

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 4080 -  
17/И - 189

### ФОНДОЦЕНОЧНЫХ СРЕДСТВ

для проведения промежуточной аттестации обучающихся по дисциплине  
«Технологии программирования на платформе Java»

Индекс по учебному плану: Б1.В.ДВ.01.02

Направление подготовки: 09.04.01 «Информатика и вычислительная техника»

Квалификация: магистр

Магистерская программа: Разработчик-программист (информатика как вторая компетенция)

Вид(ы) профессиональной деятельности: научно-исследовательская

Заведующий кафедрой ПМИ С.С.Зайдуллин

Разработчик доцент кафедры ПМИ, к.т.н. П.И. Тутубалин

Казань 2017 г.

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

**Технологии программирования на платформе Java**

(наименование дисциплины)

Содержание фонда оценочных средств (ФОС) соответствует требованиям федерального государственного стандарта высшего образования (ФГОС ВО) по направлению 09.04.01 «Информатика и вычислительная техника», учебному плану направления 09.04.01 «Информатика и вычислительная техника».

Разработанные ФОС обладают необходимой полнотой и являются актуальными для оценки компетенций, осваиваемых обучающимися при изучении дисциплины «Технологии программирования на платформе Java». Разработанные ФОС полностью соответствуют задачам будущей профессиональной деятельности обучающихся, установленных ФГОС ВО по направлению 09.04.01 «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 с применением перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий (ПК-7).

Существенные недостатки отсутствуют.

Заключение. Учебно-методическая комиссия делает вывод о том, что представленные материалы соответствуют требованиям ФГОС ВО по направлению 09.04.01 «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>24</b>

## **Введение**

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Технологии программирования на платформе Java» – 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 магистратуры по направлению 09.04.01 «Информатика и вычислительная техника».

Задачи ФОС по дисциплине «Технологии программирования на платформе Java»:

- 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 «Технологии программирования на платформе Java» 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 «Технологии программирования на платформе Java» разработан в соответствии с требованиями ФГОС ВО по направлению 09.04.01 «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 «Технологии программирования на платформе Java» изучается в 6 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 «Технологии программирования на платформе Java» при очной форме обучения.

Таблица 1

Оценочные средств для промежуточной аттестации (очная форма обучения)

№ п/п	Семестр	Форма промежуточной аттестации	Оценочные средства
1.	3	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 «Технологии программирования на платформе Java», 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
1.	6	Клинтсткие средства Java	ПК-7	ПК-7.ЗУВ	Экзамен
2.	6	Серверные средства Java	ПК-7	ПК-7.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.	3	ПК-7	ПК-7.3	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.	3	ПК-7	ПК-7.У	Практические навыки	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных ре-	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных ре-	Уметь применять при практической разработке программного обеспечения подходы к разработке алгоритмических и программных ре-

					шений в области системного и прикладного программирования	шений в области математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей	шений 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 «Технологии программирования на платформе Java» 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 2</b>	<b>31</b>		<b>31</b>	
Тест текущего контроля по разделу	10		10	
Защита лабораторных работ	21		21	
<b>Раздел 3.</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. Какой результат выполнения программы?

```
public class B {  
    public static void main(String[] args){  
        big_loop: for(int i=0;i<3;i++){  
            try{  
                for(int j=0;j<3;j++){  
                    if(i==j) continue;  
                    else if(i>j)continue big_loop;  
                    System.out.print("A");  
                }  
            }  
            finally {  
                System.out.print("B");  
            }  
            System.out.print("C");  
        }  
    }  
}
```

- ABVCCA
- AAVCBV
- AACACAVVCSACA
- AAACA
- ACVVCSA

2. Какой результат выполнения кода?

```
Integer i=5000;
```

```
System.out.println(i.hashCode());
```

- 0000A5F8
- 5000
- Ошибка компиляции

3. После выполнения какой строки только один объект будет доступен для сборки мусора (Garbage Collection)?

```
01 public class Q3 {  
02     Q3 ags = null;  
03     public static void main(String[] args){  
04         Q3 a1 = new Q3();  
05         Q3 a2 = new Q3();  
06         Q3 a3 = new Q3();  
07         a1.ags = new Q3();  
08         a2.ags = a1.ags;  
09         a3.ags = a2.ags;  
10         a1 = null;  
11         a2 = null;  
12         a3 = null;  
13     }  
14 }
```

- 7
- 8
- 9
- 10
- 11
- 12
- 13

- 14
- В этом методе ни один объект не может быть уничтожен сборщиком мусора

4. Какой результат выполнения данного кода?

```
public class Starter extends Thread{
    private int x;
    public static void main(String[] args) throws Exception{
        new Starter().makeItSo();
    }
    public Starter(){
        x=5;
        start();
    }
    public void makeItSo() throws Exception{
        join();
        x=x-1;
        System.out.println(x);}
    public void run(){x*=2;}
}
```

- 4
- 5
- 8
- 9
- Ошибка компиляции
- Ошибка выполнения
- Нет возможности определить вывод

5. В какой из строк приведён верный код?

```
public class Q5{
    public static void main(String[] args){
```

```
int x,y,z;
x=1; y=2; z=3;
System.out.println(z>=y>=x);//1
System.out.println(z>=y&&y>=x);//2
}}
```

- 1
- 2

6. Какие из перечисленных идентификаторов являются корректными?

- `_1c$ac`
- `$ac`
- `_cb`
- `int`
- `if_`
- `abc`
- `1ab`

7. Что произойдёт в результате компиляции и выполнения программы?

```
public class Main {
    public static void main(String[] args){
        String strA="text";
        String strB="text";
        strA+="1 ";
        strB+="1 ";
        System.out.print(strA!=strB);
        System.out.print(";");
        strA="text1 ";
        strB="text1 ";
        System.out.print(strA!=strB);
    }
}
```

```
}}
```

- true;false
- false;false
- true>true
- false>true

8. Что произойдёт после компиляции кода и запуска программы без аргументов?

```
public class Q8 {  
    public static void main(String[] args){  
        System.out.println(args.length);  
    }  
}
```

- Код не скомпилируется
- Код не скомпилируется, но во время исполнения будет выброшено исключение `NullPointerException`
- Код скомпилируется, а на консоль будет выведено число 0
- Код скомпилируется, но ничего не выведется на консоль

9. Что напечатает следующий код?

```
public class Q9 {  
    private String name;  
    Q9(String name){  
        this.name=name;}  
    public void test(){  
        test=new Test("Three");}  
    public String toString(){  
        return name;}  
    public static void main(String[] args){
```

```
Q9 a1=Q9("One");  
Q9 a2=Q9("Two");  
a1.test(a2);  
System.out.println(a2);}}
```

- Ошибка компиляции
- Ошибка выполнения
- One
- Two
- Three

10. Наследует ли класс конструкторы свое суперкласса?

- Да
- Нет

11. Каким образом можно выделить большее максимальное количество памяти для Java Virtual Machine?

- Необходимо вызвать метод `Runtime.getRuntime().maxMemory();` и JAVA автоматически увеличит максимальное количество памяти
- Необходимо запустить JVM с ключом `-Xmx` и соответствующим значением количества памяти. Например `-Xmx256m`.
- JVM нельзя выставить максимальное количество памяти. Оно выставляется автоматически, в зависимости от количества оперативной памяти.
- Необходимо запустить JVM с ключом `-Xms` и соответствующим значением количества памяти. Например `-Xms256m`.

12. Каким будет результат компиляции и выполнения данного кода?

```
class A1{ }  
interface I1{ }
```

```

class B1 extends A1 implements I1 {}

public class OverloadTest{
    static public void foo(A1 a){System.out.print("A");}
    static public void foo(B1 a){System.out.print("B");}
    static public void foo(I1 a){System.out.print("I");}
    public static void main(String[] args){
        A1 a=new B1();
        OverloadTest.foo(a);
        OverloadTest.foo((I1)a);
    }
}

```

- AA
- AB
- AI
- BA
- BB
- BI
- Ошибка компиляции

13. Что произойдёт после выполнения данного кода?

```

public class Q13 {
    public static void main(String[] args){
        List list=new ArrayList(3);//1
        list.add(new Integer(100));
        list.add(new Integer(200));
        list.add(new Integer(300));
        list.add(new Integer(400));//2
        System.out.println(list.size());//3
    }
}

```

- В строке 2 будет выброшено `IndexOutOfBoundsException()`
- В строке 1 произойдёт ошибка компиляции, т.к. у класса `ArrayList` нет соответствующего конструктора
- Компиляция произойдёт успешно и будет напечатано 4
- В строке 3 будет выброшено `IndexOutOfBoundsException()`

14. Что будет напечатано следующим кодом?

```
public class Q14 {
    public static void var(Integer x,int y){
        System.out.println("Integer int");}
    public static void var(Object... x){
        System.out.println("Object");}
    public static void var(int... x){
        System.out.println("int...x");}
    public static void var(Integer... x){
        System.out.println("Integer...");}
    public static void main(String... args){
        int i=0;
        Integer i2=127;
        var(i,i2);
    }
}
```

- `Integer int`
- `Object`
- `int...x`
- `Integer...`

15. Определена следующая переменная строкового типа

```
String s="abc";
```

Какие из перечисленных ниже операций допустимы?

- `int j=s[0];`
- `s+=s+s;`
- `s=2*s;`
- `s=(String)(46>>2);`

16. Что будет напечатано в результате выполнения следующего кода?

```
public class Cycle {
    public static void main(String... args){
        int i=1;
        do while(i<1)
            System.out.print("i is "+ ++i);
        while(i>1);
    }
}
```

- Ошибка компиляции
- `i=1`
- `i=2`
- Программа зациклится
- Программа выполнится, но ничего не выдаст на экран

17. Что будет напечатано в результате выполнения данного кода?

```
public class MathTest {
    public static void main(String... args){
        System.out.print((-7%5)>(7%-5));
        System.out.print(";");
        System.out.print(Math.abs(Math.ceil(-2.5))<Math.ceil(Math.abs(-
2.5)));
    }
}
```

- `true;false`

- false;false
- true>true
- false>true

18. Каким будет результат выполнения следующей программы?

```
import java.util.*;
public class TestFormater {
    public static void main(String... args){
        Integer I1=0;
        Integer I2=-1;
        Integer I3=1;
        Formatter f=new Formatter();
        f.format("%1$b",I1.toString()).format("%1$b",I2.toString()).format("%1$b",I3.toString());
        System.out.println(f.toString());
    }
}
```

- true true true
- 0 -1 1
- true false true
- false false true

19. Что будет выведено на печать после выполнения данного кода?

```
import java.util.*;
public class MapTest {
    public static void main(String... args){
        Map<String,Integer> map1=new HashMap<String,Integer>();
        Map<String,Integer> map2=new HashMap<String,Integer>();
        map1.put("Number1",new Integer(100));
        map1.put("Number2",new Integer(200));
    }
}
```

```

map1.put("Number3",new Integer(300));
List<Map> list=new ArrayList<Map>();
list.add(map1);
list.add(map2);
HashMap resultMap=(HashMap)list.get(0);
System.out.println("Number: "+resultMap.get("Number2"));
}}

```

Будет напечатано: Number: 300

Результат невозможно предугадать

Будет напечатано: Number: 100

Произойдёт ошибка компиляции, так как в интерфейсе Map нет методу put()

Будет напечатано: null

Будет напечатано: Number: 200

20. Что произойдёт во время компиляции и выполнения данного кода?

```

public class Q20 {
    public static void main(String[] args){
        int i=0,j=5;
        tp:for(;;){
            i++;
            for(;;){
                if(i>--j){break tp;}}
            System.out.println("i="+i+", j="+j);
        }
    }
}

```

- Будет выведено: “i=1, j=0”
- Будет выведено: “i=1, j=4”
- Будет выведено: “i=3, j=4”

- Будет выведено: “i=3, j=0”
- Ошибка компиляции

## **6.2. Экзаменационные вопросы**

1. Создание простейшей программы на Java, её компиляция в байт-код и запуск, использование командной строки.
2. Массивы в Java. Использование классов обёрток в Java.
3. Использование коллекций в Java.
4. Описание классов, интерфейсов, наследование классов и осуществление интерфейсов.
5. Обработка исключений.
6. Механизм использования событий и уведомлений: стандартный паттерн.
7. Механизм использования событий и уведомлений: явное описание.
8. Стандартные классы обработки событий, назначение, функции подписки приёмников.
9. Поток. Способы создания потоков. Синхронизация потоков. Конфликты при выполнении потоков.
10. Классы пакета java.net для работы с сетью. TCP протокол.
11. Классы пакета java.net для работы с сетью. UDP протокол.
12. Технология RMI. Интерфейс Remote. Реализация удаленного интерфейса.
13. Технология RMI. Передача объектов в RMI.
14. Графика в Java. Пакет java.awt, его основные классы. События окна.
15. Графика в Java. Пакет java.awt, его основные классы. Дочерние окна.
16. Графика в Java. Пакет java.awt, его основные классы. Компоновка элементов.
17. Пакеты Java для работы с сервлетами. Жизненный цикл сервлета. Контейнеры сервлетов.

18. Пакеты Java для работы с сервлетами. Особенности разработки сервлетов.
19. Пакеты Java для работы с сервлетами. Правила использования сервлетов.
20. Технология Java Beans. Самодиагностика компонента.
21. Технология Java Beans. Контроль состояния компонента.
22. Технология Java Beans. Сохранение состояния компонента.
23. Технология JavaServer Pages.
24. Архитектура web-приложения. Странице - центричная архитектура(Model1). Шаблон проектирования (Model-View-Controller, Model1).
25. Основы синтаксиса JSP.
26. Предопределённые jsp-переменные.
27. Директивы JSP-сервлета.
28. JSP и компоненты Java beans.
29. Язык выражений в стандарте JSP.
30. Связь JSP и JSF. Компонентно-ориентированную модель (Model2).
31. JDBC. Класс java.sql.Connection.
32. JDBC. Класс Statement.
33. JDBC. Класс ResultSet.
34. JDBC. Класс PreparedStatement: создание объектов; Передача входных (IN) параметров.
35. JDBC. Класс PreparedStatement: совместимость типов данных входных параметров; использование объектов.
36. JDBC. Класс PreparedStatement: отсылка значений NULL в качестве входного параметра; отправка очень больших входных параметров.
37. JDBC. Основные команды SQL. Использование транзакций при работе с БД в Java (краткий обзор).
38. JDBC. Транзакции: отключение режима автоматической фиксации; выполнение; утверждение.

- 39.JDBC. Транзакции: использование транзакций для сохранения целостности данных.
- 40.JDBC. Транзакции: настройка и возврат к точкам сохранения; освобождение точек сохранения; использование метода обратного вызова (откат, rollback).

## 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