

**Приложение 2 к РПД Программирование  
09.03.02 Информационные системы и технологии  
Направленность (профиль) – Информационные системы и технологии  
Форма обучения – очная  
Год набора - 2015**

**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)	Информационные системы и технологии
4.	Дисциплина (модуль)	Программирование
5.	Форма обучения	очная
6.	Год набора	2015

**2. Перечень компетенций**

- |   |
|---|
| <ul style="list-style-type: none"><li>– владеть широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (ОПК-1);</li><li>– способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований (ПК-25).</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:	
<b>3 семестр</b>					
1. Языки программирования	ОПК-1	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. Неформальное введение в язык C++	ОПК-1	Историческая справка. Пример простейшей программы. Процедуры и функции. Операторы. Выражения. Лексемы. Типы, переменные, константы. Заголовочные файлы (#include<...>). Общий вид C-программы.	уметь использовать языки системы программирования для решения профессиональных задач, работать с программными средствами общего назначения	навыками разработки и отладки программ	экспресс-опрос (тест)
3. Система программирования	ОПК-1 ПК-25	Понятия системы программирования. Техника разработки программ. Классификация ошибок в программе. Отладка программы. Основные понятия отладчика.	работать с современными системами программирования, включая объектно-ориентированные	языками процедурного и объектно-ориентированного программирования	экспресс-опрос (тест)
4. Лексемы языка C++ .	ОПК-1	Спецсимволы. Зарезервированные слова. Идентификаторы. Числа. Метки. Символьные строки. Комментарии. Строки программы	решать задачи по основным разделам курса	методами построения математической модели профессиональных задач и содержательной интерпретации полученных результатов	решение задач (в виде программы), экспресс-опрос (тест)

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
5. Константы языка C++	ОПК-1 ПК-25	Константные выражения. Типизированные константы. Объявление и использование констант.	решать задачи по основным разделам курса	методами построения математической модели	экспресс-опрос (тест)
6. Типы языка C++	ОПК-1 ПК-25	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), экспресс-опрос (тест)
7. Выражения языка C++	ОПК-1 ПК-25	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	экспресс-опрос (тест)
8. Инструкции (операторы) языка C++	ОПК-1 ПК-25	Классификация операторов. Простые операторы. Особенность оператора GOTO. Структурные операторы.	решать задачи по основным разделам курса	навыками разработки и отладки программ	экспресс-опрос (тест)
9. Блочная структура программы на C++	ОПК-1	Понятие блока. Область действия объявления (декларации). Правила видимости. Локальные и глобальные переменные. Распределение памяти.	решать задачи по основным разделам курса	навыками разработки и отладки программ	экспресс-опрос (тест)

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
10. Подпрограммы языка C++	ОПК-1	Понятие подпрограммы. Процедуры и функции. Описание подпрограмм. Способы передачи параметров в подпрограмму и <i>возвращение результата</i> . Процедурные типы.	решать типовые задачи по основным разделам курса, используя методы математического анализа	навыками разработки и отладки программ	экспресс-опрос (тест)
11. Заголовочные файлы языка C++	ОПК-1 ПК-25	Разработка собственных модулей Принципы инкапсуляции и раздельной трансляции кода. Структура модуля: интерфейс, реализация. Использование модулей. Стандартные библиотеки C++.	уметь использовать языки системы программирования для решения профессиональных задач, работать с программными средствами общего назначения	языками процедурного и объектно-ориентированного программирования	экспресс-опрос (тест)
12. Использование файлов в программах на C++	ОПК-1	Классификация файлов. Основные операции с файлами. Библиотеки функций для доступа к файлам (fstream, stdio). Примеры.	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного программирования	решение задач (в виде программы), экспресс-опрос (тест)
<b>4 семестр</b>					
1. Реализация структур данных на языке C++	ОПК-1	Понятие структуры данных. Классификация структур данных. Списки: стек, односвязный список, двусвязный список. Реализация на основе массива и ссылочных типов. Бинарные деревья и операции над ними. Реализация на основе массива и ссылочных типов. Рекурсивные функции.	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного программирования	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:	
2. Объектно-ориентированное программирование	ОПК-1	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), экспресс-опрос (тест)
3. Потокосы классы:	ОПК-1 ПК-25	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), экспресс-опрос (тест)
4. Обработка исключительных ситуаций:	ОПК-1 ПК-25	понятие и использование исключительной ситуации. общий механизм и синтаксис обработки исключений.	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного программирования	экспресс-опрос (тест)
5. Шаблоны функций и классов:	ОПК-1 ПК-25	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), экспресс-опрос (тест)
6. Объектно-ориентированный анализ	ОПК-1 ПК-25	понятие потока и операций работы с ним. Стандартные потоки. Форматирование	решать задачи по основным разделам курса	языками процедурного и объектно-ориентированного	решение задач (в виде программы), экспресс-опрос (тест)

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
		данных. методы обмена с потоками. Ошибки потоков. Файловые потоки. виды потоковых устройств. Потоки и типы, определенные пользователем.		программирования	

#### 4. Критерии и шкалы оценивания

##### 4.1. Тестовый экспресс-опрос (в т.ч. терминологический срез)

Процент правильных ответов	До 60	61-80	81-100
Количество баллов за решенный тест	0	1	2

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

**9 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.

**5 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).

**2 балла** 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.

##### 4.3. Выполнение задания на составление глоссария

	Критерии оценки	Количество баллов
1.	аккуратность и грамотность изложения, работа соответствует по оформлению всем требованиям	1
2.	полнота исследования темы, содержание глоссария соответствует заданной теме	2
	ИТОГО:	3 балла

**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. Типовые задания тестового экспресс-опроса

Цель: проведение оперативного контроля и промежуточной аттестации по итогам освоения разделов (тем) дисциплины, в т.ч. для контроля самостоятельной работы учащихся по отдельным разделам дисциплины.

Раздел 1 "Базовые средства языка C/C++"

<b>Тема 00 "Введение"</b> 1. "Что я жду от учебного курса <Программирование> ?" 2. "Программирование в моей будущей профессии ?"
<b>Тема 01 "Базовые средства языка программирования" 2/2</b> Письменный ответ на вопросы в виде: - ментальной карты понятий; - тезисов - связного текста  1. Что такое структурное программирование? 2. Что относится к базовым средствам языка C++ ? 3. Из каких этапов состоит процесс подготовки программы

для выполнения?

4. Что представляет собою алфавит языка C/C++ ?
5. Что такое идентификаторы и какими они бывают ?
6. Что такое ключевые слова и зачем они нужны?
7. Что такое константы и где они используются?
8. Для чего используются комментарии и какими они должны быть?

*Тема 02 "Базовые средства языка C/C++" 2/4*

*Сформулировать вопросы по разделам указанной Темы:*

- первый вопрос по указанному разделу;
- второй (и более) вопрос по любым выбранным разделам Темы.

- 1) Типы данных C++ ;
- 2) Основные типы данных [];
- 3) Структура программы [];
- 4) Переменные и выражения [];
- 5) Область действия идентификатора [];
- 6) Операции [];
- 7) Выражения ;
- 8) Преобразования типов [];

*Тема 03 "Базовые средства языка C/C++" 2/6*

*Письменно*

*Задание по вариантам:*

- а) Краткое объяснение сути подраздела - "для чего нужен?",*
- б) Для рассматриваемого оператора - обязательный пример программного кода с пояснением*

1. Базовые конструкции структурного программирования
2. Оператор «выражение» []
3. Операторы ветвления []
4. Операторы цикла []
5. Операторы передачи управления []

*б) любой другой вариант по собственному выбору*

*Тема 04 "Базовые средства языка C/C++" 4/6*

*Указатели и массивы*

*1. Что такое указатель? Для чего используется? Какими бывают указатели?*

*Примеры.*

*2. Что означает "инициализация указателя"? Какие способы инициализации бывают?*

*Примеры.*

*3. Какие операции можно выполнять над указателями? Для чего нужны эти операции?*

*Примеры.*

*4. Что такое ссылка? Для чего используется? Чем отличается от указателя?*

*Примеры.*

*5. Что такое "динамический массив"? Для чего используется? Чем отличается от обычного? Примеры.*

*6. Как в языке C представляется строка? Способы инициализации строк. Какие операции можно выполнять над строками? Примеры.*

*Тема 05 "Типы данных, определяемые пользователем " 6/6*

**Контрольное задание по теме "Типы данных, определяемые пользователем"  
[общий вопрос]**

- 1. Для чего в языках программирования нужны и используются "типы данных, определяемые пользователем"?**
- 2. Пояснить назначение, привести и прокомментировать пример использования следующих конструкций языка программирования C/C++:**

[обязательный вариант]

2.1. Структура ;

[вариант по выбору]

2.1. ...

2.2. Переименование типов;

2.3. Перечисление;

2.4. Битовое поле;

2.5. Объединение.

Тема 06 Модульное программирование. Функции. [2/4]

Задание 1

[общие вопросы]

**1. В чем заключается идея модульного программирования?**

**2. С какой целью и как в программах используются функции?**

[вариант по выбору]

**3. Пояснить назначение, привести и прокомментировать пример использования следующих конструкций языка программирования C/C++:**

а. объявление функции;

б. определение функции;

в. вызов функции.

Задание 2

[общие вопросы]

**1. Что означает и как в C++ обеспечивается "взаимодействие функций"?**

**2. Почему взаимодействие функций играет важную роль в модульном программировании?**

[задание по вариантам]

**3. Пояснить назначение, привести и прокомментировать пример использования следующих конструкций языка программирования C/C++:**

а. передача параметров по значению и по ссылке;

б. передача параметров по значению и по указателю;

в. передача массивов в качестве параметра;

г. способы возвращения результатов из функции.

[вариант по выбору]

**4. Пояснить назначение, привести и прокомментировать пример использования следующих конструкций языка программирования C/C++:**

а. параметры функций со значениями по умолчанию;

б. рекурсивные функции;

в. перегрузка функций ;

г. передача параметров в функцию main()

Тема 07 Функции стандартной библиотеки: файлы, строки, символы, математические функции

Контрольное задание по теме " Стандартная библиотека функций" 2/4

[общие вопросы]

**1. Что такое, для чего используется и что в себя включает стандартная библиотека функций C/C++ ?**

**2. Что такое и как в программах на языке C/C++ организуется потоковый ввод/вывод?**

[задание по вариантам]

**3. Пояснить назначение, привести и прокомментировать пример использования следующих конструкций языка программирования C/C++:**

а. открытие потока и ввод из потока;

б. открытие потока и вывод в поток.

**3. Пояснить смысл и назначение следующих элементов программы на языке C/C++:**

в. буферизированный ввод/вывод;

г. используемый в функции `foren(...)` указатель на структуру типа `FILE`;

Контрольное задание по теме "Стандартная библиотека функций" 6/6

[общие вопросы]

**1. Почему в программах на языках программирования высокого уровня математические функции, функции работы со строками и файлами и др. подключаются дополнительно через библиотеки, а не входят в базовый состав языка?**

[задание по вариантам]

**1. Обработку каких ошибок файлового ввода/вывода необходимо предусмотреть в программе? Почему нельзя написать такую программу сразу без этих ошибок? Какие конструкции языка используются для обработки таких ошибок?**

**2. Как в программах на языке C/C++ представляются строки (строковые литералы)? Какие действия можно совершать над строками? Какие существуют способы выполнения этих действий?**

**3. Для чего и как в программах на языках программирования можно использовать графическую подсистему компьютера? Какими примитивами (элементарными объектами) и их свойствами можно при этом управлять?**

*Тема 07\_2 "Возможности графической библиотеки: создание анимации"*

Письменный ответ на вопросы в виде:

- ментальной карты понятий;

- тезисов

- связного текста

- *Графическая подсистема компьютера*
- *Виды компьютерной графики*
- *Функции графической библиотеки*
- *Создание анимированной графики*

*Тема 08 : Директивы препроцессора. Поименованные области*

Контрольное задание по теме "Директивы препроцессора" 2/4

[общие вопросы]

**1. Что такое "препроцессор", для чего и как он используется при разработке программ на языке C\C++?**

[варианты]

1) 1+а, 2+а, 3+б, 4+б;

2) 1+б, 2+б, 3+а, 4+а;

[вопросы]

**1. Что делает препроцессор при выполнении директивы X?**

**2. Как директива X влияет на исходный текст программы?**

**3. Какая цель преследуется программистами при использовании в программе директивы X?**

**4. Если бы не было директивы X, что приходилось бы делать программистам?**

[значение X]

**а. директива `#include`;**

**б. директива `#define`.**

Контрольное задание по теме "Директивы препроцессора" 4/4

[вариант 01]

1. Что такое и для чего используется область действия идентификаторов?
2. В чем заключается "правило одного определения"? В чем смысл этого "правила"?  
[вариант 02]
3. Что такое и для чего используются в программах "внешние объявления"?
4. Что представляет собою, как и зачем используется пространство имен (в т.ч. пространство имен стандартной библиотеки)?

*Тема 09 Кодирование и документирование программ*

Контрольное задание по теме "Кодирование и документирование" 2/2

A.[общий вопрос]

**Что означает принцип кодирования: "лучше по-простому, чем по-умному"?**

**Почему рекомендуется придерживаться именно этого принципа? Что произойдет, если игнорировать этот принцип и другие рекомендации по кодированию, документированию и оформлению программ?**

B.[варианты 01-07]

**1) Почему вместо непосредственного кодирования на языке программирования на первом шаге рекомендуется "записать программу на естественном языке"? Когда и почему этой рекомендацией можно(нельзя) пренебречь?**

**Какие можно дать рекомендации по использованию в программе: 2) функций; 3) переменных и констант; 4) условных и циклических конструкций; 5) динамической памяти; 6) комментариев; 7) форматирования текста исходного кода?**

V.[вариант по выбору] из списка Б.

*Тема 10 Проектирование и тестирование программы*

*Сформулировать вопросы по разделам указанной Темы:*

*- первый вопрос по указанному разделу;*

*- второй (и более) вопрос по любым выбранным разделам Темы.*

"Этапы создания программ"

1. I этап. Постановка задачи
2. II этап. Разработка внутренних структур данных;
3. III этап. Проектирование
4. IV этап. Структурное программирование
5. V этап. Нисходящее тестирование.

**Ключ:** все вопросы открытого типа, что позволяет снизить давление на учащихся ограниченным списком вариантов ответов, и дать им максимально свободные и профессионально творческие ответы во всем многообразии их правильных вариантов.

## 5.2. 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);
}

```

### **5.3. Вопросы к промежуточной аттестации**

#### **3 семестр**

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. Потoki. 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.

#### 4 семестр

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. Поточковые классы: понятие потока и операций работы с ним. Стандартные потоки. Форматирование данных.
21. Понятие конструктора. Типы конструкторов.
22. Поточковые классы: методы обмена с потоками. Ошибки потоков. Файловые потоки.
23. Статические элементы класса.
24. Поточковые классы: виды потоковых 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.

#### 5.4. 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.

**Файлы            ввод/вывод**

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) «Информационные системы и технологии»

(код, направление, профиль)

### ТЕХНОЛОГИЧЕСКАЯ КАРТА

Шифр дисциплины по РУП		<b>Б1.В.ДВ.6.1</b>									
Дисциплина		<b>Программирование</b>									
Курс	<b>2</b>	семестр	<b>3</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>108/3</b>		Кол-во семестров	<b>1</b>		Форма контроля	<b>Экзамен</b>			
ЛК <sub>общ./тек. сем.</sub>	<b>66/32</b>		ПР/СМ <sub>общ./тек. сем.</sub>	<b>-/-</b>		ЛБ <sub>общ./тек. сем.</sub>	<b>66/32</b>		СРС <sub>общ./тек. сем.</sub>	<b>12/8</b>	

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

- владеть широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (ОПК-1);
- способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований (ПК-25).

Код формируемой компетенции	Содержание задания	Количество мероприятий	Максимальное количество баллов	Срок предоставления
<b>Вводный блок</b>				
Не предусмотрен				
<b>Основной блок</b>				
ОПК-1 ПК-25	Экспресс-опрос (тест)	12	24	В начале каждой лекции (проверка знаний предыдущей темы)
ОПК-1 ПК-25	Решение задач (в виде программы)	4	36	По согласованию с преподавателем
<b>Всего:</b>			<b>60</b>	
ОПК-1 ПК-25	Экзамен	Вопрос 1	5	В сроки сессии
		Вопрос 2	5	
		Решение задачи	30	
<b>Всего:</b>			<b>40</b>	
<b>Итого:</b>			<b>100</b>	
<b>Дополнительный блок</b>				
ОПК-1	Выполнение дополнительной лабораторной работы		9	по согласованию с преподавателем
ПК-25	Подготовка глоссария		3	
<b>Всего:</b>			<b>12</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.В.ДВ.6.1</b>			
Дисциплина		<b>Программирование</b>			
Курс	<b>2</b>	семестр	<b>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 <sub>час/ЗЕТ</sub>		<b>108/3</b>	Кол-во семестров	<b>1</b>	Форма контроля
ЛК <sub>общ./тек. сем.</sub>		<b>66/34</b>	ПР/СМ <sub>общ./тек. сем.</sub>	<b>-/-</b>	ЛБ <sub>общ./тек. сем.</sub>
				<b>66/34</b>	СРС <sub>общ./тек. сем.</sub>
					<b>12/4</b>

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

- владеть широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий (ОПК-1);
- способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований (ПК-25).

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-1 ПК-25	Экспресс-опрос (тест)	6	12	В течение семестра
ОПК-1 ПК-25	Решение задач (в виде программы)	5	45	В течение семестра
ОПК-1 ПК-25	Подготовка глоссария	1	3	В течение семестра
<b>Всего:</b>			<b>60</b>	
ОПК-1 ПК-25	Экзамен	Вопрос 1	5	В сроки сессии
		Вопрос 2	5	
		Решение задачи	30	
<b>Всего:</b>			<b>40</b>	
<b>Итого:</b>			<b>100</b>	
<i><b>Дополнительный блок</b></i>				
ОПК-1	Выполнение дополнительной лабораторной работы		9	по согласованию с преподавателем
<b>Всего:</b>			<b>9</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.