

**Приложение 2 к РПД Технологии программирования  
09.03.02 Информационные системы и технологии  
Направленность (профиль) – Программно-аппаратные комплексы  
Форма обучения – заочная  
Год набора - 2019**

**ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ  
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

**1. Общие сведения**

1.	Кафедра	Информатики и вычислительной техники
2.	Направление подготовки	09.03.02 Информационные системы и технологии
3.	Направленность (профиль)	Программно-аппаратные комплексы
4.	Дисциплина (модуль)	Технологии программирования
5.	Форма обучения	заочная
6.	Год набора	2019

**2. Перечень компетенций**

– способен разрабатывать, документировать, тестировать и адаптировать прикладное программное обеспечение информационных систем, принимать участие в управлении работами по разработке информационных систем (ПК-1)
--

### 3. Критерии и показатели оценивания компетенций на различных этапах их формирования

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
1. Введение в технологию программирования	ПК-1	этапы развития технологий программирования	разрабатывать блок-схемы алгоритмов	терминологией объектно-ориентированной разработки алгоритмов	Тест
2. Введение в анализ алгоритмов	ПК-1	понятие анализ алгоритма; примеры расчета анализа сложности алгоритмов	выбирать наилучший алгоритм с точки зрения его производительности	методикой оценки анализа сложности алгоритмов	
3. Рекурсивные алгоритмы и динамическое программирование	ПК-1	понятие рекурсии и рекурсивного алгоритма; примеры рекурсивных алгоритмов; ключевую идею динамического программирования; отличия в реализации рекурсивных и динамических алгоритмов	отличать рекурсивный алгоритм от итерационного; выполнять построение дерева рекурсивных вызовов; выполнять разбиение исходной задачи на последовательность подзадач;	методикой расчета сложности рекурсивного алгоритма; методикой разработки алгоритмов динамического программирования;	
4. Алгоритмы сортировки	ПК-1	основные алгоритмы внутренней и внешней сортировки	выбирать алгоритм сортировки в соответствии с поставленной задачей; выполнять анализ алгоритмов сортировки	навыками разработки алгоритмов сортировки в средах программирования	Тест, лабораторные работы (2), групповые дискуссии (2)
5. Динамические структуры данных	ПК-1	основные принципы организации динамических структур данных	разрабатывать собственные алгоритмы работы с динамическими структурами данных в средах программирования	навыками работы с библиотечными динамическими структурами данных в средах программирования	Тест
6. Алгоритмы поиска	ПК-1	основные алгоритмы поиска в различных структурах данных	выбирать алгоритм поиска в соответствии с поставленной задачей; выполнять анализ алгоритмов поиска	навыками разработки алгоритмов поиска в средах программирования	
7. Алгоритмы на деревьях	ПК-1	основные алгоритмы на деревьях	выбирать алгоритм работы на деревьях в соответствии с поставленной задачей; выполнять анализ алгоритмов	навыками разработки алгоритмов на деревьях в средах программирования	Тест, лабораторные работы (2), групповые дискуссии (2)

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
8. Алгоритмы на графах	ПК-1	основные алгоритмы на графах	выбирать алгоритм работы на графах в соответствии с поставленной задачей; выполнять анализ алгоритмов	навыками разработки алгоритмов на графах в средах программирования	

## 4. Критерии и шкалы оценивания

### 4.1. Тест

Процент правильных ответов	до 50	51-60	61-80	81-100
Количество баллов за ответы	0	1	2	3

### 4.2. Выступление с докладом

Баллы	Характеристики ответа обучающегося
10	<ul style="list-style-type: none"><li>- обучающийся глубоко и всесторонне усвоил проблему;</li><li>- уверенно, логично, последовательно и грамотно его излагает;</li><li>- опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью;</li><li>- умело обосновывает и аргументирует выдвигаемые им идеи;</li><li>- делает выводы и обобщения;</li><li>- свободно владеет понятиями</li></ul>
5	<ul style="list-style-type: none"><li>- обучающийся твердо усвоил тему, грамотно и по существу излагает ее, опираясь на знания основной литературы;</li><li>- не допускает существенных неточностей;</li><li>- увязывает усвоенные знания с практической деятельностью;</li><li>- аргументирует научные положения;</li><li>- делает выводы и обобщения;</li><li>- владеет системой основных понятий</li></ul>
2	<ul style="list-style-type: none"><li>- тема раскрыта недостаточно четко и полно, то есть обучающийся освоил проблему, по существу излагает ее, опираясь на знания только основной литературы;</li><li>- допускает несущественные ошибки и неточности;</li><li>- испытывает затруднения в практическом применении знаний;</li><li>- слабо аргументирует научные положения;</li><li>- затрудняется в формулировании выводов и обобщений;</li><li>- частично владеет системой понятий</li></ul>
0	<ul style="list-style-type: none"><li>- обучающийся не усвоил значительной части проблемы;</li><li>- допускает существенные ошибки и неточности при рассмотрении ее;</li><li>- испытывает трудности в практическом применении знаний;</li><li>- не может аргументировать научные положения;</li><li>- не формулирует выводов и обобщений;</li><li>- не владеет понятийным аппаратом</li></ul>

### 4.3. Выполнение лабораторной работы

**10 баллов** выставляется, обучающийся выполнил полностью все задания указанные в лабораторной работе и может аргументировано пояснить ход своего решения.

**5 баллов** выставляется, если обучающийся выполнил не менее 85 % заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения и указать.

**2 балла** выставляется, если обучающийся решил не менее 50% заданий указанных в лабораторной работе, и может аргументировано пояснить ход своего решения.

**0 баллов** выставляется, если обучающийся не может аргументированно пояснить ход своего решения.

В случае если сроки сдачи работ превышены, количество баллов сокращается на 50%.

### 4.4. Групповая дискуссия (устные обсуждения проблемы или ситуации)

<b>Критерии оценивания</b>	<b>Баллы</b>
<ul style="list-style-type: none"> <li>– обучающийся ориентируется в проблеме обсуждения, грамотно высказывает и обосновывает свои суждения, владеет профессиональной терминологией, осознанно применяет теоретические знания, материал излагает логично, грамотно, без ошибок;</li> <li>– при ответе обучающийся демонстрирует связь теории с практикой.</li> </ul>	<b>2</b>
<ul style="list-style-type: none"> <li>– обучающийся грамотно излагает материал; ориентируется в проблеме обсуждения, владеет профессиональной терминологией, осознанно применяет теоретические знания, но содержание и форма ответа имеют отдельные неточности;</li> <li>– ответ правильный, полный, с незначительными неточностями или недостаточно полный.</li> </ul>	<b>1</b>
<ul style="list-style-type: none"> <li>– обучающийся излагает материал неполно, непоследовательно, допускает неточности в определении понятий, не может доказательно обосновать свои суждения;</li> <li>– обнаруживается недостаточно глубокое понимание изученного материала.</li> </ul>	<b>0</b>

#### **4.5. Выполнение задания на составление глоссария**

	<b>Критерии оценки</b>	<b>Количество баллов</b>
1	аккуратность и грамотность изложения, работа соответствует по оформлению всем требованиям	<b>2</b>
2	полнота исследования темы, содержание глоссария соответствует заданной теме	<b>3</b>
	<b>ИТОГО:</b>	<b>5 баллов</b>

**5. Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

##### **5.1. Типовое тестовое задание**

1. Каков порядок сложности алгоритма бинарного (двоичного) поиска элемента в массиве?

1.  $O(n)$
2.  $O(n^2)$
3.  $O(\log(n))$
4.  $O(2^n)$

2. Каков порядок сложности алгоритма линейного поиска элемента в массиве?

1.  $O(n)$
2.  $O(n^2)$
3.  $O(\log(n))$
4.  $O(2^n)$

3. Каков порядок сложности «наивного» алгоритма поиска подстроки (длиной  $m$ ) в строке (длиной  $n$ ):

1.  $O(n*m)$
2.  $O(n-m)$
3.  $O(n+m)$
4.  $O(n/m)$

4. Для любой строки  $S$ , через  $S[1..i]$ , обозначается?

1. префикс строки  $S$ , заканчивающийся в позиции  $j$ ;
2. суффикс строки  $S$ , начинающийся в позиции  $i$ ;
3. подстрока строки  $S$ , начинающаяся в позиции  $i$  и заканчивающаяся в позиции  $j$ ;
4. длина строки  $S$ .

5. Суффикс строки  $S$  называется собственным если?

1. он не пустой и совпадает с префиксом;
2. он не пустой и не совпадает с  $S$ ;
3. он не пустой и совпадает с  $S$ ;
4. он не пустой.

6. Рассматриваем алгоритм вычисления префикс – функции для символов строки  $S$ . Пусть мы построили решение для первых  $i$  символов. Чему равно  $\Pi_{i+1}(S)$ , если  $i+1$  символ строки  $S$  совпадает с символом, стоящим за префиксом  $\Pi_i(S)$ ?

1.  $\Pi_{i+1}(S) = \Pi_i(S)$ ;
2.  $\Pi_{i+1}(S) = \Pi_i(S) + 1$ ;
3.  $\Pi_{i+1}(S) = \Pi_i(S) - 1$ .

7. В каком из перечисленных методов внутренней сортировки выгодно применяется метод установки «барьера»?

1. Сортировка выбором.
2. Сортировка «пузырьком».
3. Сортировка вставками.
4. Сортировка слиянием.

8. В каком из перечисленных методов внутренней сортировки используется тот факт, что при обработке некоторого элемента  $a[i]$  элементы  $a[1], a[2], \dots, a[i-1]$ , уже могут быть упорядоченными?

1. Сортировка Шелла.
2. Сортировка «пузырьком».
3. Сортировка вставками.
4. Бинарная сортировка.

9. Многопроходная внутренняя сортировка, при которой исходный массив разбивается на части, каждая из которых сортируется отдельно, причем на каждом проходе число частей уменьшается, называется?

1. Сортировкой Шелла.
2. Сортировкой «пузырьком».
3. Сортировкой вставками.
4. Бинарной сортировкой.

10. Представлен фрагмент псевдокода для некоторого алгоритма внутренней сортировки. Какой оператор закрыт рамкой?

```

for i := 2 to n do
  x := a[i];
  a[0] := x;
  j := i - 1;
  while (x < 0 a[j]) do
    a[j+1] := a[j];
    j := j - 1;

```



1.  $a[j + 1] := x;$
2.  $a[j - 1] := x;$
3.  $a[j] := x;$
4. нет оператора.

**Ключ:** 1-3, 2-1, 3-1, 4-1, 5-2, 6-2, 7-3, 8-4, 9-1, 10 - 1

### 5.2. Примерные темы докладов

1. Процессы жизненного цикла программного продукта по стандарту ISO/IEC 12207.
2. Проектирование интерфейса пользователя. Модели и технологии.
3. Организация процесса тестирования программного обеспечения.
4. Функциональное тестирование программного обеспечения или тестирование «черного ящика».
5. Структурное тестирование программного обеспечения или тестирование «белого ящика».
6. Модели надежности программного обеспечения.
7. Обзор современных сред разработки программного обеспечения.
8. Общие принципы разработки распределенных систем.
9. Введение в унифицированный процесс разработки программных приложений (RUP).
10. Применение case-средств проектирования в процессе разработки программных приложений.
11. Общие принципы компонентно-ориентированной разработки.
12. Аспектно и атрибутно – ориентированная разработка программных приложений.
13. Платформа .NET.
14. Общие принципы и технологии разработки приложений для систем реального времени.
15. Общие принципы разработки приложений для встраиваемых систем.
16. Общие принципы разработки корпоративных приложений.
17. Использование паттернов проектирования в процессе разработки программных приложений.
18. Процессы жизненного цикла программного продукта.

### 5.3. Пример задания на лабораторную работу

*Тема:* Разработать программу вычислительного эксперимента для исследования характеристик эффективности сортировки одним из методов (согласно варианту). Проверить совпадение модельных представлений и результатов вычислительного эксперимента. Предусмотреть вывод исходной и результирующей информации в файл.

*Требования:*

Разработать программу в среде Visual Studio или Borland Developer Studio, приложение типа Windows Application, на языке программирования или C# или C++.

В программе реализовать:

- реализовать в программе один из методов сортировки массивов согласно варианту задания в виде отдельной функции;
- использовать динамические массивы;
- вычислительный эксперимент выполнить для нескольких массивов, число которых устанавливает пользователь;
- число элементов в массиве устанавливает пользователь, массив создается с помощью генератора случайных чисел;
- для каждого эксперимента привести теоретическое и практическое (полученное в результате эксперимента) число сравнений и перестановок;
- результаты эксперимента записать в файл;
- обработать исключительные ситуации;
- предусмотреть диалог с пользователем;
- отобразить графически на диаграмме полученные результаты эксперимента для каждого массива.

#### **5.4. Вопросы к экзамену**

1. Стихийное программирование.
2. Структурное программирование. Модульное программирование.
3. Объектно-ориентированное программирование.
4. Компонентно-ориентированное программирование.
5. Анализ алгоритмов. Скорости роста, классификация скоростей роста. Пример анализа сложности алгоритмов.
6. Рекурсивные алгоритмы. Определение рекурсии. Схемы прямой (простой) и косвенной (сложной) рекурсии. Примеры рекурсивных алгоритмов. Пример: «Ханойская башня».
7. Реализация механизма рекурсивного вызова процедуры.
8. Постановка задачи. Идея динамического программирования. Сравнение рекурсии и динамического программирования.
9. Задачи, решаемые методом динамического программирования: задача о полоске.
10. Задачи, решаемые методом динамического программирования: задача о короле.
11. Задачи, решаемые методом динамического программирования: задача о разложении числа.
12. Задачи, решаемые методом динамического программирования: задача о ранце.
13. Алгоритм сортировки вставками. Анализ сложности алгоритма.
14. Бинарная сортировка. Анализ сложности алгоритма.
15. Сортировка выбором. Анализ сложности алгоритма.
16. Обменная сортировка. Анализ сложности алгоритма.
17. Шейкерная сортировка. Анализ сложности алгоритма.
18. Сортировка Шелла. Анализ сложности алгоритма.
19. Сортировка элементов массива методом слияния. Анализ сложности алгоритма.
20. Алгоритм быстрой сортировки. Анализ сложности алгоритма.
21. Пирамидальная сортировка. Анализ сложности алгоритма.
22. Внешние сортировки. Классификация методов внешней сортировки. Простое слияние.
23. Внешние сортировки. Естественное слияние.
24. Внешние сортировки. Получение начальных серий для многопутевого слияния посредством выбора с замещением.
25. Многофазная сортировка.
26. Каскадная сортировка.
27. Совпадение строк. Постановка задачи. Наивный алгоритм поиска. Оценка сложности. Ускорение наивного алгоритма. Оценка сложности.

28. Префикс – функция. Алгоритм поиска префикс – функции. Оценка сложности.
29. Алгоритм Кнута - Морриса – Пратта (КМП). Оценка сложности.
30. Хеширование. Постановка задачи. Хеш-функции. Методы разрешения коллизий.
31. Алгоритмы вставки поиска и удаления для хеш-таблицы, имеющей три состояния элементов. Переполнение таблицы и рехеширование. Оценка качества хеш-функций.
32. Линейные структуры данных. Стек, очередь, дек.
33. Линейные списки, циклические списки. Основные операции со списками.
34. Деревья, бинарные деревья. Представление бинарных деревьев. Прохождение бинарных деревьев.
35. Сортировка с прохождением бинарного дерева в симметричном порядке. Сортировка методом турнира с выбыванием.
36. Применение бинарных деревьев для сжатия информации.
37. Представление сильноветвящихся деревьев. Множественная задача точного поиска образцов в тексте: наивный алгоритм, обобщенный КМП.
38. Представление графов в памяти компьютера.
39. Поиск в глубину. Оценка сложности алгоритма.
40. Топологическая сортировка. Оценка сложности алгоритма.
41. Определение наличия циклов в графе. Оценка сложности алгоритма.
42. Обход в ширину. Оценка сложности алгоритма.
43. Раскраска графа.
44. Алгоритм Прима.
45. Алгоритм Краскала.
46. Алгоритм Дейкстры.

**ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ**  
**ОСНОВНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА**  
**09.03.02 Информационные системы и технологии**  
**Направленность (профиль) «Программно-аппаратные комплексы»**

(код, направление, профиль)

**ТЕХНОЛОГИЧЕСКАЯ КАРТА**

Шифр дисциплины по РУП	<b>Б1.В.01</b>		
Дисциплина	<b>Технологии программирования</b>		
Курс	<b>3</b>	семестр	<b>5</b>
Кафедра	<b>Информатики и вычислительной техники</b>		
Ф.И.О. преподавателя, звание, должность	<b>Тоичкин Николай Александрович, канд. техн. наук, доцент кафедры информатики и вычислительной техники</b>		
Общ. трудоемкость <sup>час/ЗЕТ</sup>	<b>180/5</b>	Кол-во семестров	<b>2</b>
Форма контроля	<b>Экзамен</b>		
ЛК общ./тек. сем.	<b>4/4</b>	ПР/СМ <sup>общ./тек. сем.</sup>	<b>-/-</b>
ЛБ <sup>общ./тек. сем.</sup>	<b>8/8</b>	СРС общ./тек. сем.	<b>159/159</b>

**Компетенции обучающегося, формируемые в результате освоения дисциплины:**

– способен разрабатывать, документировать, тестировать и адаптировать прикладное программное обеспечение информационных систем, принимать участие в управлении работами по разработке информационных систем (ПК-1)

Код формируемой компетенции	Содержание задания	Количество мероприятий	Максимальное количество баллов	Срок предоставления
<i><b>Вводный блок</b></i>				
Не предусмотрен				
<i><b>Основной блок</b></i>				
ПК-1	Решение бланочных тестов	4	12	В течение семестра
ПК-1	Лабораторные работы	4	40	В течение семестра по расписанию занятий
ПК-1	Групповые дискуссии	4	8	В течение семестра по расписанию занятий
ПК-1	<b>Итого:</b>		<b>60</b>	
ПК-1	Экзамен	Вопрос 1 Вопрос 2	20 20	В сроки сессии
	<b>Всего:</b>		<b>40</b>	
	<b>Итого:</b>		<b>100</b>	
<i><b>Дополнительный блок</b></i>				
ПК-1	Подготовка докладов по теме		10	по согласованию с преподавателем
ПК-1	Выполнение задания на составление глоссария		5	
	<b>Всего:</b>		<b>15</b>	

Шкала оценивания в рамках балльно-рейтинговой системы МАГУ: «2» - 60 баллов и менее, «3» - 61-80 баллов, «4» - 81-90 баллов, «5» - 91-100 баллов.