

**Приложение 2 к РПД Программирование  
09.03.02 Информационные системы и технологии  
Направленность (профиль) – Программно-аппаратные комплексы  
Форма обучения – заочная  
Год набора - 2017**

**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)	Программирование
5.	Форма обучения	заочная
6.	Год набора	2017

**2. Перечень компетенций**

<ul style="list-style-type: none"><li>- способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий (ОПК-6);</li><li>- способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем (ОПК-7).</li></ul>
--

### 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. Языки программирования	ОПК-6; ОПК-7	Алфавит, синтаксис, семантика языка программирования. Методы описания синтаксических конструкций языков программирования. Эволюция языков программирования. Классификация языков программирования. Парадигмы программирования.	ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования	методами и инструментальными средствами разработки программ	решение задач (в виде программы)
2. Типы языка C++	ОПК-6; ОПК-7	Понятие типа данных. Классификация типов данных. Базовые типы. Структурные типы. С-Строки. Ссылки и указатели. Процедурные типы. Эквивалентность и совместимость типов. <i>Типы из стандартной библиотеки (string, AnsiString, ...)</i>	решать задачи по основным разделам курса	методами построения математической модели	решение задач (в виде программы)
3. Выражения языка C++	ОПК-6; ОПК-7	Операнды и операции. Приоритеты операций. Арифметические операции. Булевские операции. Логические операции. Операции над строками. Операции над множествами. Операции отношения.	решать задачи по основным разделам курса	навыками разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня	решение задач (в виде программы)

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
4. Инструкции (операторы) языка C++	ОПК-6; ОПК-7	Классификация операторов. Простые операторы. Особенность оператора GOTO. Структурные операторы.	решать задачи по основным разделам курса	навыками разработки и отладки программ	решение задач (в виде программы)
5. Объектно-ориентированное программирование	ОПК-6; ОПК-7	Основные понятия и принципы ООП. Классы. Инкапсуляция: скрытые поля и методы. Наследование: перекрытие методов. Полиморфизм: виртуальные методы. Конструкторы и деструкторы. Динамические объекты.	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного программирования	решение задач (в виде программы)
6. Поточковые классы:	ОПК-6; ОПК-7	понятие потока и операций работы с ним. Стандартные потоки. Форматирование данных. методы обмена с потоками. Ошибки потоков. Файловые потоки. виды потоковых устройств. Потоки и типы, определенные пользователем.	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного программирования	решение задач (в виде программы)

## 4. Критерии и шкалы оценивания

### 4.1. Решение задач (в виде программы)

**15 баллов** выставляется, если обучающийся выполнил все лабораторные работы, правильно изложил все варианты решения, аргументировав их, с обязательной ссылкой на соответствующие нормативы (если по содержанию это необходимо) в поставленные сроки.

**10 баллов** выставляется, если обучающийся решил не менее 85% рекомендованных задач, правильно изложил все варианты решения, аргументировав их, с обязательной ссылкой на соответствующие нормативы (если по содержанию это необходимо).

**5 баллов** выставляется, если обучающийся решил не менее 65% рекомендованных задач, правильно изложил все варианты их решения, аргументировав их, с обязательной ссылкой на соответствующие нормативы (если по содержанию это необходимо).

**0 баллов** - если обучающийся выполнил менее 50% задания, и/или неверно указал варианты решения.

**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. Пример задачи

Написать программу обработки одномерного и двумерного массива целочисленных значений. Целевая функция для обработки - минимальное значение.

**Решение:**

```
#include <dos.h>
#include <stdio.h>
#include<iostream.h>

int A[5] = {1,2,3,4,5};
//int M[5][5] = {{1,2,3,4,5},{1,2,3,4,5},{1,2,3,4,5}};

int count (int* K, int N1)
{
    int Min=K[0];
    for(int i=0;i<N1;i++) if(Min>K[i]) Min=K[i];

return Min;
};

void count2 (int* K, int N1, int &Min)
{
Min=K[0];
    for(int i=0;i<N1;i++) if(Min>K[i]) Min=K[i];
};

int count3 (int** K, int N1, int N2)
{
    int Min=K[0][0];
    for(int i=0;i<N1;i++)
    for(int j=0;j<N2;j++)
    if(Min>K[i][j]) Min=K[i][j];

    return Min;
};

void main()
{
    cout<<count(A, 5);
    int rez;
```

```

count2(A, 5, rez);
cout<<rez;

int **M = new int*[5];
for(int i=0;i<5;i++){
M[i] = new int[5]; //{{1,2,3,4,5},{1,2,3,4,5},{1,2,3,4,5}};
for(int j=0;j<5;j++) M[i][j] = i+j+1;
}
cout<<count3(M,5,5);
}

```

## 5.2. Вопросы к промежуточной аттестации

### 2 курс

1. Языки программирования: назначение, особенности, классификация. Цели и задачи курса.
2. Состав алгоритмического языка (на примере C++): алфавит, лексемы.
3. Концепция типа данных. Основные (базовые) типы данных языка C++.
4. Структура программы на языке C++. Функция main(). Этапы создания исполняемой программы.
5. Переменные и константы в языке C++. Область действия, видимости идентификатора и время жизни переменной. Размещение переменных в памяти.
6. Операции языка C++: классификация и примеры. Построение выражений. Порядок выполнения операций. Изменение порядка выполнения операций.
7. Структурное программирование. Базовые конструкции структурного программирования
8. Оператор "выражение". Операции присваивания в языке C++.
9. Условный оператор. Порядок выполнения, особенности, синтаксис в C++.
10. Оператор множественного выбора: порядок выполнения, особенности, синтаксис в C++.
11. Циклы. Оператор цикла с предусловием: порядок выполнения, особенности, синтаксис в C++.
12. Циклы. Оператор цикла с постусловием: порядок выполнения, особенности, синтаксис в C++
13. Циклы. Оператор цикла с параметром: порядок выполнения, особенности, синтаксис в C++.
14. Указатели. Инициализация указателей. Операции с указателями в C++: разыменование, взятие адреса, арифметические и логические.
15. Ссылки. Инициализация ссылок. Операции со ссылками в C++.
16. Массивы в C++: одномерные и многомерные, статические и динамические. Действия над массивами: доступ к элементам, инициализация, и др.
17. С-строки. Действия над С-строками. Функции библиотеки для обработки С-строк в C++
18. Процедуры выделения и освобождения динамической памяти. Ошибочные ситуации при использовании динамической памяти.
19. Типы, определяемые пользователем. Переименование типов (typedef) и перечисления (enum).
20. Типы, определяемые пользователем. Структуры (struct): описание, инициализация, доступ к полям.
21. Модульное программирование. Функции. Объявление, определение и вызов функций в C++. Возвращение результата из функции.
22. Функции. Передача параметров в функцию: по адресу, по значению. Передача массивов в качестве параметров. Перегрузка функции.
23. Стандартная библиотека. Классификация и примеры функций стандартной библиотеки: математические, строки, ввода/вывода.

24. Потоки. Классификация потоков. Стандартные потоки. Операции над потоками: ввод из потока, вывод в поток. Методы обмена с потоками.
25. Файл - как особый способ хранения данных. Работа с файлами в C++. Файловые потоки. Последовательность работы с файлами. Операции чтения, записи и поиска в файле.
26. Этапы создания исполняемой программы. Препроцессор. Директивы препроцессора.
27. Поименованные области (namespace). Проект. Раздельная компиляция и компоновка: интерфейс, реализация, использование собственного модуля.
28. Технология создания программ. Рекомендации по кодированию и документированию программы: цели, переменные, операторы, функции, комментарии и др.
29. Технология создания программ. Проектирование и тестирование программ. Структурный подход. Этапы развития проекта и их назначение.
30. Динамические структуры данных. Линейный список, стек, очередь. Типовые операции обработки динамических структур данных.

### 3 курс

1. Структура, цель и задачи учебного курса.
2. Перегрузка операторов. Интерпретация операций в C++.
3. Объектно-ориентированное видение мира. Проблемы использования ООП в профессиональной сфере (Бизнес-информатика, Информатика и вычислительная техника, Информационные системы и технологии).
4. Типы отношений между классами. Иерархия классов.
5. Методы повышения абстракции. Объектно-ориентированное программирование.
6. Типы отношений между объектами. Иерархия объектов.
7. Типы, определяемые пользователем. Базовые механизмы ООП.
8. Производные классы. Множественное наследование.
9. Понятие класса в ООП. Описание класса.
10. Виртуальные методы. Механизм раннего и позднего связывания.
11. Понятие объекта в ООП. Описание объектов.
12. Простое наследование. Правила наследования.
13. Управление доступом к данным и функциям-членам. Константные члены.
14. Производные классы. Наследование. Ключи доступа.
15. Функции-члены. Указатель this.
16. Конкретные типы. Проблемы, связанные с конкретными типами.
17. Понятие конструктора. Свойства конструктора.
18. Рекомендации по составу класса. Подготовка исходного кода к компиляции.
19. Массивы объектов. Параметры по умолчанию.
20. Потокосовые классы: понятие потока и операций работы с ним. Стандартные потоки. Форматирование данных.
21. Понятие конструктора. Типы конструкторов.
22. Потокосовые классы: методы обмена с потоками. Ошибки потоков. Файловые потоки.
23. Статические элементы класса.
24. Потокосовые классы: виды потокосовых устройств. Потоки и типы, определенные пользователем.
25. Дружественные функции и классы.
26. Обработка исключительных ситуаций: понятие и использование исключительной ситуации.
27. Деструктор. Свойства деструктора.

28. Обработка исключительных ситуаций: общий механизм и синтаксис обработки исключений.
29. Перегрузка операторов. Правила перегрузки операторов.
30. Шаблоны: параметризация алгоритмов и типов данных. Применение шаблонов к типам, определяемым пользователем.

### 5.3. Примеры практических экзаменационных заданий:

<b>Массивы элементов:            заполнение, поиск, замена, вывод</b>	
1.1.	Написать функцию поиска минимального элемента в одномерном массиве. Массив и количество элементов передать в функцию в виде параметров. Результат вернуть как результат функции.
1.2.	Написать функцию поиска суммы положительных элементов одномерного массива, расположенных на четных местах. Все необходимые для вычисления данные передать в функцию через параметры. Результат вычисления вернуть как результат функции.
<b>Динамические структуры – указатели: списки, динамические массивы, операции над указателями</b>	
1.3.	Определить пользовательский тип, включающий в себя следующие поля: строка, целое число, указатель на байт, указатель на число с плавающей точкой. Внести изменения в созданный тип, так чтобы объекты этого типа могли образовывать односвязный список. Создать односвязный список из n объектов созданного типа.
1.4.	Определить пользовательский тип, включающий в себя следующие поля: целое число и указатель на создаваемый пользовательский тип. Предполагая, что имеется уже созданный односвязный список элементов описанного типа, определить функцию вставки нового элемента после заданного (указателем) элемента в этом списке.
<b>Файлы                            ввод/вывод</b>	
1.5.	Дан файл с целыми числами. Необходимо определить функцию, которая в качестве параметров принимает название файла и ссылку на переменную беззнакового целого типа. Функция должна читать заданный файл и подсчитывать сумму элементов в файле. Результат возвращается через ссылку.
1.6.	Дан файл с произвольным текстом. Необходимо определить функцию, которая в качестве параметров принимает название файла и указатель на переменную целого беззнакового типа. Функция должна подсчитать количество символов пробел в указанном файле и вернуть значение через параметр (через указатель).

### Вычисление арифметических выражений:

$$\frac{(a+b)^2 - (a^2 + 2ab)}{b^2}$$
 Вычислить значение заданного арифметического выражения. Исходные данные вводятся из файла. Вычисление оформить в виде функции, в которую передается имя файла с исходными данными. Функция возвращает результат вычислений. Результат записывается в файл **result.txt**. (контрольный тест: a=1000, b=0.0001)

$$\frac{\sin(a/b)^3 - \ln(a+ab^2)}{\sqrt{a-b}}$$
 Вычислить значение заданного арифметического выражения. Исходные данные вводятся из файла. Вычисление оформить в виде функции, в которую передается имя файла с исходными данными. Функция возвращает результат вычислений (через параметр по ссылке). Результат записывается в файл **result.txt**. (контрольный тест: a=1000, b=0.0001)

**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.О.31</b>				
Дисциплина	<b>Программирование</b>				
Курс	<b>2</b>	семестр	<b>3-4</b>		
Кафедра	<b>Информатики и вычислительной техники</b>				
Ф.И.О. преподавателя, звание, должность	<b>Федоров Андрей Михайлович, канд. техн. наук, доцент кафедры информатики и вычислительной техники</b>				
Общ. трудоемкость <sup>час/ЗЕТ</sup>	<b>216/6</b>	Кол-во семестров	<b>2</b>	Форма контроля	<b>Экзамен</b>
ЛК <sub>общ./тек. сем.</sub>	<b>8/4</b>	ПР/СМ <sub>общ./тек. сем.</sub>	<b>-/-</b>	ЛБ <sub>общ./тек. сем.</sub>	<b>12/6</b>
				СРС <sub>общ./тек. сем.</sub>	<b>286/187</b>

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

<ul style="list-style-type: none"> <li>- способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий (ОПК-6);</li> <li>- способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем (ОПК-7).</li> </ul>
---

Код формируемой компетенции	Содержание задания	Количество мероприятий	Максимальное количество баллов	Срок предоставления
<i>Вводный блок</i>				
Не предусмотрен				
<i>Основной блок</i>				
ОПК-6; ОПК-7	Решение задач (в виде программы)	4	60	По согласованию с преподавателем
		<b>Всего:</b>	<b>60</b>	
ОПК-6; ОПК-7	Экзамен	Вопрос 1	20	В сроки сессии
		Вопрос 2	20	
		<b>Всего:</b>	<b>40</b>	
		<b>Итого:</b>	<b>100</b>	
<i>Дополнительный блок</i>				
ОПК-6; ОПК-7	Выполнение дополнительной лабораторной работы		15	по согласованию с преподавателем
		<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.

## ТЕХНОЛОГИЧЕСКАЯ КАРТА

Шифр дисциплины по РУП		<b>Б1.О.31</b>	
Дисциплина		<b>Программирование</b>	
Курс	<b>2-3</b>	семестр	<b>4-5</b>
Кафедра		<b>Информатики и вычислительной техники</b>	
Ф.И.О. преподавателя, звание, должность		<b>Федоров Андрей Михайлович, канд. техн. наук, доцент кафедры информатики и вычислительной техники</b>	
Общ. трудоемкость <sub>час/ЗЕТ</sub>		<b>36/1</b>	Кол-во семестров
			<b>2</b>
		Форма контроля	<b>Экзамен</b>
ЛК <sub>общ./тек. сем.</sub>	<b>8/4</b>	ПР/СМ <sub>общ./тек. сем.</sub>	<b>-/-</b>
		ЛБ <sub>общ./тек. сем.</sub>	<b>12/6</b>
		СРС <sub>общ./тек. сем.</sub>	<b>286/99</b>

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

- способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий (ОПК-6);
- способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем (ОПК-7).

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>				
ОПК-6; ОПК-7	Решение задач (в виде программы)	4	60	В течение семестра по расписанию занятий
<b>Всего:</b>			<b>60</b>	
ОПК-6; ОПК-7	Экзамен	Вопрос 1	20	В сроки сессии
		Вопрос 2	20	
<b>Всего:</b>			<b>40</b>	
<b>Итого:</b>			<b>100</b>	
<i><b>Дополнительный блок</b></i>				
ОПК-6; ОПК-7	Выполнение дополнительной лабораторной работы		15	по согласованию с преподавателем
<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.