

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

УТВЕРЖДАЮ
Декан факультета
Фионова Л. Р.
« 03 » _____ 2019 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

М1.В.01. Современное программирование

Направление подготовки 01.04.02. Прикладная математика и информатика

Направленность(магистерская программа) Математическое и программное обеспечение вычислительных машин

Квалификация выпускника – *магистр*

Форма обучения очная

Пенза, 2019

1. Цели освоения дисциплины

Формируемые дисциплиной «Современное программирование» знания и умения готовят выпускника данной образовательной программы к выполнению следующих обобщенных трудовых функций (трудовых функций):

- «Оценка возможности создания архитектурного проекта программного средства» (ПС 06.003 «Архитектор программного обеспечения», ТФ Н/01.6);
- «Выбор технологий и средств разработки программного обеспечения, включая системы управления исходным кодом» ((ПС 06.003 «Архитектор программного обеспечения», ТФ I/05.6).

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Современное программирование» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули) ОПОП».

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 дисциплин: «Разработка параллельного и распределенного программного обеспечения», «Разработка программного обеспечения мобильных систем», «Разработка приложений научной визуализации», «Разработка приложений визуализации трехмерных сцен и объектов», последующего прохождения практик: Учебной (проектной (проектно-технологической)) практики, Производственной (преддипломной) практики, Производственной практики (научно-исследовательской работы), выполнению и защите выпускной квалификационной работы.

3. Результаты освоения дисциплины «Современное программирование»

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

Коды компет енции	Наименование компетенции	Индикатор достижения компетенции	В результате освоения дисциплины обучающийся должен:
ПК-2	<i>Способен разрабатывать и применять математические методы, системное и прикладное программное обеспечение для решения задач производственно-технологической деятельности с учетом требований информационной безопасности</i>	<i>ПК-2.3. Разрабатывает и применяет методы, системное и прикладное программное обеспечение для решения профессиональных задач с учетом требований информационной безопасности</i>	Знать принципы современного программирования и его применения для решения профессиональных задач: Уметь применять современное программирование, математические методы, системное и прикладное программное обеспечение для решения задач производственно-технологической деятельности; Владеть: практическими навыками использования современного программирования для решения профессиональных

			задач с учётом требований безопасности
--	--	--	----------------------------------------

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)		
				Контактная работа				Самостоятельная работа					Опрос на лабораторном занятии	Проверка выполнения индивидуального задания	Проверка отчёта выполнения лабораторной
				Всего	Лекция	Лабораторные занятия	Др. виды контакт. работы	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	16
1.	Раздел 1. Новые средства языка C++	1	1-4	8	2	6		10	2	6	2				
1.1.	Тема 1.1. Новые средства языка C++		1	2	2	2		2	2						
1.2.	Лабораторная работа 1. Изменения объявлений и определений объектов, введённые в новый стандарт языка C++		1	2				4		3	1		1	2	2
1.3.	Лабораторная работа 2. Обработка строк в новом стандарте языка C++		2-3	4		4		4		3	1		2	3	4
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++		4-5	4		4	4		3	1		4	5	6
3.5.	Лабораторная работа 4. Обработка списков в новом стандарте языка C++		6-7	4		4	4		3	1		6	7	8
3.6.	Лабораторная работа 5. Обработка множеств в новом стандарте языка C++		8-9	4		4	4		3	1		8	9	10
3.7.	Лабораторная работа 6. Обработка отображений в новом стандарте языка C++		10-11	4		4	4		3	1		10	11	12
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++		12-13	4		4	4		3	1		12	13	14
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	1		2	2						
6.2	Лабораторная работа 8. Применение функциональных объектов для обработки коллекций в обобщённых алгоритмах		14-15	4		4	4		3	1		14	15	16

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		16-17	4		4	4		3	1		16	17	17
	Подготовка к экзамену			3		3	36				36			
	Общая трудоемкость, в часах			54	17	34	3	90	18	27	9	36	Промежуточная аттестация	
												Форма	Семестр	
												Зачёт		
												Экзамен	1	

4.2. Содержание дисциплины (модуля)

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.

В целях реализации индивидуального подхода к обучению студентов, осуществляющих учебный процесс по собственной траектории в рамках индивидуального рабочего плана, изучение данной дисциплины базируется на следующих возможностях: обеспечение внеаудиторной работы со студентами в том числе в электронной образовательной среде с использованием соответствующего программного оборудования, дистанционных форм обучения. возможностей интернет-ресурсов, индивидуальных консультаций и т.д.

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	2
	1.2	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть диапазонный цикл for, автоматическое выведение типа с помощью ключевого слова auto. Разработать программу согласно индивидуальному заданию Подготовить отчет по лабораторной работе	1,2,3,4	3 1
3-4	2.1	<i>Подготовка к аудиторному занятию</i>	Изучить основные компоненты стандартной библиотеки шаблонов	1,2,3,4	2
	1.3	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания</i>	Рассмотреть класс string, набор операций над элементами класса.	1,2,3,4	3

		<i>Оформление отчета по лабораторной работе</i>	Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе		1
5-6	3.1	<i>Подготовка к аудиторному занятию</i>	Изучить контейнеры vector, deque, array, list, forward_list, набор операций, которые допустимо выполнять над элементами этих контейнеров.	1,2,3,4	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 отчёт по лабораторной работе	1,2,3,4	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.	1,2,3,4	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 отчёт по лабораторной работе	1,2,3,4	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.	1,2,3,4	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 отчёт по лабораторной работе	1,2,3,4	3 1
11-12	4.1	<i>Подготовка к аудиторному занятию</i>	Изучить категории итераторов, вспомогательные функции для работы с итераторами, адаптеры итераторов.	1,2,3,4	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	1,2,3,4	3 1

			Подготовить отчёт по лабораторной работе		
13-14	5.1	<i>Подготовка к аудиторному занятию</i>	Изучить алгоритмы подсчёта элементов, нахождения максимума и минимума, поиска элементов, сравнения диапазонов.	1,2,3,4	2
	5.3	<i>Подготовка к лабораторной работе. Выполнение индивидуального задания Оформление отчета по лабораторной работе</i>	Рассмотреть алгоритм for_each(), алгоритмы подсчёта элементов, сравнения диапазонов, поиска элементов, копирования элементов, сортировки. Разработать программу согласно индивидуальному заданию Подготовить отчёт по лабораторной работе	1,2,3,4	3 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.	1,2,3,4	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	1,2,3,4	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	1,2,3,4	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	1,2,3,4	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.

Подготовка к лабораторным занятиям проводится посредством изучения курса лекций, дополнительной литературы, Интернет-ресурсов.

Задание к выполнению каждой лабораторной работы состоит с общей части, которая сформулирована в разделе "Задание к выполнению" и уточнения варианта, который приведён в разделе "Варианты заданий". Студент должен заранее ознакомиться со своим заданием и, если у него возникают какие-либо вопросы относительно задания, поставить эти вопросы преподавателю до начала работы.

Выполнение начинается с разработки алгоритма решения задачи. На этом этапе должны быть детально проанализированы условия задания. Хотя представление схемы алгоритма не является обязательным элементом проектирования программы, но схема является удобным инструментом для осмысления задачи и оптимизации решения. При разработке алгоритма следует уделять внимание его упрощению, минимизации объёма вычислений, удалению лишних операций и т.п.

Написание текста программы начинается с определения переменных, которые необходимы для функционирования алгоритма. Большая часть переменных может быть определена ещё на этапе проектирования схемы алгоритма.

Для каждой переменной необходимо определить её тип и тщательно проверить, удовлетворяет ли диапазон значений выбранного типа тем значениям, которые может реально принимать переменная. Для массивов и символьных строк следует убедиться, что их размерность соответствует возможным размерам агрегаций данных.

Если схема алгоритма сделана достаточно тщательно, написание кодовой части программы сводится к записи каждого элемента схемы оператором языка программирования. Если в составе оператора встречается обращение к функции, следует проверить соответствие состава, последовательности и типов параметров, и возвращаемого значения спецификациям функции. Одновременно необходимо убедиться, что был включён файл-заголовок с описанием этой функции в программу.

Подготовленный текст программы следует набрать в текстовом редакторе и сохранить его в файле, который затем необходимо скомпилировать, чтобы получить исполняемый файл.

Отчёт к лабораторной работе должен содержать:

- Тему работы
- Цель работы
- Задание для выполнения, включая индивидуальное задание
- Описание алгоритма программы
- Описание переменных и структур данных, которые применяются в программе
- Описание ключевых программных решений, принятых при реализации алгоритма в тексте программы
- Текст программы
- Результат работы программы
- Выводы

Подготовка к экзамену проводится посредством изучения курса лекций, изучения дополнительной литературы, Интернет-ресурсов.

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	ПК-2
2	Текущий: собеседование при защите лабораторных работ	Разделы 1 – 6	ПК-2
3	Промежуточный: экзамен (2 вопроса)	Разделы 1 – 6	ПК-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 материально-техническое обеспечение дисциплины «Современное программирование»

а) учебная 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) http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=8109
2. Шибанов С.В. Основы программирования на языке C++ : учебное пособие. – Пенза: Изд-во Пенз. гос. ун-та, 2008. (125 экземпляров) <http://kleopatra.pnzgu.ru/cgi->

bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=9655

3. Синицин С.В., Михайлов А.С., Хлытчиев О. И. Программирование на языке высокого уровня: Учебник. – М.: Академия, 2010. — 400 с. (экземпляров неограниченно) ЭБС Академия <http://www.academia-moscow.ru/catalogue/4831/38050/>
4. Павловская, Т.А. С\C++. Программирование на языке высокого уровня / Т.А. Павловская. – СПб.: Питер, 2012. (5 экземпляров) http://kleopatra.pnzgu.ru/cgi-bin/irbis64r_91/cgiirbis_64.exe?P21DBN=KATL&I21DBN=KATL_PRINT&S21FMT=fullw_print&C21COM=F&Z21MFN=14891

б) Интернет-ресурсы http://algolist.manual.ru/maths/teornum/count_sys.php,
http://comp-science.narod.ru/Progr/Syst_Sch.html, <http://th-algoritmov.narod.ru/3.htm>.

в) Программное обеспечение:

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

г) Другое материально-техническое обеспечение

При проведении лабораторных занятий используются 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 ВО по направлению подготовки 01.04.02 «Прикладная математика и информатика», утвержденного приказом Министерства образования и науки Российской Федерации от «10» января 2019 г. № 13.

Программу составили:


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 № 12 от «26» июня 2019 года

Зав. кафедрой «Компьютерные технологии» _____  В. И. Горбаченко

Программа одобрена методической комиссией факультета вычислительной техники

Протокол № 10 от «03» июня 2019 года

Председатель методической комиссии
факультета вычислительной техники

_____  Т. В. Глотова
(подпись) (Ф.И.О.)