

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Амурский государственный университет»

УТВЕРЖДАЮ
Проректор по учебной работе

Н.В.Савина

«19»

20 18 г.



РАБОЧАЯ ПРОГРАММА
Программирование

Направление подготовки 09.03.02 – Информационные системы и технологии

Направленность (профиль) образовательной программы

Безопасность информационных систем

Квалификация выпускника бакалавр

Программа подготовки академический бакалавриат

Год набора 2018

Форма обучения очная

Курс: 1 Семестр: 1, 2

Экзамен 1 семестр (36 академических часов.), 2 семестр (36 академических часов.)

Лекции 54 (академических часов.)

Лабораторные работы 72 (академических часов.)

Самостоятельная работа 162 (академических часов.)

Общая трудоемкость дисциплины 360 (академических часов.), 10 (з.е.)

Составитель Т.А. Галаган, доцент, канд. техн. наук

Факультет математики и информатики

Кафедра информационных и управляющих систем

2018 г.

Рабочая программа составлена на основании Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии», утвержденного приказом Министерства образования и науки 12.03.2015 г., № 219

Рабочая программа обсуждена на заседании кафедры информационных и управляющих систем

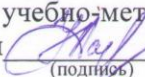
« 15 » 05 2018 г., протокол № 9

Заведующий кафедрой  А.В. Бушманов
подпись И.О.Ф.

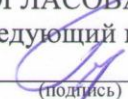
Рабочая программа одобрена на заседании УМС направления подготовки 09.03.02 «Информационные системы и технологии»

« 29 » 05 2018 г., протокол №9

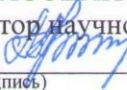
Председатель  А. В. Бушманов
подпись И.О.Ф.

СОГЛАСОВАНО
Начальник учебно-методического
управления  Н.А. Чалкина
(подпись)

« 29 » 05 2018 г.

СОГЛАСОВАНО
Заведующий выпускающей кафедрой
 А. В. Бушманов
(подпись)

« 15 » 05 2018 г.

СОГЛАСОВАНО
Директор научной библиотеки
 Л.А. Проказина
(подпись)

« 29 » 05 2018 г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целями освоения дисциплины «Программирование» являются изучение основных принципов процедурного, модульного и объектно-ориентированного программирования; обучение правилам и подходам к разработке и отладке программного обеспечения на языке программирования высокого уровня C++.

Задачи дисциплины: изучение основных синтаксических конструкций языка программирования C++, правил и рекомендаций построения программ на указанном языке; обучение способам разработки эффективных алгоритмов программ и приемам использования встроенных возможностей указанного языка программирования и дополнительно подключаемых библиотек; привитие устойчивых практических навыков применения современных инструментальных средств при разработке программного обеспечения, их отладки.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Программирование» относится к дисциплинам базовой части.

Для изучения дисциплины «Программирование» студент должен обладать стартовыми навыками алгоритмизации, уметь анализировать и обобщать информацию, владеть первоначальными навыками работы с компьютером, желательно обладать аналитическим складом мышления, что могло быть получено в результате изучения предметов «Информатика» и «Математика» в объеме образовательной программы средней школы.

Изучения дисциплины «Программирование» является основой для изучения дальнейших дисциплин, использующих ЭВМ и программирование, таких как «Объектно-ориентированный анализ, программирование», «Базы данных», «Системное программное обеспечение» и других.

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Обучение студентов данной дисциплине должно способствовать развитию следующей компетенции:

способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи (ОПК-6);

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

Знать современные технические и программные средства взаимодействия с ЭВМ; технологию разработки алгоритмов и программ; методы отладки и решения задач на ЭВМ; основы объектно-ориентированного подхода к программированию (ОПК-6).

Уметь ставить задачу, разрабатывать алгоритм ее решения, работать с современными системами программирования, включая объектно-ориентированные (на примере языка C++) (ОПК-6).

Владеть парадигмами структурного, процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ на языке программирования высокого уровня C++ (ОПК-6).

4. МАТРИЦА КОМПЕТЕНЦИЙ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

Разделы дисциплины	Компетенции
	ОПК-6
1	2
Введение в программирование	+
Программы разветвляющейся структуры	+
Программы циклической структуры	+

1	2
Обработка одномерных массивов	+
Обработка двумерных массивов	+
Создание пользовательских функций	+
Указатели и ссылки. Средства использования динамической памяти	+
Типы данных, определяемые пользователем	+
Файловый ввод-вывод	+
Основы объектно-ориентированного программирования	+

5. СТРУКТУРА ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 10 зачетных единиц, 360 акад. часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды контактной работы, включая самостоятельную работу студентов и трудоемкость (в академических часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				лекции	лаб. раб	сам раб	
1	Введение в программирование	1	1, 2	4	4	10	Тестовое задание
2	Программы разветвляющейся структуры	1	3 – 6	6	8	20	Тестовое задание
3	Программы циклической структуры	1	7, 8	6	4	10	Тестовое задание
4	Обработка одномерных массивов	1	9 – 13	8	8	20	Тестовое задание Кейс-задание
5	Обработка двумерных массивов	1	14 – 16	8	8	20	Тестовое задание
6	Создание пользовательских функций.	1	17, 18	4	4	10	Тестовое задание Кейс-задание
7	Указатели и ссылки. Средства использования динамической памяти	2	1– 4	4	6	12	Кейс-задание
8	Типы данных, определяемые пользователем	2	5 – 8	4	12	24	Тестовое задание
9	Файловый ввод-вывод	2	9 – 12	4	6	12	Тестовое задание
10	Основы объектно-ориентированного программирования	2	13 – 18	6	12	24	Тестовое задание Кейс-задание
	ИТОГО 360 акад. часов			54	72	162	Экзамен 1 семестр (36 акад. час.), 2 семестр (36 акад. час.)

6. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1 Лекции

№ п/п	Наименование разделов	Содержание разделов
1	2	3
1	Введение в программирование	Этапы создания программы, языки программирования

1	2	3
	рование	вания, структура системы программирования. Общая характеристика языка C++. Основные конструкции языка: алфавит, идентификаторы, ключевые слова. Структура программы на языке C++. Простые типы данных. Переменные и константы. Основные операции. Арифметические и логические выражения. Операторы ввода-вывода библиотеки <code>iostream.h</code> . Использование манипуляторов вывода. Библиотека математических функций <code>math.h</code> .
2	Программы разветвляющейся структуры	Понятие алгоритма. Правила составления блок-схем алгоритмов. Линейные и разветвляющиеся алгоритмы. Условный оператор <code>if</code> . Правила вычисления логических выражений. Оператор выбора <code>switch</code> .
3	Программы циклической структуры	Циклические алгоритмы. Операторы цикла языка C++: с предусловием (<code>while</code>), с постусловием (<code>do while</code>), с заданным числом повторений (<code>for</code>). Операторы передачи управления – <code>break</code> , <code>return</code> , <code>continue</code> , <code>go to</code> .
4	Обработка одномерных массивов	Объявление, инициализация, обработка одномерных массивов. Алгоритмы нахождения минимального и максимального значений, суммы и произведения элементов массива. Алгоритмы сортировки: метод «пузырька», метод прямого выбора. Функции обработки символьных строк библиотеки <code>string.h</code> .
5	Обработка двумерных массивов	Объявление, инициализация двумерных массивов различных типов. Ввод-вывод элементов двумерного массива. Обработка элементов двумерных массивов случайным образом, по строкам, по столбцам.
6	Создание пользовательских функций.	Объявление и определение функций. Фактические и формальные параметры. Понятие прототипа функции. Вызов функции. Передача значений с использованием оператора <code>return</code> . Область действия и время жизни переменной. Понятие рекурсии.
7	Указатели и ссылки. Средства использования динамической памяти	Объявление указателей. Основные операции над ними. Связь между указателями и массивами. Понятие ссылки. Использование ссылок для передачи значений из функции. Функции динамического распределения памяти <code>new()</code> , <code>delete()</code> .
8	Типы данных, определяемые пользователем	Перечисляемый тип. Переименование типов с помощью <code>typedef</code> . Структурный шаблон и синтаксис его объявления. Понятие «поле шаблона». Объявление, инициализация и обработка структурных переменных. Указатели на структуру. Вложенные структуры. Переменные типа объединение, особенности их использования.
9	Файловый ввод-вывод	Виды файлов: текстовые и бинарные. Функции открытия и закрытия файла. Особенности обработки данных текстовых и бинарных файлов. Функции чтения, записи, перемещения внутреннего указателя.
10	Основы объектно-ориентированного программирования	Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм. Определение класса в C++. Поля и методы класса. Спецификаторы управления доступом. Операция разре-

1	2	3
		шения видимости. Объекты. Виды конструкторов. Деструкторы. Производный класс. Простое и множественное наследование. Правила объявления конструкторов и деструкторов в производных классах. Перегрузка функций и операций. Виртуальные функции. Абстрактный класс.

6.2 Лабораторные работы

6.2.1. Изучение среды программирования Dev C++, обработка простейшей программы.

6.2.2. Разветвляющиеся программы.

6.2.3. Программы циклической структуры.

6.2.4. Обработка одномерных массивов.

6.2.6. Обработка двумерных массивов.

6.2.7. Создание пользовательских функций.

6.2.8. Работа с указателями.

6.2.9. Использование ссылок для передачи значений из функции.

6.2.10. Обработка структурных данных с использованием методов распределения динамической памяти.

6.2.11. Работа с файлами данных.

6.2.12. Создание классов. Конструкторы, деструкторы.

6.2.13. Реализация принципа наследования.

6.2.14. Перегрузка операций.

6.2.15. Виртуальные функции и абстрактные классы.

7. САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	Раздел дисциплины	Форма (вид) самостоятельной работы	Трудоемкость в академических часах
1	2	3	4
1	Введение в программирование	Изучение учебной литературы Приобретение навыков работы в среде программирования	10
2	Программы разветвляющейся структуры	Изучение учебной литературы Подготовка отчета по лабораторной работе	20
3	Программы циклической структуры	Изучение учебной литературы Подготовка к контрольной работе Подготовка отчета по лабораторной работе	10
4	Обработка одномерных массивов	Изучение учебной литературы Подготовка отчета по лабораторной работе	20
5	Обработка двумерных массивов	Изучение учебной литературы Подготовка к контрольной работе Подготовка отчета по лабораторной работе	20
6	Создание пользовательских функций.	Изучение учебной литературы Подготовка к контрольной работе Подготовка отчета по лабораторной работе	10
7	Указатели и ссылки. Средства использования динамической памяти	Изучение учебной литературы Подготовка отчетов по лабораторным работам	12

1	2	3	4
8	Типы данных, определяемые пользователем	Изучение учебной литературы Подготовка к контрольной работе Подготовка отчета по лабораторной работе	24
9	Файловый ввод-вывод	Изучение учебной литературы Подготовка отчета по лабораторным работам	12
10	Основы объектно-ориентированного программирования	Изучение учебной литературы Подготовка к контрольной работе Подготовка отчета по лабораторным работам	24
	Итого		162

Перечень учебно-методического обеспечения для самостоятельной работы

1 Язык С++. Сборник задач и тестовых заданий. Учеб. пособие / Т.А. Галаган. – Благовещенск: изд-во Амур. гос. ун-та, 2014 – 112 с. Режим доступа http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/6747.pdf

2 Галаган, Т.А. Алгоритмические языки и программирование. Язык С++. Курс лекций (Рек. ДВРУМЦ) / Т.А. Галаган – Благовещенск: Амурский гос. ун-т, 2007. – 147 с.

3 Галаган, Т.А. Объектно-ориентированное программирование. Язык С++. Учебное пособие/ Т.А. Галаган – Благовещенск: изд-во Амур. гос. ун-та, 2016. – 56 с. http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/9699.pdf

4 Галаган, Т.А. Программирование. Сб. учебн.-метод. материалов для направлений подготовки 09.03.01, 09.03.02 / сост. Т.А. Галаган Благовещенск: изд-во Амур. гос. ун-та, 2017. – 62 с. http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/7730.pdf

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Образовательная технология – система, включающая в себя конкретное представление планируемых результатов обучения, форму обучения, порядок взаимодействия студента и преподавателя, методики и средства обучения, систему диагностики текущего состояния учебного процесса и степени обучения студента.

К образовательным технологиям, используемым в преподавании данной дисциплины, относятся лекции и лабораторные работы.

В изложении лекционного материала наряду с традиционной лекцией используются такие неимитационные методы обучения, как:

проблемная лекция, начинающаяся с постановки проблемы, которую необходимо решить в ходе изложения материала,

лекция с заранее запланированными ошибками, которые студенты должны обнаружить самостоятельно в конце лекции.

На лекциях используются информационные технологии – презентации. Лабораторные работы проводятся в компьютерных классах и предназначены для решения прикладных задач с использованием современных инструментальных средств.

При проведении лабораторных работ используются неигровые имитационные методы обучения:

контекстное обучение, направленное на решение профессиональных задач,

работа в команде – совместная деятельность студентов в группе, направленная на решение общей задачи с разделением ответственности и полномочий.

Методы и формы организации обучения

Методы	Лекция	Лабораторная работа	СРС
1	2	3	4

1	2	3	4
ИТ-обучение	+	+	+
Работа в команде		+	
Методы проблемного обучения	+	+	+
Неигровые имитационные методы	+	+	
Проблемный метод	+		+
Опережающая самостоятельная работа			+

Удельный вес занятий, проводимых в интерактивной форме, составляет: 28 акад. часа, 12 акад. часов лекций и 16 акад. часов лабораторных работ.

№ п/п	№ раздела (темы) дисциплины	Форма (вид) образовательных технологий	Кол-во акад. часов
1	Введение в программирование	Мультимедийная лекции	2
2	Программы разветвляющейся структуры	Работа в команде – лабораторная работа	4
3	Программы циклической структуры	Работа в команде – лабораторная работа	2
4	Обработка одномерных массивов	Лекция с ошибками	2
5	Обработка двумерных массивов	Мультимедийная лекции	4
6	Создание пользовательских функций.	Работа в команде – лабораторная работа	2
7	Указатели и ссылки. Средства использования динамической памяти	Лекция с ошибками	2
8	Типы данных, определяемые пользователем	Контекстное обучение, направленное на решение профессиональных задач	8
9	Файловый ввод-вывод		
10	Основы объектно-ориентированного программирования	Проблемная лекция	2

9. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания, типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков, характеризующих этапы формирования компетенций, а так же методические материалы, определяющие процедуры оценивания знаний, умений и навыков, отражены в фонде оценочных средств по дисциплине «Программирование».

Для оценки текущей успеваемости в данной дисциплине относятся: тестовые задания с закрытыми и открытыми видами вопросов; кейс-задания, отчеты по выполнению лабораторных работ; экзамен.

Вопросы к экзамену 1 семестра

1 Средства реализации программного обеспечения информационных систем

1.1 Этапы создания и реализации программы

1.2 Языки программирования, структура системы программирования.

1.3 Язык C++ как средство реализации программного обеспечения информационных систем:

машинописных систем:

- 1.3.1 Состав языка C++
- 1.3.2 Структура программы языка C++
- 1.3.3 Переменные, идентификаторы
- 1.3.4 Типы данных языка C++
- 1.3.5 Описание констант и переменных. Инициализация переменных
- 1.3.6 Основные операции языка C++
- 1.3.7 Директива препроцессора #include
- 1.3.8 Ввод-вывод с использованием библиотеки iostream.h, iomanip.h
- 1.3.9 Функции ввода-вывода библиотеки stdio.h
- 1.3.10 Директива препроцессора #define
- 1.3.11 Библиотека математических функций math.h
- 1.3.12 Решение простейших задач с линейной структурой алгоритма
- 2 Реализация программного обеспечения на основе структурного программирования
 - 2.1 Условный оператор if
 - 2.2 Множественный выбор: оператор switch
 - 2.3 Решение задач с разветвляющейся структурой
 - 2.4 Инструкции перехода (goto, continue, return, break)
 - 2.5 Оператор цикла с предусловием
 - 2.6 Оператор цикла с постусловием
 - 2.7 Решение задач циклической структуры(подсчет сумму(произведения) значений числового ряда, вывод значений функции на интервале)
 - 2.8 Оператор цикла с заданным числом повторений
 - 2.9 Одномерные массивы (объявление, инициализация, задание значений)
 - 2.8 Использование счетчика случайных чисел для задания значений переменных и массивов
 - 2.9 Решение задач обработки числовых значений одномерных массивов (нахождение суммы (произведения) всех элементов массива или частично (по некоторому условию), нахождение значения максимума (минимума) из всех элементов или по некоторому условию)
 - 2.10 Решение задач переупорядочивания элементов в массиве (методы сортировки «пузырька», метод прямого выбора)
 - 2.11 Двумерные массивы (объявление, инициализация, ввод значений, вывод в общепринятом виде)
 - 2.12 Решение задач обработки двумерных массивов по строкам
 - 2.13 Решение задач обработки массивов по столбцам
 - 2.14 Строки (объявление, инициализация). Функции работы со строками библиотеки string.h
 - 2.15 Решение задач обработки символьных строк
- 3 Реализация программного обеспечения на основе процедурной (модульной) парадигмы программирования
 - 3.1. Определение, вызов пользовательских функций
 - 3.2. Понятие прототипа функции
 - 3.3. Формальные и фактические параметры функции
 - 3.4. Понятие рекурсии
 - 3.5. Область действия и время жизни переменных

Вопросы к экзамену 2 семестра

- 1 Реализация программного обеспечения на основе процедурной (модульной) парадигмы программирования
 - 1.1. Объявление, инициализация указателей, операции с указателями
 - 1.2. Связь массивов и указателей
 - 1.3. Передача массива в функцию
 - 1.4. Решения задач обработки массивов способом обращения к значениям че-

рез указатели

- 1.5. Ссылки. Передача аргументов функции по ссылке
- 1.6. Перечисляемый тип
- 1.6. Объявления typedef
- 1.7. Объявление структурного шаблона и структурной переменной, инициализация, обращение к полям структурной переменной.
- 1.8. Работа со структурами через указатели. Передача структуры в функцию
- 1.9. Объединения
- 1.10. Решение задач обработки структурированных данных
- 1.11. Текстовые и бинарные файлы
- 1.12. Функции работы с файлами fopen(), fclose()
- 1.13. Функции работы с файлами fwrite(), fread(), fseek()
- 1.14. Чтение и запись текстовых файлов
- 1.15. Работа с файлами через потоки ввода-вывода
- 1.16. Решение задач обработки данных, организованных в файлы
- 2 Реализация программного обеспечения на основе объектно-ориентированной парадигмы программирования
 - 2.1. Ключевые принципы объектно-ориентированного программирования
 - 2.2. Определение класса (поля, методы, объекты)
 - 2.3. Спецификаторы доступа, особенности их применения к элементам класса
 - 2.4. Конструктор (объявление, вызов, виды)
 - 2.5. Деструктор
 - 2.6. Указатель this
 - 2.7. Статические элементы класса
 - 2.8. Перегрузка операций
 - 2.9. Принципы простого наследования.
 - 2.10. Особенности применения ключей доступа в наследовании
 - 2.11. Множественное наследование
 - 2.12. Виртуальные функции
 - 2.13. Абстрактные классы
 - 2.14. Решение задач на основе объектно-ориентированной парадигмы программирования

10. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОСНОВНАЯ ЛИТЕРАТУРА

- 1 Павловская, Т.А. С/С++. Программирование на языке высокого уровня (Допущено МинОбр РФ) – СПб.: Питер, 2009, 2010. – 461 с.
- 2 Галаган, Т.А. Алгоритмические языки и программирование. Язык С++. Курс лекций (Рек. ДВРУМЦ) / Т.А. Галаган – Благовещенск: изд-во АмГУ, 2007. – 147 с.

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

- 1 Павловская, Т.А. С/С++ Структурное программирование. Практикум. / Т.А. Павловская, Ю.А. Щупак. – СПб.: Питер, 2004. – 239 с.
- 2 Тяпичев Г.А. Быстрое программирование на С++ [Электронный ресурс] / Г.А. Тяпичев. — Электрон. текстовые данные. — М. : СОЛОН-ПРЕСС, 2010. — 373 с. — 5-98003-162-6. — Режим доступа: <http://www.iprbookshop.ru/65094.html>
- 3 Программирование на языке высокого уровня С/С++ [Электронный ресурс]: конспект лекций/ — Электрон. текстовые данные. – М.: Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2016. – 140 с. – Режим доступа: <http://www.iprbookshop.ru/48037>. – ЭБС «IPRbooks», по паролю

Программное обеспечение и интернет-ресурсы

	Наименование ресурса	Характеристика
1	2	3
1	http://www.intuit.ru	ИНТУИТ – сайт, который предоставляет возможность дистанционного обучения по нескольким образовательным программам, касающимся, в основном, информационных технологий. Содержит несколько сотен открытых образовательных курсов.
2	http://www.window.edu.ru	Единое окно доступа к образовательным ресурсам/ каталог/ профессиональное образование
3	http://e.lanbook.com/	Электронная библиотечная система «Издательства «Лань», тематические пакеты: математика, физика, инженерно-технические науки
4	http://iprbookshop.ru/	Научная электронная библиотека IPRbooks – научно-образовательный ресурс для решения задач обучения в России и за рубежом. Уникальная платформа ЭБС IPRbooks объединяет новейшие информационные технологии и учебную лицензионную литературу. Контент IPRbooks отвечает требованиям стандартов высшей школы, дополнительного и дистанционного образования.
5	Dev C++	Среда программирования на языке C++ , бесплатное распространение по лицензии GNU ссылка: http://www.gnu.org/licenses/agpl-3.0.html
6	Операционная система Windows 7 Pro	Dream Spark Premium Electronic Software Delivery (3 years) Renewal по договору – Сублицензионный договор № Tr000074357/КНВ 17 от 01 марта 2016 года

11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Сценарий «изучения дисциплины»

При изучении дисциплины применяется рейтинговая технология обучения, которая позволяет реализовать непрерывную и комплексную систему оценивания учебных достижений студентов. Непрерывность означает, что текущие оценки не усредняются, а непрерывно складываются на протяжении одного семестра. Комплексность означает учет всех форм учебной и творческой работы студента в течение семестра.

Рейтинг направлен на повышение ритмичности и эффективности самостоятельной работы студентов. Он основывается на заинтересованности каждого студента в получении более высокой оценки знаний по дисциплине.

Принципы рейтинга: непрерывный контроль и получение более высокой оценки за работу, выполненную в срок. Рейтинг включает в себя три вида контроля: текущий, промежуточный и итоговый по дисциплине.

Текущий контроль – это выполнение лабораторных работ по темам дисциплины, тестирование по разделам дисциплины.

Рекомендации по планированию и организации времени, необходимого на изучение дисциплины и выполнения лабораторных работ

Задания к лабораторным работам формируются на основе материала, изложенного на лекциях. Последовательность тем заданий также соответствует последовательности изложения лекционного материала. Задания выполняются индивидуально, либо в маленьких рабочих группах, сформированных для решения определенной задачи.

Каждый студент (рабочая группа) получает индивидуальный вариант для выполне-

ния задания лабораторной работы. Перечень вариантов заданий приведен в учебно-методическом пособии: Язык С++. Сборник задач и тестовых заданий. Для студентов направлений подготовки 230100.62 «Информатика и вычислительная техника», 230400.62 «Информационные системы и технологии», 230700.62 – «Прикладная информатика». – Благовещенск: Амурский гос. ун-т, 2013 – 78 с.

Наряду с вариантами заданий данное пособие содержит контрольные вопросы и тестовые задания, выполнение которых поможет студенту успешно справиться с заданием.

Задания к лабораторным работам выдаются заранее, как правило, в начале семестра, и для их успешного их выполнения необходимо предварительное освоение теоретического материала и разбор, приведенных на лекции примеров программ, проработка алгоритма решения разобранных задач. Для этого наряду с конспектами можно воспользоваться учебно-методическим обеспечением для самостоятельной работы, указанным в рабочей программе, и самопроверкой с помощью тестовых заданий, размещенных там же.

Для подготовки к выполнению лабораторных работ и повторения, усвоения (изучения пропущенного) теоретического материала студентам рекомендуется самостоятельно организовать по месту проживания дополнительное рабочее место, оборудованное персональным компьютером, подключённым к сети Интернет, и установленным программным обеспечением, необходимым для разработки программ и указанным в рабочей программе.

Для успешной сдачи лабораторной работы и получения максимальных баллов за нее необходимо не только создать работоспособное приложение, но и использовать эффективные алгоритмы, а также привести в отчете результаты тестирования разработанной программы. Количество начисленных за работу баллов также зависит от срока сдачи работы, который указан в каждой работе при ее выдаче.

Общие рекомендации по созданию программы на языке программирования

Первый шаг состоит в продумывании наборов входных и выходных данных и записи алгоритма будущей программы на естественном языке или с использованием блок-схемы. Это позволяет продумать алгоритм в деталях, разбить программу на логические блоки, определить их последовательность, продумать комментарии к программе.

Необходимо стараться (если это возможно) разбить алгоритм на последовательность законченных действий. Каждое из них можно оформить в виде функции.

Функция не должна быть слишком большой, как правило, ее текст должен не превышать два экрана.

При расположении последовательности функций нужно помнить, что функция может быть вызвана только после ее объявления. Если программа содержит две-три функции, такой порядок достаточно легко определить. Иначе лучше использовать прототипы функций, расположив их в начале программы, сразу после директив препроцессора.

Всю информацию, требуемую для работы функции, необходимо передавать в качестве параметров. В вызове функции строго соблюдать соответствие количества, типов и порядок следования фактических параметров – формальным.

Желательно инициализировать переменные при их объявлении, а объявлять их как можно ближе к месту непосредственного использования. В небольших функциях удобно все объявления локальных переменных располагать в начале блока.

Если ввод переменных осуществляется через клавиатуру, требуется предварять его выводом сообщения на экран, которое обязательно должно быть информативным.

Использование локальных переменных предпочтительнее глобальных. При использовании ветвления следует избегать проверки лишних условий. Неэффективным кодом считается проверка на равенство нулю. Например, вместо `if (k == 0)` лучше писать `if (k)`.

При организации циклов лучше размещать инициализацию и приращения счетчика, проверку условия выхода из цикла в одном месте. Требуется предусматривать аварийный выход из цикла по достижению заданного максимального количества итераций.

Написанный программный код нужно тщательно отредактировать. Убрать ненуж-

ные фрагменты, сгруппировать описания, оптимизировать проверки условий, проверить условия выхода из циклов, проверить оптимальность разбиения на функции.

Следует сопроводить программу комментариями. Для грамотно написанной программы недостаточно подтверждения ее работы и даже получения верного результата. Программа должна иметь четкую структурированность, наглядность, читаемость, сопровождение комментариями, возможность легкой модификации, и желательно, грамотно разработанный, эффективный алгоритм.

По завершению кодирования программы, ее необходимо откомпилировать и выполнить.

Однократное выполнение программы не гарантирует правильность ее работы. Для подтверждения этого следует приготовить наборы исходных данных (тесты), рассчитать предварительно результат. После выполнения программы убедиться, что полученные результаты совпали с подготовленными тестовыми заданиями.

Рекомендации по подготовке и защите отчета к лабораторным работам

Отчет выполняется в электронном варианте и включает в себя: постановку задачи (задание), набор входных и выходных данных с описанием их типов, описание алгоритма программы, тестовые примеры и результаты их выполнения.

При защите отчета студент демонстрирует полученные результаты и отвечает на вопросы преподавателя, связанные с используемым теоретическим материалом и особенностями созданной программы.

Рекомендации по работе с литературой

Ввиду высокой скорости устаревания издаваемой учебной литературы по информационным технологиям, вследствие активной ежегодной модернизации комплексов аппаратно-программных средств и сопутствующей инфраструктуры информационного обеспечения, студентам рекомендуется в первую очередь ориентироваться на работу с конспектами лекций текущего года.

Советы по подготовке к экзамену

Итоговый контроль – экзамен, проводимый в 1 и 2 семестрах на основании перечня вопросов, представленных в рабочей программе. Билет включает два теоретических вопроса и задачу. Примерное содержание билета приведено в фонде оценочных средств.

Для подготовки к экзамену рекомендуется использовать конспекты лекций, рекомендованную в рабочей программе литературу, ЭВМ и все теоретические знания, и практические навыки, полученные во время проведения всех видов занятий.

12. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

В качестве основных технических средств обучения по дисциплине «Программирование» используются:

- мультимедийные лекционные аудитории, оснащенные проектором, обеспечивающим воспроизводство слайдов и текстов с экрана монитора компьютер лектора, управляющим компьютером, устройствами затемнения, обеспечения информационной безопасности и поддержания микроклимата;

- дисплейные классы кафедры информационных и управляющих систем АмГУ, оборудованные компьютерами, подключенные к ЛВС университета с возможностью подключения сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.

В качестве программного обеспечения используются средства, указанные в п.10 данного документа.

Самостоятельная работа обучающихся осуществляется в помещениях для самостоятельной работы, оснащенных компьютерной техникой с возможностью подключения сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.

13. РЕЙТИНГОВАЯ ОЦЕНКА ЗНАНИЙ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ

13.1. Балльная структура оценки за первый семестр

Семестровый модуль дисциплины						
Учебные модули		Виды контроля	Сроки выполнения (недели)	Макс. кол-во баллов	Посещение занятий, активность	Итого макс. кол-во баллов
1	Введение в программирование	отчет по лаб. раб.	1 - 4	5	2	7
2	Программы разветвляющейся структуры	отчет по лаб. раб.	5, 6	9	1	10
3	Программы циклической структуры	отчет по лаб. раб. тест №1	7, 8	7	1	8
4	Обработка одномерных массивов	отчет по лаб. раб. кейс-задание	9 - 13	13	2,5	15,5
5	Обработка двумерных массивов	отчет по лаб. раб. тест №2	14 - 16	11	1,5	12,5
6	Создание пользовательских функций.	отчет по лаб. раб. кейс-задание	17, 18	7	1	8
	Сдача экзамена			40		40
	Итого			92	8	100

13.2. Балльная структура оценки за второй семестр

Семестровый модуль дисциплины						
Учебные модули		Виды контроля	Сроки выполнения (недели)	Макс. кол-во баллов	Посещение занятий, активность	Итого макс. кол-во баллов
1	2	3	4	5	6	7
1	Указатели и ссылки. Средства использования динамической памяти	отчет по лаб. раб.	1- 4	12	2	14
2	Типы данных, определяемые пользователем	отчет по лаб. раб. тест №3	5 – 8	14	2	16
3	Файловый ввод-вывод	отчет по лаб. раб.	9 – 12	10	2	12
4	Основы объектно-ориентированного программирования	отчет по лаб. раб. тест №4 кейс-задание	13 – 18	16	2	18
	Сдача экзамена			40		40
	Итого			92	8	100

13.3. Итоговая оценка

Сумма баллов, набранных в течение семестра (с возможностью проставления предварительной оценки за экзамен)	Общая сумма баллов (с учетом сдачи экзамена в период семестровой аттестации)	Итоговая оценка
56 – 60	91 – 100	отлично
51 – 55	75 – 90	хорошо
46 – 50	51 – 74	удовлетворительно
<46	< 51	неудовлетворительно