

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ
Декан факультета вычислитель-
ной техники
Фионова Л. Р.
«14 сентября 2015 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

М1.2.7.1 Современное программирование

Направление подготовки 01.04.02. Прикладная математика и информатика

Магистерская программа Математическое и программное обеспечение вычислительных машин

Квалификация (степень) выпускника - Магистр

Форма обучения Очная

Пенза, 2015

1. Цели освоения дисциплины

Целями освоения дисциплины «Современное программирование» является формирование и развитие у будущих магистров прикладной математики и информатики общекультурных и профессиональных компетенций, формирование систематизированных знаний в области алгоритмов, технических и программных средств, применяемых для создания программного обеспечения с использованием объектно-ориентированного языка программирования C++.

2. Место дисциплины в структуре ОПОП магистратуры

Дисциплина «Современное программирование» относится к дисциплинам по выбору вариативной части блока М1.

Для освоения дисциплины магистранты используют знания, умения, навыки, сформированные в процессе изучения предметов «Алгоритмы и алгоритмические языки», «Языки и методы программирования», «Объектно-ориентированное программирование» соответствующего направления подготовки бакалавров.

Освоение данной дисциплины является основой для последующего освоения предметов, подготовки к итоговой государственной аттестации.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины «Современное программирование»

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ФГОС ВПО по данному направлению:

Коды компетенции	Наименование компетенции	Структурные элементы компетенции (в результате освоения дисциплины обучающийся должен знать, уметь, владеть)
1	2	3
ОПК-4	Способность использовать углубленные теоретические и практические знания в области прикладной математики и информатики	Знать: теоретические и практические принципы современного программирования
		Уметь: использовать углубленные теоретические и практические знания современного программирования для решения задач в области прикладной математики и информатики
		Владеть: практическими навыками использования теоретических и практических знаний современного программирования для решения задач в области прикладной математики и информатики
ПСК-1	Способность к формализации и алгоритмизации поставленных задач	Знать: базовые алгоритмы стандартной библиотеки шаблонов
		Уметь формализовать поставленные задачи и разрабатывать алгоритмы их решения в области прикладной математики и информатики
		Владеть: практическими навыками формализации и алгоритмизации поставленных задач в области прикладной математики и информатики
ПСК-2	Способен к написанию программного кода с использо-	Знать: технологии современного программирования разработки программных приложений

	ванием языков программирования, определения и манипулирования данными	Уметь: разрабатывать программный код с использованием принципов современного программирования
		Владеть: практическими навыками разработки программного кода с использованием принципов современного программирования

4. Структура и содержание дисциплины Современное программирование

4.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часа.

№ п/п	Наименование разделов и тем дисциплины (модуля)	Семестр	Недели семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)								Формы текущего контроля успеваемости (по неделям семестра)		
				Аудиторная работа			Самостоятельная работа					Опрос на лабораторном занятии	Проверка выполнения индивидуального задания	Проверка отчёта выполнения лабораторной работы
				Всего	Лекция	Лабораторные занятия	Всего	Подготовка к аудиторным занятиям	Выполнение домашнего индивидуального задания	Подготовка отчёта лабораторной работы	Подготовка к экзамену			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1.	Раздел 1. Новые средства языка C++	1	1-4	10	2	8	10	2	6	2				
1.1.	Тема 1.1. Новые средства языка C++		1	2	2		2	2						
1.2.	Лабораторная работа 1. Изменения объявлений и определений объектов, введённые в новый стандарт языка C++		1-2	4		4	4		3	1		1	2	3
1.3.	Лабораторная работа 2. Обработка строк в новом стандарте языка C++		3-4	4		4	4		3	1		3	4	5
2.	Раздел 2. Стандартная библиотека шаблонов	1	3	2	2		2	2						
2.1.	Тема 2.1. Стандартная библиотека шаблонов		3	2	2		2	2						
3	Раздел 3. Контейнеры STL	1	5-12	22	6	16	22	6	12	4				
3.1.	Тема 3.1. Последовательные контейнеры		5	2	2		2	2						

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3.2.	Тема 3.2. Ассоциативные контейнеры		7	2	2		2	2						
3.3	Тема 3.3. Неупорядоченные контейнеры		9	2	2		2	2						
3.4.	Лабораторная работа 3. Обработка векторов в новом стандарте языка C++		5-6	4		4	4		3	1		5	6	7
3.5.	Лабораторная работа 4. Обработка списков в новом стандарте языка C++		7-8	4		4	4		3	1		7	8	9
3.6.	Лабораторная работа 5. Обработка множеств в новом стандарте языка C++		9-10	4		4	4		3	1		9	10	11
3.7.	Лабораторная работа 6. Обработка отображений в новом стандарте языка C++		11-12	4		4	4		3	1		11	12	13
4	Раздел 4. Итераторы STL	1	11	6	2		2	2						
4.1.	Тема 4.1. Итераторы STL		11	2	2		2	2						
5	Раздел 5. Алгоритмы STL	1	13-15	8	4	4	8	4	3	1				
5.1.	Тема 5.1. Немодифицирующие алгоритмы		13	2	2		2	2						
5.2	Тема 5.2. Модифицирующие алгоритмы		15	2	2		2	2						
5.3.	Лабораторная работа 7. Применение обобщённых алгоритмов в новом стандарте языка C++		13-14	4		4	4		3	1		13	14	15
6.	Раздел 6. Функциональные объекты STL и лямбда-функции	1	15-18	10	2	8	10	2	6	2				
6.1.	Тема 6.1. Функциональные объекты STL и лямбда-функции		17	2	2		2	2						
6.2	Лабораторная работа 8. Применение функциональных объектов для обработки коллекций в обобщённых алгоритмах		15-16	4		4	4		3	1		15	16	17

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6.3	Лабораторная работа 9. Применение лямбда-выражений для обработки коллекций в обобщённых алгоритмах		17-18	4		4	4		3	1		17	18	
	Подготовка к экзамену						36				36			
	Общая трудоемкость, в часах			54	18	36	90	18	27	9	36	Промежуточная аттестация		
												Форма	Семестр	
												Зачёт		
												Экзамен	1	

4.2. Содержание дисциплины

Раздел 1. Новые средства языка C++

Тема 1.1. Новые средства языка C++

Автоматическое выведение типа с помощью ключевого слова `auto`. Универсальная инициализация и списки инициализации. Диапазонные циклы `for`. Семантика перемещений и `rvalue`-ссылки. Новые строковые литералы. Новые возможности шаблонов. Лямбда-выражения и лямбда-функции. Новый синтаксис объявления функций. Перечисления с ограниченной областью видимости. Новые фундаментальные типы данных.

Лабораторная работа 1. Изменения объявлений и определений объектов, введённые в новый стандарт языка C++

Лабораторная работа 2. Обработка строк в новом стандарте языка C++

Раздел 2. Стандартная библиотека шаблонов

Тема 2.1. Стандартная библиотека шаблонов

Компоненты библиотеки STL. Контейнеры. Итераторы. Алгоритмы. Адаптеры итераторов. Пользовательские обобщённые функции. Функциональные объекты. Элементы контейнеров. Ошибки и исключения в библиотеке STL.

Раздел 3. Контейнеры STL

Тема 3.1. Последовательные контейнеры

Общие возможности и операции над контейнерами. Массивы: возможности массивов, операции над массивами, обработка исключений. Векторы: возможности векторов, операции над векторами, использование векторов в качестве массивов языка C, обработка исключений. Деки: возможности деков, операции над деками, обработка исключений. Списки: возможности списков, операции над списками, обработка исключений. Последовательные списки: возможности последовательных списков, операции над последовательными списками, обработка исключений.

Тема 3.2. Ассоциативные контейнеры

Множества и мультимножества: возможности множеств и мультимножеств, операции над множествами и мультимножествами, обработка исключений. Отображения и мультиотображения: возможности отображений и мультиотображений, операции над отображениями и мультиотображениями, использование отображений как ассоциативных массивов, обработка исключений.

Тема 3.3. Неупорядоченные контейнеры

Возможности неупорядоченных контейнеров. Создание неупорядоченных контейнеров и управления ими. Другие операции над неупорядоченными контейнерами. Использование неупорядоченных отображений в качестве ассоциативных массивов. Обработка исключений

Лабораторная работа 3. Обработка векторов в новом стандарте языка C++

Лабораторная работа 4. Обработка списков в новом стандарте языка C++

Лабораторная работа 5. Обработка множеств в новом стандарте языка C++

Лабораторная работа 6. Обработка отображений в новом стандарте языка C++

Раздел 4. Итераторы STL

Тема 4.1. Итераторы STL

Категории итераторов: итераторы вывода, итераторы ввода, однонаправленные итераторы, двунаправленные итераторы, итераторы произвольного доступа, проблема инкремента и декремента итераторов вектора. Вспомогательные функции для работы с итераторами. Адаптеры итераторов. Свойства итераторов.

Раздел 5. Алгоритмы STL

Тема 5.1. Немодифицирующие алгоритмы

Обзор алгоритмов. Вспомогательные алгоритмы. Алгоритм `for_each()`. Алгоритм подсчёта элементов. Алгоритм нахождения максимума и минимума. Алгоритм поиска элементов. Алгоритм сравнения диапазонов. Предикаты для диапазонов.

Тема 5.1. Модифицирующие алгоритмы

Алгоритм копирования элементов. Алгоритм перемещение элементов. Алгоритм преобразования и объединение элементов. Алгоритм замены элементов. Алгоритмы удаления. Перестановочные алгоритмы. Алгоритмы сортировки.

Лабораторная работа 7. Применение обобщённых алгоритмов в новом стандарте языка C++

Раздел 6. Функциональные объекты STL и лямбда-функции

Тема 6.1. Функциональные объекты STL и лямбда-функции

Понятие функционального объекта. Функциональные объекты как критерий сортировки. Функциональные объекты, имеющие внутреннее состояние. Предикаты и функциональные объекты. Стандартные функциональные объекты. Функциональные адаптеры и привязки. Пользовательские функциональные объекты для функциональных адаптеров. Лямбда-функции и адаптеры. Лямбда-функции и функциональные объекты, имеющие состояние. Лямбда-функции, вызывающие глобальные функции и функции-члены. Лямбда-функция как критерий сортировки, функции-хеширования и функции-члена.

Лабораторная работа 8. Применение функциональных объектов для обработки коллекций в обобщённых алгоритмах

Лабораторная работа 9. Применение лямбда-выражений для обработки коллекций в обобщённых алгоритмах

5. Образовательные технологии

В ходе освоения дисциплины «Современное программирование» при проведении аудиторных занятий используется образовательная технология, предусматривающая такие методы и формы изучения материала как лекция, лабораторное занятие, включающие в том числе активные и интерактивные формы занятий:

- проведение интерактивной лекции (Тема 2.1. «Стандартная библиотека шаблонов»);
- проведение лекции проблемного характера (Тема 6.1. «Функциональные объекты STL и лямбда-функции»);
- проведение работы в парах (Лабораторная работа 7. «Применение обобщённых алгоритмов в новом стандарте языка C++»).

Занятия, проводимые в интерактивной форме составляют 25 % от общего количества

аудиторных занятий.

Самостоятельная работа студентов подразумевает работу под руководством преподавателя (консультации, помощь в написании и отладки программ и др.) и индивидуальную работу студента, выполняемую в том числе в компьютерном классе с выходом в Интернет на факультете вычислительной техники университета.

При реализации образовательных технологий используются следующие виды самостоятельной работы:

- работа с конспектом лекции (обработка текста);
- подготовка к лабораторной работе;
- выполнение индивидуального задания по лабораторным работам;
- поиск информации в сети «Интернет» и литературе;
- подготовка к сдаче экзамена.

6. Учебно-методическое обеспечение самостоятельной работы студентов.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

6.1. План самостоятельной работы студентов

№ нед.	Тема	Вид самостоятельной работы	Задание	Рекомендуемая литература	Количество часов
1	2	3	4	5	6
1-2	1.1	<i>Подготовка к аудиторному занятию</i>	Изучить правила выполнения универсальной инициализации, формирование списков инициализации, новый синтаксис объявления функций, новые фундаментальные типы	1,2,3,4,5,6,7	2
	1.2	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть диапазонный цикл <code>for</code> , автоматическое выведение типа с помощью ключевого слова <code>auto</code> . Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4,5	3 1
3-4	2.1	<i>Подготовка к аудиторному занятию</i>	Изучить основные компоненты стандартной библиотеки шаблонов	3,4,5,6,7	2
	1.3	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть класс <code>string</code> , набор операций над элементами класса. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4,5,7	3 1
5-6	3.1	<i>Подготовка к аудиторному занятию</i>	Изучить контейнеры <code>vector</code> , <code>deque</code> , <code>array</code> , <code>list</code> , <code>forward_list</code> , набор операций, которые допустимо выполнять над элементами этих контейнеров.	3,4	2

	3.4	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть контейнер vector, возможности контейнера, набор операций над элементами этого контейнера. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4	3 1
7-8	3.2	<i>Подготовка к аудиторному занятию</i>	Изучить контейнеры set, multiset, map, multimap, набор операций, которые допустимо выполнять над элементами этих контейнеров.	3,4	2
	3.5	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть контейнер list, возможности контейнера, набор операций над элементами этого контейнера. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4	3 1
9-10	3.3	<i>Подготовка к аудиторному занятию</i>	Изучить контейнеры hash_set, hash_multiset, hash_map, hash_multimap, набор операций, которые допустимо выполнять над элементами этих контейнеров.	3,4	2
	3.6	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть контейнеры set, multiset, возможности контейнеров, набор операций над элементами этих контейнеров. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4	3 1
11-12	4.1	<i>Подготовка к аудиторному занятию</i>	Изучить категории итераторов, вспомогательные функции для работы с итераторами, адаптеры итераторов.	3,4,5,7	2
	3.7	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть контейнеры map, multimap, возможности контейнеров, набор операций над элементами этих контейнеров. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	3,4	3 1
13-14	5.1	<i>Подготовка к аудиторному занятию</i>	Изучить алгоритмы подсчета элементов, нахождения максимума и минимума, поиска элементов, сравнения диапазонов.	3,4	2
	5.3	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть алгоритм for_each(), алгоритмы подсчета элементов, сравнения диапазонов, поиска элементов, копирования элементов, сортировки.	3,4	3

			Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе		1
15-16	5.2	<i>Подготовка к аудиторному занятию</i>	Изучить алгоритмы копирования элементов, перемещение элементов, преобразования и объединение элементов, замены элементов, удаления, алгоритмы сортировки.	3,4	2
	6.2	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть структуру функционального объекта, его внутреннее состояние, передача параметров функциональному объекту. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	3,4,5	3 1
17-18	6.1	<i>Подготовка к аудиторному занятию</i>	Изучить концепцию функционального объекта, стандартные функциональные объекты, концепцию лямбда-функции, правила вызова глобальных функций из лямбда-функций, использование лямбда-функций как критерий сортировки, функции-хеширования и функции-члена	3,4,5	2
	6.3	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть структуру лямбда-функции, её внутреннее состояние, правила вызова глобальных функций из лямбда-функций. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	3,4,5	3 1

6.2. Методические указания по организации самостоятельной работы студентов

Планируются следующие виды самостоятельной работы:

- подготовка к лабораторным работам занятиям,
- выполнение индивидуального задания,
- оформление отчётов по лабораторным работам,
- подготовка к экзамену,
- работа с конспектом лекций и изучение литературы при подготовке к экзаменам.

6.3. Материалы для проведения текущего и промежуточного контроля знаний студентов

Контроль освоения компетенций

№ п\п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
1	Текущий: проверка выполнения индивидуального задания на лабораторной работе	Разделы 1 – 6	ОПК-4; ПСК-1,2
2	Текущий: собеседование при защите лабораторных работ	Разделы 1 – 6	ОПК-4; ПСК-1,2
3	Промежуточный: экзамен (2 вопроса)	Разделы 1 – 6	ОПК-4; ПСК-1,2

Перечень примерных вопросов к экзамену

1. Инициализация переменных и объектов. Списки инициализации
2. Диапазонный цикл for
3. Rvalue-ссылки
4. Вариативные шаблоны
5. Лямбда-выражение и лямбда-функция
6. Основные компоненты библиотеки STL
7. Контейнеры библиотеки STL
8. Использование контейнеров vector, deque и array
9. Списки в библиотеке STL
10. Однонаправленные списки в библиотеке STL
11. Ассоциативные контейнеры в библиотеке STL
12. Неупорядоченные контейнеры в библиотеке STL
13. Ассоциативные массивы
14. Итераторы. Операции над итераторами
15. Категории итераторов
16. Алгоритмы поиска
17. Алгоритмы обработки диапазонов коллекций
18. Модифицирующие алгоритмы
19. Использование функций в качестве аргументов алгоритма
20. Использование лямбда-выражений
21. Понятие функционального объекта
22. Стандартные функциональные объекты

7. Учебно-методическое и информационное обеспечение дисциплины «Современное программирование»

а) основная литература:

1. Довбуш Г. Ф. Visual C++ на примерах / Г. Ф. Довбуш, А. Д. Хомоненко ; под ред. А. Д. Хомоненко. — СПб.: БХВ-Петербург, 2008. (60 экземпляров)
2. Шибанов С.В. Основы программирования на языке C++ : учебное пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2008. (125 экземпляров)

б) дополнительная литература:

3. Страуструп Б. Программирование: принципы и практика использования C++. : Пер. с англ. – М.: Издательский дом «Вильямс», 2011.
4. Липман С., Лажойе Ж, Му В. Язык программирования C++. Базовый курс, 5-е изд. : Пер с англ. – М.: Издательский дом «Вильямс», 2014.
5. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2012.
6. Дьюхерст С. C++. Священные знания. – Пер. с англ. – СПб.: Символ-Плюс, 2012.
7. Павловская, Т.А. C\C++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: Питер, 2012.

в) программное обеспечение

Все работы выполняются на персональных компьютерах под управлением ОС Windows в среде Visual Studio на языке C++.

и Интернет-ресурсы http://algotlist.manual.ru/math/teornum/count_sys.php,
http://comp-science.narod.ru/Progr/Syst_Sch.html, <http://th-algoritmov.narod.ru/3.htm>.

8. Материально-техническое обеспечение дисциплины

В целях оптимизации учебного процесса студенты используют рабочие места в компьютерном классе, оборудованном локальной сетью и выходом в Internet, имеющиеся в библиотеке учебники. Все лабораторные работы выполняются на персональных компьютерах.

Рабочая программа дисциплины «Современное программирование» составлена в соответствии с требованиями ФГОС ВПО с учетом рекомендаций ПрООП по направлению подготовки 01.04.02 «Прикладная математика и информатика»

Программу составил:

1. Абрамов И.А., доцент кафедры КТ



Настоящая программа не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.

Программа одобрена на заседании кафедры «Компьютерные технологии»

Протокол № 2

от « 16 » сентября 2015 года

Зав. кафедрой «Компьютерные технологии» _____



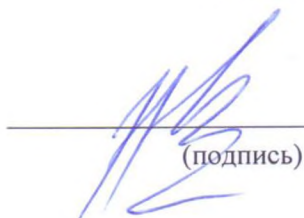
В. И. Горбаченко

Программа одобрена методической комиссией факультета вычислительной техники

Протокол № 2

от « 15 » октября 2015 года

Председатель методической комиссии
Факультета вычислительной техники



(подпись)

Н.Н. Кончов

(Ф.И.О.)

Сведения о переутверждении программы на очередной учебный год и регистрации изменений

Учебный год	Решение кафедры (№ протокола, дата, подпись зав. кафедрой)	Внесенные изменения	Номера листов		
			замененных	новых	аннулированных
2016/2017	№1 от 29.08.2016 <i>[подпись]</i>	уменьшен п.5	9	—	—
2017/2018	№1 от 30.08.2017 <i>[подпись]</i>	без изменений	—	—	—
2018/2019	№1 от 31.08.2018 <i>[подпись]</i>	без изменений	—	—	—