МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

Уровень основной образовательной программы: **Бакалавриат**
Направление подготовки (специальность): **09.03.02 Информационные системы и технологии**
Профиль: **Информационные системы и технологии**
Форма обучения: **очная**
Факультет: **ФВС, Факультет вычислительных систем**
Кафедра: **ЭМИС, Кафедра экономической математики, информатики и статистики**
Курс: **2**
Семестр: **3**
Учебный план набора 2014 года

Распределение рабочего времени

№	Виды учебной деятельности	3 семестр	Всего	Единицы
1	Лекции	36	36	часов
2	Практические занятия	36	36	часов
3	Лабораторные занятия	36	36	часов
4	Всего аудиторных занятий	108	108	часов
5	Самостоятельная работа	72	72	часов
6	Всего (без экзамена)	180	180	часов
7	Подготовка и сдача экзамена	36	36	часов
8	Общая трудоемкость	216	216	часов
		6	6	З.Е

Экзамен: 3 семестр

Томск 2016

Согласована на портале № 1396

Рассмотрена и одобрена на заседании кафедры
протокол № 12 от «30» 6 2016 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований Федерального Государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.02 Информационные системы и технологии, утвержденного 2015-03-12 года, рассмотрена и утверждена на заседании кафедры «___» _____ 20__, протокол №_____.

Разработчики:

доцент каф. ЭМИС _____ Шельмина Е. А.

Заведующий обеспечивающей каф. ЭМИС _____ Боровской И. Г.

Рабочая программа согласована с факультетом, профилирующей и выпускающей кафедрами направления подготовки (специальности).

Декан ФВС _____ Козлова Л. А.

Заведующий профилирующей каф. ЭМИС _____ Боровской И. Г.

Заведующий выпускающей каф. ЭМИС _____ Боровской И. Г.

Эксперты:

профессор каф. ЭМИС _____ Колесникова С. И.

Согласована на портале № 1396

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Целью данной учебной дисциплины является обучение программированию на языке C++ с использованием принципов объектно-ориентированного программирования.

1.2. Задачи дисциплины

- Изучение основ теории объектно-ориентированного программирования, в том числе:;
- пути эволюции технологий программирования от алгоритмического к ООП;;
- основных принципов объектно-ориентированного построения программных систем;;
- понятий классов, объектов и взаимоотношений между ними;;
- изучение средств объектно-ориентированного программирования языка C++, средств стандартной библиотеки STL.;

2. Место дисциплины в структуре ОПОП

Дисциплина «Объектно-ориентированное программирование» (Б1. Дисциплины (модули)) Б1. Дисциплины (модули) профессионального цикла обязательных дисциплин.

Предшествующими дисциплинами, формирующими начальные знания, являются следующие дисциплины: Инновационные технологии, Программирование на языках высокого уровня.

Последующими дисциплинами являются: Языковые средства создания гипердокументов.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- ОПК-1 владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий;
- ОПК-5 способностью использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению;
- ПК-12 способностью разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные);

В результате изучения дисциплины студент должен:

- **знать** о общие направления в изучении информационных систем и технологий объектно-ориентированного программирования; современные компьютерные технологии поиска и анализа информации в области объектно-ориентированного программирования; основные принципы разработки средств реализации информационных технологий с использованием языка C++; принципы объектно-ориентированного программирования; основные формы наследования; способы реализации полиморфизма в языке C++;
- **уметь** применять базовые знания при решении задач в области объектно-ориентированного программирования; осуществлять оптимальный поиск необходимой информации для обоснования принятых идей при разработке программ на объектно-ориентированном языке C++; применять способы реализации информационных технологий в области объектно-ориентированного программирования; создавать классы на C++ и использовать их; осуществлять перегрузку операторов в C++; создавать иерархию классов на C++; использовать полиморфизм; использовать библиотеку потоков; использовать стандартную библиотеку шаблонов STL;
- **владеть** базовыми знаниями для решения практических задач в области объектно-ориентированного программирования; передовыми технологиями комплексного анализа поисковой информации при принятии аргументированных решений в области объектно-ориентированного программирования; методами разработки средств реализации информационных технологий с использованием языка программирования C++; методами и технологиями разработки программных продуктов, использующих представление отдельных элементов предметной области в виде самостоятельных объектов пользовательских классов.

Согласована на портале № 1396

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 6 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

№	Виды учебной деятельности	3 семестр	Всего	Единицы
1	Лекции	36	36	часов
2	Практические занятия	36	36	часов
3	Лабораторные занятия	36	36	часов
4	Всего аудиторных занятий	108	108	часов
5	Самостоятельная работа	72	72	часов
6	Всего (без экзамена)	180	180	часов
7	Подготовка и сдача экзамена	36	36	часов
8	Общая трудоемкость	216	216	часов
		6	6	З.Е

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

№	Названия разделов дисциплины	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа	Всего часов (без экзамена)	Формируемые компетенции
1	Введение в объектно-ориентированное программирование и язык С++	2	0	0	1	3	ОПК-1
2	Классы	4	4	4	9	21	ОПК-1, ПК-12
3	Указатель this	4	2	0	3	9	ОПК-1
4	Динамическое распределение памяти при работе с классами	4	4	6	11	25	ПК-12
5	Обработка исключительных ситуаций	2	4	4	7	17	ОПК-5, ПК-12
6	Основные конструкции языка С++	2	0	0	1	3	ОПК-1, ОПК-5
7	Функции языка С++	4	4	2	5	15	ОПК-5, ПК-12
8	Перегрузка операций и дружественные классы	4	4	8	9	25	ОПК-5, ПК-12
9	Наследование	2	4	4	7	17	ОПК-1, ОПК-5

Согласована на портале № 1396

10	Виртуальные функции и классы	4	4	2	7	17	ОПК-5, ПК-12
11	Объекты абстрактных типов	2	2	0	3	7	ПК-12
12	Шаблоны классов	2	4	6	9	21	ОПК-1, ОПК-5, ПК-12
	Итого	36	36	36	72	180	

5.2. Содержание разделов дисциплины (по лекциям)

Содержание разделов дисциплин (по лекциям) приведено в таблице 5.2.

Таблица 5.2 - Содержание разделов дисциплин (по лекциям)

Названия разделов	Содержание разделов дисциплины по лекциям	Трудоемкость, ч	Формируемые компетенции
3 семестр			
1 Введение в объектно-ориентированное программирование и язык C++	Философия программирования, заложенная в языке C++, и ее место в развитии языков программирования. Методика создания программ. Имена, их область существования и видимости. Атрибуты компоновки имен. Структура программы на языке C++.	2	ОПК-1
	Итого	2	
2 Классы	Объекты и классы. Процедурное и объектно-ориентированное программирование. Понятие класса. Определение и реализация класса. Деструкторы и конструкторы классов. Члены-данные и члены-методы классов. Уровни доступа класса.	4	ОПК-1, ПК-12
	Итого	4	
3 Указатель this	Создание и использование объектов класса. Работа с указателем this. Создание массивов объектов. Абстрактные типы данных.	4	ОПК-1
	Итого	4	
4 Динамическое распределение памяти при работе с классами	Классы и динамическое распределение памяти. Конструкторы копирования и перегруженные операторы присваивания. Конструирование объекта в динамической памяти. Статические члены класса. Использование указателей на объекты.	4	ПК-12
	Итого	4	
5 Обработка исключительных	Обработка исключительных ситуаций.	2	ОПК-5,

Согласована на портале № 1396

ситуаций	Итого	2	ПК-12
6 Основные конструкции языка С++	Представление данных. Имена типов данных. Встроенные и составные типы данных. Квалификатор const. Циклы и выражения сравнения. Операторы ветвления и логические операции.	2	ОПК-1, ОПК-5
	Итого	2	
7 Функции языка С++	Функции языка С++ , их отличия от функций языка С. Рекурсивный вызов функции. Тип «указатель на функцию». Ссылочные переменные. Аргументы, принимаемые по умолчанию. Перегрузка функций. Шаблоны функций.	4	ОПК-5, ПК-12
	Итого	4	
8 Перегрузка операций и дружественные классы	Работа с классами. Перегрузка операций. Использование дружественных классов. Автоматическое преобразование и приведение типов для классов.	4	ОПК-5, ПК-12
	Итого	4	
9 Наследование	Наследование классов. Защита доступа при наследовании. Преобразование типов вверх и вниз.	2	ОПК-1, ОПК-5
	Итого	2	
10 Виртуальные функции и классы	Виртуальные функции. Статическое и динамическое связывание. Реализация виртуальных функций. Абстрактные базовые классы.	4	ОПК-5, ПК-12
	Итого	4	
11 Объекты абстрактных типов	Повторное использование программного кода в С++. Классы, содержащие объекты абстрактных типов в качестве элементов. Виртуальные базовые классы.	2	ПК-12
	Итого	2	
12 Шаблоны классов	Создание шаблонов классов. Использование шаблонов классов.	2	ОПК-1, ОПК-5, ПК-12
	Итого	2	
Итого за семестр		36	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представ-лены в таблице 5.3.

Таблица 5.3 - Разделы дисциплины и междисциплинарные связи

№	Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин

Согласована на портале № 1396

		1	2	3	4	5	6	7	8	9	10	11	12
Предшествующие дисциплины													
1	Инновационные технологии	+						+					
2	Программирование на языках высокого уровня						+	+					
Последующие дисциплины													
1	Языковые средства создания гипердокументов		+										

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4

Таблица 5. 4 – Соответствие компетенций и видов занятий, формируемых при изучении дисциплины

Компетенции	Виды занятий			
	Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа
ОПК-1	+	+	+	+
ОПК-5	+	+	+	+
ПК-12	+	+	+	+

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП

7. Лабораторный практикум

Содержание лабораторных работ приведено в таблице 7.1.

Таблица 7. 1 – Содержание лабораторных работ

Названия разделов	Содержание лабораторных работ	Трудоемкость, ч	Формируемые компетенции
3 семестр			
2 Классы	Выполнение индивидуального задания на тему «Разработка простейшего пользовательского типа данных».	4	ОПК-1, ПК-12
	Итого	4	
4 Динамическое распределение памяти при работе с классами	Создание классов для работы с динамическими данными переменной	6	ПК-12

Согласована на портале № 1396

	размерности.		
	Итого	6	
5 Обработка исключительных ситуаций	Выполнение индивидуального задания на тему «Обработка исключительных ситуаций».	4	ОПК-5, ПК-12
	Итого	4	
7 Функции языка С++	Изучение интегрированной среды разработки Visual Studio 8.0. Использование объектов типа «указатель на функцию» для обеспечения полиморфного поведения программы.	2	ОПК-5, ПК-12
	Итого	2	
8 Перегрузка операций и дружественные классы	Выполнение индивидуального задания на тему «Создание класса с перегрузкой некоторых операций».	8	ОПК-5, ПК-12
	Итого	8	
9 Наследование	Выполнение индивидуального задания на тему «Наследование классов».	4	ОПК-1, ОПК-5
	Итого	4	
10 Виртуальные функции и классы	Выполнение индивидуального задания на тему «Виртуальные функции».	2	ОПК-5, ПК-12
	Итого	2	
12 Шаблоны классов	Создание шаблона класса для описания обобщенного массива.	6	ОПК-1, ОПК-5, ПК-12
	Итого	6	
Итого за семестр		36	

8. Практические занятия

Содержание практических работ приведено в таблице 8.1.

Таблица 8. 1 – Содержание практических работ

Названия разделов	Содержание практических занятий	Трудоемкость, ч	Формируемые компетенции
3 семестр			
2 Классы	Разработка простейшего пользовательского типа данных	4	ОПК-1, ПК-12
	Итого	4	
3 Указатель this	Создание класса с использованием указателя this	2	ОПК-1
	Итого	2	
4 Динамическое распределение памяти при работе с классами	Создание классов для работы с динамическими данными переменной размерности	4	ПК-12

Согласована на портале № 1396

	Итого	4	
5 Обработка исключительных ситуаций	Обработка исключительных ситуаций	4	ОПК-5, ПК-12
	Итого	4	
7 Функции языка С++	Решение задач на тему «Функции языка С++»	4	ОПК-5, ПК-12
	Итого	4	
8 Перегрузка операций и дружественные классы	Создание класса с перегрузкой некоторых операций	4	ОПК-5, ПК-12
	Итого	4	
9 Наследование	Разработка иерархии классов	4	ОПК-1, ОПК-5
	Итого	4	
10 Виртуальные функции и классы	Разработка класса с использованием виртуальных функций	4	ОПК-5, ПК-12
	Итого	4	
11 Объекты абстрактных типов	Разработка класса, содержащего объекты абстрактных типов в качестве элементов	2	ПК-12
	Итого	2	
12 Шаблоны классов	Создание шаблона класса	4	ОПК-1, ОПК-5, ПК-12
	Итого	4	
Итого за семестр		36	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 - Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость ч	Формируемые компетенции	Формы контроля
3 семестр				
1 Введение в объектно-ориентированное программирование и язык С++	Проработка лекционного материала	1	ОПК-1	Конспект самоподготовки
	Итого	1		
2 Классы	Подготовка к практическим занятиям, семинарам	4	ОПК-1, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	4		
	Итого	9		

Согласована на портале № 1396

3 Указатель this	Подготовка к практическим занятиям, семинарам	2	ОПК-1	Тест, Конспект самоподготовки
	Проработка лекционного материала	1		
	Итого	3		
4 Динамическое распределение памяти при работе с классами	Подготовка к практическим занятиям, семинарам	4	ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	6		
	Итого	11		
5 Обработка исключительных ситуаций	Подготовка к практическим занятиям, семинарам	2	ОПК-5, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	4		
	Итого	7		
6 Основные конструкции языка C++	Проработка лекционного материала	1	ОПК-1, ОПК-5	Конспект самоподготовки
	Итого	1		
7 Функции языка C++	Подготовка к практическим занятиям, семинарам	2	ОПК-5, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	2		
	Итого	5		
8 Перегрузка операций и дружественные классы	Подготовка к практическим занятиям, семинарам	4	ОПК-5, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	4		
	Итого	9		
9 Наследование	Подготовка к практическим занятиям, семинарам	2	ОПК-1, ОПК-5	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного	1		

Согласована на портале № 1396

	материала			
	Оформление отчетов по лабораторным работам	4		
	Итого	7		
10 Виртуальные функции и классы	Подготовка к практическим занятиям, семинарам	4	ОПК-5, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	2		
	Итого	7		
11 Объекты абстрактных типов	Подготовка к практическим занятиям, семинарам	2	ПК-12	Тест, Конспект самоподготовки
	Проработка лекционного материала	1		
	Итого	3		
12 Шаблоны классов	Подготовка к практическим занятиям, семинарам	4	ОПК-1, ОПК-5, ПК-12	Тест, Конспект самоподготовки, Отчет по лабораторной работе
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	4		
	Итого	9		
Итого за семестр		72		
	Подготовка к экзамену	36		Экзамен
Итого		108		

10. Курсовая работа

Не предусмотрено РУП

11. Рейтинговая система для оценки успеваемости студентов

11.1. Балльные оценки для элементов контроля

Таблица 11.1 – Бальные оценки для элементов контроля

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
3 семестр				
Конспект самоподготовки	4	4	4	12
Отчет по лабораторной работе	10	10	10	30
Тест	10	8	10	28

Согласована на портале № 1396

Экзамен				30
Нарастающим итогом	24	46	70	100

11.2. Пересчет баллов в оценки за контрольные точки

Пересчет баллов в оценки за контрольные точки представлен в таблице 11.2.

Таблица 11. 2 – Пересчет баллов в оценки за контрольные точки

Баллы на дату контрольной точки	Оценка
≥ 90% от максимальной суммы баллов на дату КТ	5
От 70% до 89% от максимальной суммы баллов на дату КТ	4
От 60% до 69% от максимальной суммы баллов на дату КТ	3
< 60% от максимальной суммы баллов на дату КТ	2

11.3. Пересчет суммы баллов в традиционную и международную оценку

Пересчет суммы баллов в традиционную и международную оценку представлен в таблице 11.3.

Таблица 11. 3 – Пересчет суммы баллов в традиционную и международную оценку

Оценка (ГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично) (зачтено)	90 - 100	A (отлично)
4 (хорошо) (зачтено)	85 - 89	B (очень хорошо)
	75 - 84	C (хорошо)
	70 - 74	D (удовлетворительно)
3 (удовлетворительно) (зачтено)	65 - 69	
	60 - 64	E (посредственно)
2 (неудовлетворительно) (не зачтено)	Ниже 60 баллов	F (неудовлетворительно)

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Головин И.Г. Языки и методы программирования. - М. : Академия, 2012. - 304 с. (наличие в библиотеке ТУСУР - 30 экз.)

12.2. Дополнительная литература

1. Ашарина, И.В. Объектно-ориентированное программирование в C++: лекции и упражнения. [Электронный ресурс] : учебное пособие. — Электрон. дан. — М. : Горячая линия-Телеком, 2012. — 319 с. — Загл. с экрана. [Электронный ресурс]. - http://e.lanbook.com/books/element.php?pl1_id=5115

12.3. Учебно-методическое пособие и программное обеспечение

1. Объектно-ориентированное программирование: Методические указания по выполнению лабораторных и практических работ / Панасенко Е. А., Касимов В. З. - 2012. 19 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/1514, свободный.

2. Объектно-ориентированное программирование: Методические указания к самостоятельной работе студентов / Панасенко Е. А. - 2012. 6 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/1566, свободный.

12.4. Базы данных, информационно справочные и поисковые системы

1. Поисковая система google.ru

13. Материально-техническое обеспечение дисциплины

При выполнения практических заданий по дисциплине используются персональные ЭВМ с

Согласована на портале № 1396

процессорами Pentium 4, операционная система MS Windows XP.

14. Фонд оценочных средств

Фонд оценочных средств приведен в приложении 1.

15. Методические рекомендации по организации изучения дисциплины

Без рекомендаций.

Согласована на портале № 1396

Приложение 1

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____ П. Е. Троян
«___» _____ 20__ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Объектно-ориентированное программирование

Уровень основной образовательной программы: **Бакалавриат**
Направление подготовки (специальность): **09.03.02 Информационные системы и технологии**
Профиль: **Информационные системы и технологии**
Форма обучения: **очная**
Факультет: **ФВС, Факультет вычислительных систем**
Кафедра: **ЭМИС, Кафедра экономической математики, информатики и статистики**
Курс: **2**
Семестр: **3**

Учебный план набора 2014 года

Разработчики:

- доцент каф. ЭМИС Шельмина Е. А.

Экзамен: 3 семестр

Томск 2016

Согласована на портале № 1396

1. Введение

Фонд оценочных средств (ФОС) является приложением к рабочей программе дисциплины (практики) и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения студентом установленных результатов обучения.

ФОС по дисциплине (практике) используется при проведении текущего контроля успеваемости и промежуточной аттестации студентов.

Перечень закрепленных за дисциплиной (практикой) компетенций приведен в таблице 1.

Таблица 1 – Перечень закрепленных за дисциплиной компетенций

Код	Формулировка компетенции	Этапы формирования компетенций
ПК-12	способностью разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	Должен знать о общие направления в изучении информационных систем и технологий объектно-ориентированного программирования; современные компьютерные технологии поиска и анализа информации в области объектно-ориентированного программирования; основные принципы разработки средств реализации информационных технологий с использованием языка C++; принципы объектно-ориентированного программирования; основные формы наследования; способы реализации полиморфизма в языке C++;; Должен уметь применять базовые знания при решении задач в области объектно-ориентированного программирования; осуществлять оптимальный поиск необходимой информации для обоснования принятых идей при разработке программ на объектно-ориентированном языке C++; применять способы реализации информационных технологий в области объектно-ориентированного программирования; создавать классы на C++ и использовать их; осуществлять перегрузку операторов в C++; создавать иерархию классов на C++; использовать полиморфизм; использовать библиотеку потоков; использовать стандартную библиотеку шаблонов STL;; Должен владеть базовыми знаниями для решения практических задач в области объектно-ориентированного программирования; передовыми технологиями комплексного анализа поисковой информации при принятии аргументированных решений в области объектно-ориентированного программирования; методами разработки средств реализации
ОПК-5	способностью использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	
ОПК-1	владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий	

Согласована на портале № 1396

		информационных технологий с использованием языка программирования C++; методами и технологиями разработки программных продуктов, использующих представление отдельных элементов предметной области в виде самостоятельных объектов пользовательских классов.;

Общие характеристики показателей и критериев оценивания компетенций на всех этапах приведены в таблице 2.

Таблица 2 – Общие характеристики показателей и критериев оценивания компетенций по этапам

Показатели и критерии	Знать	Уметь	Владеть
Отлично (высокий уровень)	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем	Контролирует работу, проводит оценку, совершенствует действия работы
Хорошо (базовый уровень)	Знает факты, принципы, процессы, общие понятия в пределах изучаемой области	Обладает диапазоном практических умений, требуемых для решения определенных проблем в области исследования	Берет ответственность за завершение задач в исследовании, приспосабливает свое поведение к обстоятельствам в решении проблем
Удовлетворительно (пороговый уровень)	Обладает базовыми общими знаниями	Обладает основными умениями, требуемыми для выполнения простых задач	Работает при прямом наблюдении

2 Реализация компетенций

2.1 Компетенция ПК-12

ПК-12: способностью разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные).

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания представлены в таблице 3.

Таблица 3 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	Основные принципы разработки средств реализации информационных технологий с использованием языка C++;	Применять способы реализации информационных технологий в области объектно-ориентированного программирования;	Методами разработки средств реализации информационных технологий с использованием языка программирования C++;
Виды занятий	• Практические занятия;	• Практические занятия;	• Лабораторные занятия;

Согласована на портале № 1396

	• Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Самостоятельная работа;
Используемые средства оценивания	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Экзамен;

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 4.

Таблица 4 – Показатели и критерии оценивания компетенции на этапах

Состав	Знать	Уметь	Владеть
Отлично (высокий уровень)	• принципы разработки программ с использованием языка программирования С++;; • способы реализации полиморфизма в языке С++;; • способы реализации шаблонов функций и классов в языке С++;; • правила использования обработчика исключительных ситуаций;;	• применять способы разработки программ на языке С++;; • разрабатывать шаблоны функций и классов в С++;; • разрабатывать программы с использованием обработчика исключительных ситуаций;; • использовать стандартную библиотеку шаблонов STL; ;	• методами и технологиями разработки программных продуктов на С++ с использованием шаблонов функций и классов, обработчика исключительных ситуаций;;
Хорошо (базовый уровень)	• способы реализации шаблонов функций в языке С++;; • принципы разработки программ с использованием языка программирования С++;; • способы реализации полиморфизма в языке С++;;	• применять способы разработки программ на языке С++;; • разрабатывать шаблоны функций и классов в С++;; • использовать стандартную библиотеку шаблонов STL;;	• методами и технологиями разработки программных продуктов на С++ с использованием шаблонов функций при работе в команде;;
Удовлетворительно (пороговый уровень)	• принципы разработки программ с использованием языка программирования С++;; • способы реализации полиморфизма в языке	• применять способы разработки программ на языке С++;; • разрабатывать шаблоны функций и классов в С++;;	• методами и технологиями разработки программных продуктов на С++ с использованием шаблонов функций при

Согласована на портале № 1396

	C++;;		работе в команде и при прямом наблюдении;;

2.2 Компетенция ОПК-5

ОПК-5: способностью использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению.

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания представлены в таблице 5.

Таблица 5 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	Современные компьютерные технологии поиска и анализа информации в области объектно-ориентированного программирования;	Осуществлять оптимальный поиск необходимой информации для обоснования принятых идей при разработке программ на объектно-ориентированном языке C++;	Передовыми технологиями комплексного анализа поисковой информации при принятии аргументированных решений в области объектно-ориентированного программирования;
Виды занятий	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Лабораторные занятия; • Самостоятельная работа;
Используемые средства оценивания	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Экзамен;

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 6.

Таблица 6 – Показатели и критерии оценивания компетенции на этапах

Состав	Знать	Уметь	Владеть
Отлично (высокий уровень)	• современные компьютерные технологии поиска информации в области объектно-ориентированного программирования;; • компьютерные технологии анализа предметной области;;	• осуществлять поиск информации для разработки программных комплексов с использованием объектно-ориентированного языка программирования C+	• компьютерными технологиями анализа информации в области объектно-ориентированного программирования;; • инструментарием объектно-ориентированного программирования;;

Согласована на портале № 1396

	• основные принципы разработки программных комплексов на языке С++;; • определения класса и объекта, основные формы наследования, способы реализации полиморфизма в языке С++ с учетом проведенного анализа информации в области объектно-ориентированного программирования;;	+;; • анализировать предметную область;; • реализовывать оптимальный поиск информации для обоснования принятых идей при создании классов, иерархии классов, шаблонов функций и классов, использовании перегрузки функций и операторов в С++; ;	• способен разработать, отладить и протестировать программный продукт на языке С++;;
Хорошо (базовый уровень)	• современные компьютерные технологии поиска информации в области объектно-ориентированного программирования;; • компьютерные технологии анализа предметной области;; • основные принципы разработки программных комплексов на языке С++;; • определения класса и объекта, основные формы наследования в языке С++ с учетом проведенного анализа информации в области объектно-ориентированного программирования;;	• осуществлять поиск информации для разработки программных комплексов с использованием объектно-ориентированного языка программирования С++;; • анализировать предметную область;; • реализовывать оптимальный поиск информации для обоснования принятых идей при создании классов, иерархии классов, шаблонов функций, использовании перегрузки функций в С++;;	• компьютерными технологиями анализа информации в области объектно-ориентированного программирования;; • инструментарием объектно-ориентированного программирования;; • способен разработать, отладить и протестировать программный продукт на языке С++ при работе в команде;;
Удовлетворительно (пороговый уровень)	• современные компьютерные технологии поиска информации в области объектно-ориентированного программирования;; • компьютерные технологии анализа предметной области;; • основные принципы разработки	• осуществлять поиск информации для разработки программ на языке С++;; • анализировать предметную область;; • реализовывать оптимальный поиск информации для обоснования принятых идей при создании классов, шаблонов	• компьютерными технологиями анализа информации в области объектно-ориентированного программирования;; • способен разработать, отладить и протестировать программный продукт на языке С++ при работе в команде при

	программных комплексов на языке С++;;	функций, использовании перегрузки функций в С++;;	прямом наблюдении;;

2.3 Компетенция ОПК-1

ОПК-1: владением широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий.

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания представлены в таблице 7.

Таблица 7 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	Общие направления в изучении информационных систем и технологий объектно-ориентированного программирования;	Применять базовые знания при решении задач в области объектно-ориентированного программирования;	Базовыми знаниями для решения практических задач в области объектно-ориентированного программирования;
Виды занятий	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Лабораторные занятия; • Самостоятельная работа;
Используемые средства оценивания	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Конспект самоподготовки; • Тест; • Экзамен;	• Отчет по лабораторной работе; • Экзамен;

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 8.

Таблица 8 – Показатели и критерии оценивания компетенции на этапах

Состав	Знать	Уметь	Владеть
Отлично (высокий уровень)	• основные направления изучения информационных систем;; • основные определения объектно-ориентированного программирования;; • принципы объектно-ориентированного программирования;; • классификацию объектно-	• применять базовые знания для проведения анализа предметной области;; • применять методику создания программ с использованием объектно-ориентированного языка программирования С++ при решении профессиональных	• методикой самостоятельной разработки программ с использованием объектно-ориентированного языка программирования С++;;

Согласована на портале № 1396

	ориентированных языков программирования;; • методику создания программ с использованием объектно-ориентированного языка программирования С++;; • философию объектно-ориентированного программирования;;	задач;; • применять принципы объектно-ориентированного программирования при разработке программ на языке С++;;	
Хорошо (базовый уровень)	• основные определения объектно-ориентированного программирования;; • принципы объектно-ориентированного программирования;; • методику создания программ с использованием объектно-ориентированного языка программирования С++;; • философию объектно-ориентированного программирования;;	• применять базовые знания для проведения анализа предметной области;; • применять методику создания программ с использованием объектно-ориентированного языка программирования С++ при решении профессиональных задач;; • применять принципы объектно-ориентированного программирования при разработке программ на языке С++;;	• методикой разработки программ с использованием объектно-ориентированного языка программирования С++ при работе в команде;;
Удовлетворительно (пороговый уровень)	• основные определения объектно-ориентированного программирования;; • методику создания программ с использованием объектно-ориентированного языка программирования С++;; • философию объектно-ориентированного программирования;;	• применять базовые знания для проведения анализа предметной области;; • применять принципы объектно-ориентированного программирования при разработке программ на языке С++;;	• методикой разработки программ с использованием объектно-ориентированного языка программирования С++ при работе в команде при прямом наблюдении;;

Согласована на портале № 1396

3 Типовые контрольные задания

Для реализации вышеперечисленных задач обучения используются типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы, в следующем составе.

3.1 Вопросы на самоподготовку

– Дружественные структуры. Библиотека RTTI. Стандартная библиотека шаблонов. Ввод-вывод данных в C++ . Пространства имен.

3.2 Тестовые задания

– Тест №1 1. Объект в объектно-ориентированном программировании – это… a. элемент, необходимый для выполнения какой либо операции b. основная единица ООП, которая объединяет в себе как описывающие его свойства, так и действия c. процедура, которая начинает выполняться после реализации определенного события 2. Выберите наиболее подходящее определение класса. a. тип, содержащий набор функций b. тип, который отображает состояние некоторого объекта c. тип, описывающий поведение некоторой сущности d. тип, описывающий характеристики и поведение объекта 3. Как называется способность объекта скрывать свои данные и реализацию от других объектов системы? a. полиморфизм b. инкапсуляция c. абстракция d. наследование 4. Какая разница между идентичностью (identity) и равенством (equality) объектов в ООП? a. идентичность означает, что у объектов есть общий неабстрактный предок, а равенство - любой общий предок b. идентичность означает, что у объектов одинаковые поля, а равенство - что они содержат одинаковые данные c. идентичность означает, что объекты являются экземплярами одного и того же класса, а равенство - что они содержат одинаковые данные d. идентичность означает, что две ссылки указывают на один и тот же объект, а равенство - что они содержат одинаковые данные 5. Правильный вариант программы, выводящей "Hello World": a. #include int main() {cout<<"Hello World"; return 0;} b. #include int main() {cout>>"Hello World"; return 0;} c. #include void main() {cout<<"Hello World"; return 0;} 6. Отметьте свойства языка Си++, которые могут быть источниками возможных оши-бок программирования a. наличие встроенных типов данных b. возможность создания абстрактных классов c. наличие указателей d. возможность динамического распределения памяти e. возможность определения символических констант f. возможность преобразования типов 7. Что будет выведено в стандартный поток в результате выполнения программы int main() { int x1 = 5; x1 = x2 - 2; int x2 = x1; cout << x2; return 0;} a. -2 b. 5 c. 0 d. ошибка компиляции 8. Каково преимущество использования ключевого слова const вместо директивы #define? a. константу, определенную с помощью const, можно изменять во время рабо-ты b. к константе, определенной с помощью const, можно применить операции инкремента и декремента c. константа, определенная с помощью const, доступна в других модулях про-граммы d. константа, определенная с помощью const, имеет тип, и компилятор может проследить за ее использованием в соответствии с объявленным типом 9. Какое приведение типов используется в следующем выражении? int a = 0; float f = 3.4; f += (int)a; a. неявное приведение типов b. явное приведение типов c. стандартное приведение типов 10. Класс B наследован от класса A. Отметьте верное для класса B. a. объект класса B может использоваться как объект базового класса b. класс B должен быть определен с ключевым словом derived c. класс B может непосредственно обращаться к внутренним атрибутам базо-вого класса d. класс B наследует все операторы базового класса 11. Конструктор класса - это метод, который вызывается при создании объекта для .. a. выделения памяти под динамические атрибуты класса b. выделения памяти под статические атрибуты класса c. инициализации атрибутов объекта d. загрузки методов класса в память 12. Какими по умолчанию объявляются методы класса? a. private b. public c. protected d. по умолчанию не объявляются 13. В чем заключается принцип полиморфизма? a. в наличии виртуальных методов b. в наличии множественного наследования c. в использовании виртуального наследования 14. Какая из записей является правильной записью абстрактного класса? a. abstract class A { virtual int f() = 0; }; b. class A { virtual int f() = 0; }; c. class A { virtual int f(); }; 15. Что нужно сделать для освобождения памяти после выполнения такого кода ? char *a; a = new char[20]; a. delete [] a; b. delete a[]; c. delete a; 16. Какая операция позволяет получить значение, записанное по адресу, который со-держится в указателе? a. * b. ? c. ^ d. & 17. Операция "." обозначает a. что атрибут

Согласована на портале № 1396

объекта, следующий за этим оператором, будет изменен b. обращение к атрибуту объекта, используемое в специальных случаях c. обращение к атрибуту объекта 18. Каким будет результат следующей программы: int a = 5*3; float b = 1.5f; b += --a/2; cout << b; a. 8.0 b. 9.0 c. 8.5 d. 9.5 e. 7.5 19. Прототип функции задает a. тип функции, включая количество и тип аргументов и тип результата b. возможность выполнения этой функции из программ на других языках программирования c. имя функции и минимальное количество параметров 20. Как называется функция, которая вызывает саму себя? a. конструктором b. деструктором c. подставляемой d. рекурсивной 21. В чем разница между фактическими и формальными параметрами? a. формальные параметры могут использоваться только вне тела функции, а b. фактические - используются как вне функции, так и внутри ее c. формальные параметры определены в заголовке функции, а фактические - значения, с которыми функция вызывается d. нет различий 22. В каких выражениях используются унарные арифметические операции? a. c1 + d2 b. s2 % d % 2 c. –b 23. В программе на языке Си++ обязательно имеется функция a. head b. start c. prime d. main e. finish 24. Ключевое слово void обозначает что функция a. возвращает число с плавающей запятой b. возвращает целое число c. ничего не возвращает d. является главной 25. Укажите зарезервированное ключевое слово для динамического выделения памяти a. malloc b. new c. create d. value Тест №2 1. Какой будет результат следующего выражения? template T sum(T *a, T *b) { T f = 5; return (*a + *b) - f;} int main() {int i = 10, j = 20; double x = 5.1, y = 2.2; cout << sum(&i, &j) << " " << sum(&x, &y);} a. 25 2.3 b. 20 2.3 c. 25 22 2. Укажите правильное объявление шаблона функции, если в программе производит-ся вызов double x = zero<double>(); a. template TT zero() { return 0;} b. template TT zero() { return 0;} c. template TT zero { return 0;} 3. При определении класса-шаблона a. он должен быть включен в общий контекст (namespace) STL b. он должен быть наследован от класса Template c. он должен быть отмечен ключевым словом template d. он должен включать в себя, по крайней мере, один параметр-тип 4. Можно ли в шаблоне класса опеределить статический метод? a. да b. нет, будет ошибка компиляции c. да, но результат работы программы непредсказуем 5. Отметьте все утверждения, которые считаете верными: a. нельзя с помощью шаблона создать функцию с таким же именем, как у явно определенной функции b. цель введения шаблонов – создание функций, которые могут обрабатывать разнотипные данные c. в качестве описания шаблона функции используется прототип шаблона: template <список _параметров_шаблона > 6. Отметьте верное утверждение: a. шаблон может быть членом класса или шаблона класса b. шаблон может быть членом только шаблона класса c. шаблон может быть членом только класса d. шаблон не может быть членом класса или шаблона-класса 7. Сколько параметров может быть у шаблона при определении шаблона функции ? a. столько, сколько аргументов у функции b. столько, сколько типов используется для параметризации 8. Для чего предназначены манипуляторы потоков ввода-вывода? a. для управления аргументами функций классов istream и ostream b. для перенаправления потоков ввода-вывода c. для создания объеков классов istream и ostream 9. Что будет выведено на экран после выполнения программы? int main() { try{ try {throw 3.14; } catch (int) { cout << "Exception 1"; }} catch (...) {cout << "Exception 2"; } return 0;} a. Exception 1Exception 2 b. Exception 1 c. Exception 2 10. Какие ключевые слова используются для создания и обработки исключительных ситуаций? a. try b. delete c. catch d. return e. throw 11. Блок try catch a. должен стоять в функции main b. заключает участок кода, в котором может сложиться исключительная ситуа-ция c. может быть повторен несколько раз в одной функции d. не может быть вложенным 12. Если заданы классы ... то что будет выведено при выполнении оператора class A {... } A1; class B : public A { ... } B1; class C : public B { ... } C1; то что будет выведено при выполнении оператора throw (C1); а обработка исключительной ситуации записана catch (B& b) { cout << 1; } catch (C& c) { cout << 2; } catch (A& a) { cout << 3; } catch (...) { cout << 4; } a. 1 b. 2 c. 3 d. 4 e. 1 2 3 4 f. 2 3 4 13. Укажите правильное использование оператора friend a. class A { friend int CountPass(); private: short i;}; b. class A { public : friend int A::CountPass(); private: short i;}; c. class A { public : int A1::CountPass(); friend: short i;}; 14. Сколько производных классов можно получить из базового класса? a. неограниченное количество b. количество производных классов определяется количеством базовых клас-сов c. определяется наличием абстрактного класса 15. Абстрактный класс – это класс, в котором a. есть виртуальный конструктор b. есть виртуальный деструктор c. есть хотя бы один виртуальный метод 16. Наследование характеризуется … a. посылкой сообщений объектам b. способностью объекта сохранять свойства и методы класса-родителя c.

Согласована на портале № 1396

возможностью задания в иерархии объектов различных действий в методе с одним именем 17. Понятие this в классе a. Указатель this является скрытым аргументом метода, превращает функцию в область памяти только для чтения b. Объект this является аргументом метода другого класса, существует во всех методах и указывает на адрес: this -><объект> c. Указатель this является скрытым аргументом метода, существует во всех ме-тодах объекта и указывает на его (объект) адрес: this ->< объект > 18. Какой принцип ООП нарушает следующий фрагмент кода: class Counter { public int count; public void increment() { count++; } public int get() { return count; }} public class Main { public static void main(String[] args) { Counter counter = new Counter(); counter.count = 5; }} a. инкапсуляция b. равенство c. композиция d. полиморфизм e. агрегация 19. Какой правильный вариант описания шаблона семейства классов? a. template class Array{. . . }; b. template (class T) class Array {. . . }; c. template {class T} class Array{. . . } 20. Порождение иерархии объектов происходит в процессе … a. наследования b. полиморфизма c. вызова соответствующего метода d. инкапсуляции 21. Сколько родительских классов может иметь класс? a. два b. один c. неограниченно 22. Какая функция выполняет начальную инициализацию данных в классе? a. деструктор b. нет правильного ответа c. конструктор 23. Что такое cout? a. объект типа iostream b. класс, который выводит данные на терминал c. переменная, которую программист должен создать для вывода данных 24. В объектно-ориентированном программировании структуру и функционирование множества объектов с подобными характеристиками, атрибутами и поведением описывает … a. класс b. полиморфизм c. событие d. метод e. сокрытием информации и комбинированием данных и методов внутри объ-екта 25. В объектно-ориентированном программировании при создании объекта первым вызывается … a. конструктор b. произвольный метод в зависимости от ситуации c. первый описанный в программе метод d. деструктор

3.3 Экзаменационные вопросы

– Философия программирования, заложенная в языке C++. Методика создания программ. Имена, их область существования и видимости. Атрибуты компоновки имен. Структура программы на языке C++. Представление данных. Имена типов данных. Встроенные и составные типы данных. Квалификатор const. Функции языка C+ , их отличия от функций языка C. Рекурсивный вызов функции. Тип «указатель на функцию». Ссылочные переменные. Аргументы, принимаемые по умолчанию. Перегрузка функций. Шаблоны функций. Понятие класса. Определение и реализация класса. Деструкторы и конструкторы классов, их вызов. Члены-данные и члены-методы классов. Уровни доступа класса. Создание и использование объектов класса. Работа с указателем this. Создание массивов объектов. Работа с классами. Перегрузка операций. Использование дружественных классов. Конструкторы копирования и перегруженные операторы присваивания. Статические члены класса. Использование указателей на объекты. Наследование классов. Защита доступа при наследовании. Виртуальные функции, механизм их вызова. Статическое и динамическое связывание. Реализация виртуальных функций. Абстрактные базовые классы. Виртуальные базовые классы. Шаблоны классов. Множественное наследование. Потоковые классы. Форматирование. Флаги форматирования. Манипуляторы. Средства преобразования типов. Явные преобразования static_cast, dynamic_cast, reinterpret_cast. Обработка исключительных ситуаций.

3.4 Темы лабораторных работ

– Выполнение индивидуального задания на тему «Обработка исключительных ситуаций».
– Создание шаблона класса для описания обобщенного массива.
– Выполнение индивидуального задания на тему «Виртуальные функции».
– Выполнение индивидуального задания на тему «Наследование классов».
– Создание классов для работы с динамическими данными переменной размерности.
– Выполнение индивидуального задания на тему «Создание класса с перегрузкой некоторых операций».
– Выполнение индивидуального задания на тему «Разработка простейшего пользовательского типа данных».
– Изучение интегрированной среды разработки Visual Studio 8.0. Использование объектов типа «указатель на функцию» для обеспечения полиморфного поведения программы.

Согласована на портале № 1396

4 Методические материалы

Для обеспечения процесса обучения и решения задач обучения используются следующие материалы:

– методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы фор-мирования компетенций, согласно п. 12 рабочей программы.

4.1. Основная литература

1. Головин И.Г. Языки и методы программирования. - М. : Академия, 2012. - 304 с. (наличие в библиотеке ТУСУР - 30 экз.)

4.2. Дополнительная литература

1. Ашарина, И.В. Объектно-ориентированное программирование в C++: лекции и упражнения. [Электронный ресурс] : учебное пособие. — Электрон. дан. — М. : Горячая линия-Телеком, 2012. — 319 с. — Загл. с экрана. [Электронный ресурс]. - http://e.lanbook.com/books/element.php?pl1_id=5115

4.3. Учебно-методическое пособие и программное обеспечение

1. Объектно-ориентированное программирование: Методические указания по выполнению лабораторных и практических работ / Панасенко Е. А., Касимов В. З. - 2012. 19 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/1514, свободный.

2. Объектно-ориентированное программирование: Методические указания к самостоятельной работе студентов / Панасенко Е. А. - 2012. 6 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/1566, свободный.

4.4. Базы данных, информационно справочные и поисковые системы

1. Поисковая система google.ru

Согласована на портале № 1396