ОПК-6).

Существенные недостатки отсутствуют.

Заключение. Учебно-методическая комиссия делает вывод о том, что представленные материалы соответствуют требованиям ФГОС ВО по направлению 09.03.01 «Информатика и вычислительная техника» и рекомендуются для использования в учебном процессе.

Рассмотрено на заседании учебно-методической комиссии института КТЗИ от 31 августа 2017 г., протокол №.8 .

Председатель УМК института КТЗИ



В.В. Родионов

## Содержание

<b>ВВЕДЕНИЕ</b>	<b>4</b>
<b>1. ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ</b>	<b>5</b>
<b>2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ</b>	<b>5</b>
<b>3. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b>	<b>5</b>
<b>4. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЯ ШКАЛЫ ОЦЕНИВАНИЯ</b>	<b>5</b>
<b>5. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРУ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ</b>	<b>8</b>
<b>6 КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b>	<b>10</b>
<b>ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ</b>	<b>23</b>

## **Введение**

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Программирование на языках высокого уровня» – это комплект методических и контрольно-измерительных материалов, предназначенных для определения уровня сформированности компетенций, оценивания знаний, умений, владений на разных этапах освоения дисциплины для проведения промежуточной аттестации обучающихся по дисциплине.

ФОС ПА является составной частью учебного и методического обеспечения программы бакалавриата по направлению 09.03.01 «Информатика и вычислительная техника».

Задачи ФОС по дисциплине «Программирование на языках высокого уровня»:

- оценка запланированных результатов освоения дисциплины обучающимися в процессе изучения дисциплины, в соответствии с разработанными и принятыми критериями по каждому виду контроля;

- контроль и управление процессом приобретения обучающимися необходимых знаний, умений, навыков и формирования компетенций, определенных в ФГОС ВО по направлению подготовки

ФОС ПА по дисциплине «Программирование на языках высокого уровня» сформирован на основе следующих основных принципов оценивания:

- пригодности (валидности) (объекты оценки соответствуют поставленным целям обучения);

- надежности (использования единообразных стандартов и критериев для оценивания запланированных результатов);

- эффективности (соответствия результатов деятельности поставленным задачам).

ФОС ПА по дисциплине «Программирование на языках высокого уровня» разработан в соответствии с требованиями ФГОС ВО по направлению 09.03.01 «Информатика и вычислительная техника» для аттестации обучающихся на соответствие их персональных достижений требованиям поэтапного формирования соответствующих составляющих компетенций и включает контрольные вопросы (или тесты) и типовые задания, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций.

### **1. Формы промежуточной аттестации по дисциплине**

Дисциплина «Программирование на языках высокого уровня» изучается в 8 семестре при очной форме обучения и завершается промежуточной аттестацией в форме экзамена.

### **2. Оценочные средства для промежуточной аттестации**

Оценочные средства для промежуточной аттестации по дисциплине «Программирование на языках высокого уровня» при очной форме обучения.

Таблица 1

Оценочные средств для промежуточной аттестации (очная форма обучения)

№ п/п	Семестр	Форма промежуточной аттестации	Оценочные средства
1.	2	Экзамен	ФОС ПА

### **3. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины**

Перечень компетенций и их составляющих, которые должны быть сформированы при изучении темы соответствующего раздела дисциплины «Программирование на языках высокого уровня», представлен в таблице 2.

Таблица 2

Перечень компетенций и этапы их формирования  
в процессе освоения дисциплины

№ п/п	Этап формирования (семестр)	Наименование раздела	Код формируемой компетенции (составляющей компетенции)		Форма промежуточной аттестации
			ОПК-6	ОПК-6.ЗУВ	
1.	2	Язык высокого уровня Си (компилируемый)	ОПК-6	ОПК-6.ЗУВ	Экзамен
2.	2	Язык высокого уровня JavaScript (интерпретируемый)	ОПК-6	ОПК-6.ЗУВ	Экзамен

### **4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкалы оценивания**

Показатели и критерии оценивания сформированности компетенций на зачете, приведены в таблице 3.

Таблица 3

## Показатели и критерии оценивания сформированности компетенций на экзамене

№ п/п	Этап формирования (семестр)	Код формируемой компетенции (составляющей компетенции)		Критерии оценивания	Показатели оценивания (планируемые результаты обучения)		
					Пороговый уровень	Продвинутый уровень	Превосходный уровень
1.	2	ОПК-6	ОПК-6.3	Теоретические навыки	Знать подходы к разработке алгоритмических и программных решений в области системного и прикладного программирования	Знать подходы к разработке алгоритмических и программных решений в области математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей	Знать подходы к разработке алгоритмических и программных решений в области образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям

2.	2	ОПК-6	ОПК-6.У	Практические навыки	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных решений в области системного и прикладного программирования	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных решений в области математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных решений в области образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям
----	---	-------	---------	---------------------	---	--	--

Формирование оценки при промежуточной аттестации по итогам освоения дисциплины зависит от уровня освоения компетенций, которые обучающийся должен освоить по данной дисциплине. Связь между итоговой оценкой и уровнем освоения компетенций (шкала оценивания) представлена в таблице 4.

Таблица 4

Описание шкалы оценивания

Шкала оценивания		Описание оценки в требованиях к уровню и объему компетенций
Словесное выражение	Выражение в баллах	
Отлично	от 86 до 100	Освоен <b>превосходный</b> уровень всех компетенций (составляющих компетенций)
Хорошо	от 71 до 85	Освоен <b>продвинутый</b> уровень всех компетенций (составляющих компетенций)
Удовлетворительно	от 51 до 70	Освоен <b>пороговый</b> уровень всех компетенций (составляющих компетенций)
Не удовлетворительно	до 51	Не освоен <b>пороговый</b> уровень всех компетенций (составляющих компетенций)

**5. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Формирование оценки по результатам текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Программирование на языках высокого уровня» приведено в таблице 5.

Таблица 5 Формирование оценки по итогам освоения дисциплины

Наименование контрольного мероприятия	Рейтинговые показатели			
	I аттестация	II аттестация	по результатам текущего кон- троля	по итогам промежуточной аттестации (зачета /экзамена)
<b>Раздел 1</b>	<b>31</b>		<b>31</b>	
Тест текущего контроля по разделу	10		10	
Защита лабораторных работ	21		21	
<b>Раздел 2.</b>		<b>33</b>	<b>33</b>	
Тест текущего контроля по разделу		10	10	
Защита лабораторных работ		23	23	
<b>Промежуточная аттестация (Экзамен):</b>				<b>36</b>
– тест промежуточной аттестации по дис- циплине				20
– ответы на экзаменационные вопросы в письменной форме				16

**6 Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины**

**6.1 Тестовые задания**

1.

```
var a = 1;  
var b = ++a+a;  
alert(b);
```

Результат работы кода - сообщение с текстом

1

2

4

Этот код не будет работать

2.

```
var a = 1;  
var b = { toString:function(){return '1'} };  
alert(a+b);
```

Результат работы кода - сообщение с текстом

1[object Object]

2

11

Код не работает

3.

```
var a = Array();  
for(var i = 0;i<10;i++)  
a[i] = function(){return i};  
alert(a[3]());
```

Результат работы кода - сообщение с текстом

10

3

2

i

Другое

4.

```
var a = 1;  
var b = a++;  
alert(b);
```

Результат работы кода - сообщение с текстом

1

2

0

Другое

5.

```
var a = 1;  
{
```

```
var a = 5;  
}  
alert(a);
```

Результат работы кода - сообщение с текстом

1  
5  
Другое  
0

6. Чему будет равно z?

```
int z, x=5, y=-10, a=4, b=2;  
z = x++ - --y * b / a;
```

5  
6  
10  
11  
12

7. Какую функцию следует использовать в паре с функцией, выделяющей память?

unalloc()  
dropmem()  
dealloc()  
release()  
free()

8. Как правильно приростить "ptr"?

```
void *ptr;
myStruct myArray[10];
ptr = myArray;

ptr = ptr + sizeof(struct myStruct);
++(int*)ptr;
ptr = ptr + sizeof(myArray);
increment(ptr);
ptr = ptr + sizeof(ptr);
```

9. Что будет напечатано?

```
char* myFunc (char *ptr)
{
    ptr += 3;
    return (ptr);
}

int main()
{
    char *x, *y;
    x = "HELLO";
    y = myFunc (x);
    printf ("y = %s \n", y);
    return 0;
}
```

y = HELLO

```
y = ELLO
y = LLO
y = LO
x = O
```

10. Какой из вариантов описания следующего кода отражает его работу?

```
struct node *nPtr, *sPtr; /* указатели на связный список. */
for (nPtr = sPtr; nPtr; nPtr = nPtr -> next)
{
    free(nPtr);
}
```

Будут перебираться элементы списка до тех пор пока они не закончатся.

Может вызвать ошибку, т.к. память где размещается элемент списка "nPtr" высвобождается преждевременно.

В цикле следует заменить запись "nPtr = nPtr -> next" на "nPtr = nPtr.next" для устранения ошибки "Segmentation fault".

Применён неверный синтаксис освобождения памяти.

Цикл никогда не завершится.

## 6.2. Экзаменационные вопросы

1. Основные понятия об языках высокого уровня: классификация.
2. Основные понятия об языках высокого уровня: алгоритмика, алгоритм, программное обеспечение, система программирования.
3. Основные понятия об языках высокого уровня: язык программирования, транслятор, компилятор, исходный модуль, интерпретатор, программирование, отладка программы, тестирование, тест, верификация, операция, операнд.

4. Способы записи алгоритма, тестирование, тест, верификация, операция, операнд.
5. Си: побитовые операторы, примеры.
6. Си: приоритет и очередность вычислений, примеры.
7. Си: области видимости.
8. Си: заголовочные файлы, примеры.
9. Си: статические переменные, примеры.
10. Си: регистровые переменные, примеры.
11. Си: блочная структура, примеры.
12. Си: инициализация, примеры.
13. Препроцессор языка Си. Включение файла. Макроподстановка.
14. Препроцессор языка Си. Макроподстановка.
15. Препроцессор языка Си. Условная компиляция.
16. Си: указатели и массивы.
17. Си: адресная арифметика. Примеры, основные моменты, допустимые операции.
18. Си: символьные указатели. Примеры, основные моменты.
19. Си: массивы указателей, указатели на указатели, примеры.
20. Си: многомерные массивы, примеры.
21. Си: инициализация массивов указателей, сравнение указателей и многомерных массивов, примеры.
22. Си: аргументы командной строки, примеры.
23. Си: указатели на функции, примеры.
24. Си: сложные объявления, примеры.
25. Си: структуры, основные сведения о структурах, примеры.
26. Си: структуры и функции, примеры.
27. Си: массивы структур, примеры.
28. Си: указатели на структуры, примеры.
29. Си: структуры со ссылками на себя, примеры.
30. Си: средство typedef, примеры.

- 31.Си: объединения, примеры.
- 32.Си: битовые поля, примеры.
- 33.Си: списки аргументов функций переменной длины, примеры.
- 34.Си: управление ошибками (stderr и exit), примеры.
- 35.Си: произвольный доступ (fseek), примеры.
- 36.Си: стандартная библиотека (ввод-вывод: <stdio.h>, проверки класса символа: <ctype.h>, функции, оперирующие со строками: <string. h>, математические функции: <math. h>, функции общего назначения: <stdlib. h>, диагностика: <assert. h>), примеры функций (их назначение).
- 37.Си: стандартная библиотека (списки аргументов переменной длины: <stdarg.h>, дальние переходы: <setjmp. h>, сигналы: <signal. h>, функции даты и времени: <time.h>, зависящие от реализации пределы: <limits.h> и <float.h>), примеры функций (их назначение).
- 38.Ведение в JavaScript. Два мифа связанных с JS. Клиентский JavaScript, пример.
- 39.Ведение в JavaScript. Клиентский JavaScript, пример. Проверка своих сценариев.
- 40.JavaScript: набор символов, чувствительность к регистру, символы разделители и переводы строк, необязательные точки с запятой. Примеры.
- 41.JavaScript: комментарии, литералы, идентификаторы. Примеры.
- 42.JavaScript: зарезервированные слова. Примеры.
- 43.JavaScript: типы данных и значения. Примеры.
- 44.JavaScript: числа, целые литералы, шестнадцатеричные и восьмеричные литералы. Примеры.
- 45.JavaScript: числа, литералы вещественных чисел, работа с числами, специальные числовые значения. Примеры.
- 46.JavaScript: строки, строковые литералы, управляющие последовательности в строковых литералах. Примеры.

- 47.JavaScript: строки, работа со строками, преобразование чисел в строки.  
Примеры.
- 48.JavaScript: строки, преобразование строк в числа. Примеры.
- 49.JavaScript: логические значения, преобразование логических значений.  
Примеры.
- 50.JavaScript: функции, функциональные литералы. Примеры.
- 51.JavaScript: объекты, создание объектов. Примеры.
- 52.JavaScript: объекты, объектные литералы. Примеры.
- 53.JavaScript: объекты, преобразование объектов. Примеры.
- 54.JavaScript: массивы, создание массивов. Примеры.
- 55.JavaScript: массивы, литералы массивов. Примеры.
- 56.JavaScript: специальные значения — null, undefined. Примеры.
- 57.JavaScript: преобразование типов — Таблица. Примеры.
- 58.JavaScript: объекты обертки для элементарных типов данных. Примеры.
- 59.JavaScript: обращение по значению и по ссылке — Таблица. Примеры.
- 60.JavaScript: элементарные и ссылочные типы. Примеры.
- 61.JavaScript: копирование и передача строк. Примеры.
- 62.JavaScript: сравнение строк, порядок проведения операций над типами данных в JavaScript — Таблица. Примеры.
- 63.JavaScript: переменные, типизация переменных. Примеры.
- 64.JavaScript: объявление переменных, повторные и опущенные объявления. Примеры.
- 65.JavaScript: область видимости переменной, отсутствие блочной области видимости. Примеры.
- 66.JavaScript: отсутствие блочной области видимости. Примеры.
- 67.JavaScript: неопределенные и неинициализированные переменные.  
Примеры.
- 68.JavaScript: сборка мусора. Примеры.
- 69.JavaScript: переменные как свойства, глобальный объект. Примеры.

- 70.JavaScript: переменные как свойства, локальные переменные объект вызова. Пример.
- 71.JavaScript: переменные как свойства, контексты исполнения в JavaScript. Примеры.
- 72.JavaScript: переменные как свойства, цепочка областей видимости и разрешения имени переменной. Примеры.
- 73.JavaScript: операторы, количество операндов, тип операндов. Примеры.
- 74.JavaScript: операторы, приоритет операторов, ассоциативность операторов. Примеры.
- 75.JavaScript: операторы, равенство (==). Примеры.
- 76.JavaScript: операторы, идентичность (===). Примеры.
- 77.JavaScript: операторы, неравенство (!=). Примеры.
- 78.JavaScript: операторы, неидентичность (!==). Примеры.
- 79.JavaScript: операторы, оператор !, операторы отношения, операторы сравнения. Примеры.
- 80.JavaScript: операторы, оператор in. Примеры.
- 81.JavaScript: оператор, оператор instanceof. Примеры.
- 82.JavaScript: операторы, подробнее о строковых операторах. Примеры.
- 83.JavaScript: операторы, логические операторы, логическое И (&&), логическое ИЛИ (||). Примеры.
- 84.JavaScript: операторы, оператор typeof, оператор создания объекта (new). Примеры.
- 85.JavaScript: операторы, оператор delete, оператор void. Примеры.
- 86.JavaScript: инструкции, инструкция for/in, метки, break, continue. Примеры.
- 87.JavaScript: инструкции, инструкция var, инструкция function, инструкция return. Примеры.
- 88.JavaScript: инструкции, инструкция throw. Примеры.
- 89.JavaScript: инструкции, инструкция try/catch/finally. Примеры.

90. JavaScript: инструкции, инструкция with, пустая инструкция. Примеры.
91. JavaScript: объекты, создание объектов. Примеры.
92. JavaScript: объекты, свойства объектов, перечисление свойств. Примеры.
93. JavaScript: объекты, проверка существования свойств, удаление свойств. Примеры.
94. JavaScript: объекты, объекты как ассоциативные массивы. Примеры.
95. JavaScript: свойства и методы универсального класса Object — свойство constructor, метод toString(), метод toLocaleString(), метод valueOf(). Примеры.
96. JavaScript: свойства и методы универсального класса Object — метод hasOwnProperty(), метод propertyIsEnumerable(), метод isPrototypeOf(). Примеры.
97. JavaScript: массивы, добавление элементов в массив, удаление элементов массива. Примеры.
98. JavaScript: массивы, длина массива, усечение и увеличение массива. Пример.
99. JavaScript: массивы, многомерные массивы. Примеры.
100. JavaScript: массивы, метод join(). Примеры.
101. JavaScript: массивы, метод reverse(). Примеры.
102. JavaScript: массивы, метод sort(). Примеры.
103. JavaScript: массивы, метод concat(). Примеры.
104. JavaScript: массивы, метод slice(). Примеры.
105. JavaScript: массивы, метод splice(). Примеры.
106. JavaScript: массивы, методы push() и pop(). Примеры.
107. JavaScript: массивы, методы unshift() и shift(). Примеры.
108. JavaScript: массивы, методы toString() и toLocaleString(), дополнительные методы массивов. Примеры.
109. JavaScript: функция, определение и вызов функций. Примеры.

110. JavaScript: функция, вложенные функции, функциональные литералы. Примеры.
111. JavaScript: функции, именование функций, аргументы функций, необязательные аргументы. Примеры.
112. JavaScript: функции, списки аргументов переменной длины: объект Arguments. Примеры.
113. JavaScript: функции, объект Arguments, свойство callee. Примеры.
114. JavaScript: функции, типы аргументов. Примеры.
115. JavaScript: функции, функции как данные. Примеры.
116. JavaScript: функции, функции как методы. Примеры.
117. JavaScript: свойства и методы функций, свойство length. Примеры.
118. JavaScript: свойства и методы функций, определение собственных свойств функций. Примеры.
119. JavaScript: свойства и методы функций, методы apply() и call(). Примеры.
120. JavaScript: классы и объекты, конструкторы. Примеры.
121. JavaScript: классы и объекты, прототипы и наследование. Примеры.
122. JavaScript: классы и объекты, чтение и запись унаследованных свойств, расширение встроенных типов. Примеры.
123. JavaScript: классы, объектно ориентированный язык JavaScript. Примеры.
124. JavaScript: классы, частные члены. Примеры.
125. JavaScript: классы, надклассы и подклассы, вызов переопределенных методов. Примеры.
126. JavaScript: модули и пространства имен, соглашения, создание модулей и пространств имен. Примеры.
127. JavaScript: модули и пространства имен, соглашения, проверка доступности модуля. Примеры.
128. JavaScript: модули и пространства имен, инициализация модуля. Примеры.

129. JavaScript: модули и пространства имен, импорт символов из пространств имен. Примеры.
130. JavaScript: модули и пространства имен, общедоступные и частные символы. Примеры.
131. JavaScript: модули и пространства имен, замыкания как частные пространства имен. Примеры.
132. JavaScript: модули и пространства имен, форматы модулей, загрузчики модулей, сборщик модулей.
133. JavaScript: шаблоны и регулярные выражения, определение регулярных выражений. Примеры.
134. JavaScript: шаблоны и регулярные выражения, символы литералов. Примеры.
135. JavaScript: шаблоны и регулярные выражения, классы символов. Примеры.
136. JavaScript: шаблоны и регулярные выражения, повторение. Примеры.
137. JavaScript: шаблоны и регулярные выражения, «нежадное» повторение. Примеры.
138. JavaScript: шаблоны и регулярные выражения, альтернативы, группировка и ссылки. Примеры.
139. JavaScript: шаблоны и регулярные выражения, задание позиции соответствия. Примеры.
140. JavaScript: шаблоны и регулярные выражения, флаги (наиболее часто используемые). Примеры.
141. JavaScript: шаблоны и регулярные выражения, методы класса String для поиска по шаблону. Примеры.
142. JavaScript: шаблоны и регулярные выражения, объект RegExp. Примеры.

143. JavaScript: шаблоны и регулярные выражения, методы класса RegExp для поиска по шаблону, свойства экземпляра RegExp. Примеры.

## Лист регистрации изменений и дополнений

№ п/п	№ страницы внесения изменений	Дата внесения изменения	Краткое содержание изменений (основание)	Ф.И.О., подпись	«Согласовано» заве- дующий кафедрой, ведущей дисциплину