УТВЕРЖДЕНО:
Ученым советом Высшей школы сервиса
Протокол № 12 от «22 » мая 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.5 Прикладное программное обеспечение

основной профессиональной образовательной программы высшего образования – программы

бакалавриата

по направлению подготовки: ***43.03.01 Сервис***

направленность (профиль): ***Геоинформационный сервис***

Квалификация: ***бакалавр***

Год начала подготовки: ***2019***

Разработчики:

должность	ученая степень и звание, ФИО
Доцент	***к.т.н., доцент Минитаева А.М.***

Рабочая программа согласована и одобрена директором ОП ОП:

должность	ученая степень и звание, ФИО
Директор Высшей школы сервиса	***к.т.н., доцент Сумзина Л.В.***

1. Аннотация рабочей программы дисциплины Б1.В5 «Прикладное программное обеспечение»

Дисциплина Б1.В.5 «Прикладное программное обеспечение» относится к части, формируемой участниками образовательного процесса первого блока программы бакалавриата по направлению подготовки 43.03.01 Сервис, профилю Геоинформационный сервис

Дисциплина направлена на формирование следующих компетенций выпускника:

ПК УВ-4 - Способен производить выбор и использовать прикладное программное обеспечение в профессиональной деятельности; в части индикаторов достижения компетенции ПКУВ-4.1. (Выбирает современные информационные технологии и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач), ПКУВ-4.2. (Использует средства информационных, компьютерных и сетевых технологий, прикладное программное обеспечение при решении задач профессиональной деятельности).

Содержание дисциплины охватывает круг вопросов, связанных с изучением правила построения схем алгоритмов программ и диаграмм классов, изображения схем алгоритмов программ и диаграммы классов, средства, предоставляемые программисту универсальным языком высокого уровня, приемы и методику решения базовых вычислительных задач; принципы организации данных на периферийных устройствах (на уровне средств, предоставляемых универсальным языком); принципы управления динамическим распределением памяти под размещение данных; принципы объектной декомпозиции и построения классов, разрабатывать алгоритмы программ решения базовых вычислительных задач, записывать их на универсальном языке программирования, навыками использования программной среды для создания программ и их отладки, приемами документирования программного обеспечения при структурном и объектном подходах.

Общая трудоемкость освоения дисциплины составляет 10 зачетных единиц, 360 часа. Преподавание дисциплины ведется на 4 курсе, с 7-8 семестры.

На очной форме обучения продолжительностью по 17 недель каждый, и предусматривает проведение учебных занятий следующих видов: лекции (68 ч.), в том числе, традиционная лекция с презентацией, практические занятия (108 ч.) в форме практических работ, самостоятельная работа обучающихся (176 ч.), групповые и индивидуальные консультации (4 ч.), промежуточная аттестация (4 ч.).

На заочной форме обучения предусматривает проведение учебных занятий следующих видов: лекции (12 ч.), в том числе, традиционная лекция с презентацией, практические занятия (16 ч.) в форме практических работ, самостоятельная работа обучающихся (324 ч.), групповые и индивидуальные консультации (4 ч.), промежуточная аттестация (4 ч.).

Программой предусмотрены следующие виды контроля: текущий контроль успеваемости в форме защиты практических работ, тестирования, промежуточная аттестация в форме экзаменов в 7 и 8 семестрах для очной и заочной форм обучения.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

№	Индекс	Планируемые результаты обучения

пп	компетенции, индикатора	(компетенции, индикатора)
1.	ПК УВ-4	Способен производить выбор и использовать прикладное программное обеспечение в профессиональной деятельности ПКУВ -4.1. Выбирает современные информационные технологии и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач ПКУВ -4.2. Использует средства информационных, компьютерных и сетевых технологий, прикладное программное обеспечение при решении задач профессиональной деятельности

3. Место дисциплины (модуля) в структуре ООП:

Дисциплина «Прикладное программное обеспечение» является частью первого блока первого блока программы бакалавриата 43.03.01 «Сервис» профиль «Геоинформационный сервис» и относится к части, формируемой участниками образовательных отношений.

4. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 10 зачетных единиц/ 360 акад.часа.

Для очной формы обучения:

Виды учебной деятельности	Всего	Семестры	
		7	8
Контактная работа обучающихся	184	92	92
в том числе:	-	-	-
Лекции	68	34	34
Практические занятия	108	54	54
Консультации	6	2	2
Промежуточная аттестация	6	2	2
Самостоятельная работа	176	88	88
Форма промежуточной аттестации		Экз	Экз
Общая трудоемкость час,	360	180	180
з.е.	10	5	5

Для заочной формы обучения:

Виды учебной деятельности	Всего	Семестры	
		7	8
Контактная работа обучающихся	36	18	18
в том числе:	-	-	-
Лекции	22	6	6

Практические занятия	22	8	8
Консультации	6	2	2
Промежуточная аттестация	6	2	2
Самостоятельная работа	324	162	162
Форма промежуточной аттестации	78	Экз	Экз
Общая трудоемкость час,	360	180	180
з.е.	10	5	5

5. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
1,2/7	1 Блок. Основы алгоритмизации и программирование	Введение в программирование	4	Традиционная с презентацией							44	Ознакомление с литературой по дисциплине на сайте ЭБС znanium.com. При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов Подготовка к 1-ой контрольной точке, в виде защиты практических работ. Подготовка ко 2-ой контрольной точке, в виде теста.
3,4/7		Введение в языки программирования С и С++	4	Традиционная с презентацией								
5,6/7		Введение в разработку программных продуктов	4	Традиционная с презентацией								
1-5/7		ПЗ 1: Установка Интегрированной Среды Разработки (IDE). Компиляция вашей первой программы			10	Практическая работа						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
6/7		Защита практических работ. (К.т.№1)			10	Защита практических работ. (К.т.№1)						
6/7		Тестирование. (К.т.№2)			7	Тестирование. (К.т.№2)						
7/7	2 Блок. Структурные типы данных Основы С++	Структура программ	4	Традиционная с презентацией							44	Самостоятельная проработка теоретического материала из рекомендованных источников. Подготовка к практическим занятиям. Подготовка к 3-ой контрольной точке, в виде защиты практических работ. Подготовка к 4-ой контрольной точке, в виде теста.
8/7		Комментарии Переменные. Инициализация и Присваивание	4	Традиционная с презентацией								
9/7		Разработка схем алгоритма	4	Традиционная с презентацией								
10/7		cout, cin и endl Функции и оператор возврата return	4	Традиционная с презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
11/7		Параметры и аргументы функций	4	Традиционная с презентацией								
12-17/7		Локальная область видимости. Ключевые слова и идентификаторы Операторы.	2	Традиционная с презентацией			-	-	-	-		
8 – 17/7		ПЗ 2: Базовое форматирование кода Предварительное объявление и прототип функции Многофайловые программы Заголовочные файлы			10	Практическая работа						
17/7		Защита практических работ. (К.т.№3)			10	Защита практических работ. (К.т.№1)						
17/7		Тестирование. (К.т.№4)			7	Тестирование.						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
						(К.т.№2)						
Консультация – 2 часа												
Промежуточная аттестация – зачет – 2 часа												
		8 семестр										
1-4/ 7,8	**3 Блок.** Директивы препроцессора **C++**	Директивы препроцессора Header guards Конфликт имён и std namespace Разработка ваших первых программ	3	Традиционная с презентацией							22	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов).Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты
5-8/ 7,8		Отладка программ: степпинг и точки останова	3	Традиционная с презентацией								
9-10/ 8		Отладка программ: стек вызовов и отслеживание переменных	3	Традиционная								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		профессиональной деятельности.		с презентацией								практических работ. Подготовка ко 2-ой контрольной точке, в виде теста.
1-9/8		ПЗ 3 Решение задач и их отладка, представление схемы алгоритма			5	Практическая работа						
10/8		Защита практических работ. (К.т.№1)	3		5	Защита практических работ. (К.т.№1)						
10/8		Тестирование. (К.т.№1)			4	Тестирование. (К.т.№1)						
11-12/8	**4 Блок**. Переменные и основные типы данных в	Детальнее об инициализации, присваивании и определении переменных	3	Традиционная с презентацией							22	Самостоятельная проработка теоретического материала из рекомендованных источников. Подготовка к практическим
13-14/		Тип данных void Размер типов данных и	3	Традиционная								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
8	C++	оператор sizeof		с презентацией								занятиям. Подготовка к 1-ой контрольной точке, в виде защиты практических работ. Подготовка к 2-ой контрольной точке, в виде теста.
15-16/8		Целочисленные типы данных Фиксированный размер целочисленных типов Типы данных с плавающей точкой	3	Традиционная с презентацией								
17/8		Логический тип данных Символьный тип данных	3	Традиционная с презентацией								
11-16/8		ПЗ 4: Литералы и магические числа, const, constexpr и символьные константы.			5	Практическая работа						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
17/8		Защита практических работ. (К.т.№2)			5	Защита практических работ. (К.т.№2)						
17/8		Тестирование. (К.т.№2)			4	Тестирование. (К.т.№2)						
1 -4 /8	**5 Блок.** Операторы в С++	Приоритет операций и правила ассоциативности Арифметические операторы	2	Традиционная с презентацией							22	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов). Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты
5-8 /8		Инкремент, декремент и побочные эффекты Sizeof, запятая и условный тернарный оператор	2	Традиционная с презентацией								
9-10/8		Операторы сравнения Логические операторы: И, ИЛИ, НЕ	2	Традиционная с презент								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
				ацией								практических работ. Подготовка ко 2-ой контрольной точке, в виде теста.
1-9/8		ПЗ 5: Конвертация чисел из двоичной системы в десятичную и наоборот Побитовые операторы Битовые флаги и битовые маски			5	Практическая работа						
10/8		Защита практических работ. (К.т.№3)			4	Защита практических работ. (К.т.№3)						
10/8		Тестирование. (К.т.№3)			4	Тестирование. (К.т.№3)						
11-16/8	6 Блок. Область видимости	Блоки стейтментов. Локальные переменные, область видимости и	2	Традиционная							22	При решении задач во время семинарских занятий описание

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
	и другие типы переменных в C++	продолжительность. Глобальные переменные.		презентацией								решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов). Подготовка к практическим занятиям. Подготовка к 3-ой контрольной точке, в виде защиты практических работ. Подготовка к 4-ой контрольной точке, в виде теста.
17/8	Введение в программирование на Python. Основные алгоритмические конструкции и библиотеки	Статические переменные. Связи, область видимости и продолжительность. Пространства имён Стейтменты using Неявное преобразование типов данных. Явное преобразование типов данных. Программа на Python, Оператор условия и выбора, Циклы функция	2	Традиционная с презентацией								
11-16/8		ПЗ 6: Введение в std::string Перечисления			5	Практическая работа						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад. часов	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		Классы enum Псевдонимы типов: typedef и type alias**.** Структуры**.** Вывод типов: ключевое слово auto, Оператор условия и выбора, Циклы функция Встроенные типы данных										
17/8		Защита практических работ. (К.т.№4)			4	Защита практических работ. (К.т.№4)						
17/8		Тестирование. (К.т.№4)			4	Тестирование. (К.т.№4)						
Консультация – 2 часа												
Промежуточная аттестация – экзамен – 2 часа												

Для заочной формы обучения

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
7	1 Блок. Порядок выполнения кода в программе. Циклы, ветвления в C++	Порядок выполнения программ. Операторы управления Операторы условного ветвления if и else Оператор switch Оператор goto Цикл while Цикл do while Цикл for	3	Традиционная с презентацией							81	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов). Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты практических работ. Подготовка ко 2-ой контрольной точке, в виде теста.
7		Операторы break и continue Генерация случайных чисел. Функции srand() и rand() Обработка некорректного ввода через std::cin		Традиционная с презентацией								
7		Введение в тестирование		Традицион								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		кода		ная с презентацией								
7		ПЗ 1: Массивы. Часть 1			2	Практическая работа						
7		Защита практических работ. (К.т.№1)			1	Защита практических работ. (К.т.№1)						
7		Тестирование. (К.т.№2)			1	Тестирование (К.т.№2)						
7	**2 Блок.** Массивы, Строки, Указатели и Ссылки в С++ Функции в С++	Массивы. Часть 2 Массивы и циклы Сортировка массивов методом выбора Многомерные массивы Строки C-style Указатели. Введение Нулевые указатели Указатели и массивы	3	Традиционная с презентацией							81	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов).

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		Адресная арифметика и индексация массива Символьные константы строк C-style Динамическое выделение памяти. Операторы new и delete Динамические массивы Указатели и const Ссылки Ссылки и const Оператор доступа к членам через указатель Цикл foreach Указатели типа void Указатели на указатели Введение в std::array Введение в std::vector. Векторы										Подготовка к практическим занятиям. Подготовка к 3-ой контрольной точке, в виде защиты практических работ. Подготовка к 4-ой контрольной точке, в виде теста.
7		Параметры и аргументы функций		Традиционная с								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		Передача аргументов по значению Передача аргументов по ссылке Передача аргументов по адресу Возврат значений по ссылке, по адресу и по значению Встроенные функции Перегрузка функций Параметры по умолчанию Указатели на функции Стек и Куча Емкость вектора. std::vector в качестве стека Рекурсия. Числа Фибоначчи и Факториал Обработка ошибок, cerr и exit		презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		Assert и static_assert Аргументы командной строки Эллипсис. Почему его не следует использовать										
7		Основы ООП в С++Урок Введение в ООП Классы. Объекты и методы классов Спецификаторы доступа public и private Инкапсуляция. Геттеры и Сеттеры Конструкторы		Традиционная с презентацией								
7		Список инициализации членов класса Инициализация нестатических членов класса		Традиционная с презентацией								
7		Делегирующие		Традицио								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		конструкторы Деструкторы Скрытый указатель this Классы и заголовочные файлы Классы и const Статические переменные-члены класса Статические методы класса Дружественные функции и классы Анонимные объекты Вложенные типы в классах Тайминг кода. Время выполнения программы		нная с презентацией								
7		Перегрузка операторов в C++ Введение в перегрузку операторов		Традиционная с презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		Перегрузка операторов через дружественные функции Перегрузка операторов через обычные функции Перегрузка операторов ввода и вывода Перегрузка операторов через методы класса Перегрузка унарных операторов плюс, минус и логического НЕ Перегрузка операторов сравнения Перегрузка операторов инкремента и декремента Перегрузка оператора индексации Перегрузка оператора (). Круглые скобки Перегрузка операций преобразования типов										

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		данных.										
7		ПЗ 2 Конструктор копирования Копирующая инициализация Конструкторы преобразования. Ключевые слова explicit и delete Перегрузка оператора присваивания Поверхностное и глубокое копирование			2	Практическая работа						
7		Защита практических работ. (К.т.№3)			1	Защита практических работ. (К.т.№1)						
7		Тестирование. (К.т.№4)			1	Тестирование. (К.т.№2)						
Консультация – 2 часа												

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
Промежуточная аттестация – зачет – 2 часа												
7	**3 Блок.** Введение в отношения между объектами в С++	Отношения между объектами Композиция объектов Агрегация Ассоциация Зависимость Контейнерные классы Список инициализации std::initializer_list	2	Традиционная с презентацией							41	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов, диаграмм классов и объектов). Составление терминологического словаря. Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты практических работ.
7		Наследование в С++Наследование. Введение Базовое наследование в С++ Порядок построения дочерних классов Конструкторы и инициализация		Традиционная с презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		дочерних классов Наследование и спецификаторы доступа. Protected										Подготовка ко 2-ой контрольной точке, в виде теста.
8		Добавление нового функционала в дочерний класс Вызов и переопределение методов родительского класса		Традиционная с презентацией								
8		ПЗ 3: Сокрытие методов родительского класса Множественное наследование			1	Практическая работа						
8		Защита практических работ. (К.т.№1)			0.5	Защита практических работ. (К.т.№1)						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
8		Тестирование. (К.т.№1)			0.5	Тестирование. (К.т.№1)						
8	4 Блок. Виртуальные функции в С++	Указатели/Ссылки и Наследование Виртуальные функции и Полиморфизм Модификаторы override и final. Ковариантный тип возврата	2	Традиционная с презентацией							41	Самостоятельная проработка теоретического материала из рекомендованных источников. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты практических работ. Подготовка к 2-ой контрольной точке, в виде теста.
8		Виртуальные деструкторы и присваивание Раннее и Позднее Связывания		Традиционная с презентацией								
8		Виртуальные таблицы Чистые виртуальные функции. Интерфейсы и Абстрактные классы		Традиционная с презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
8		Виртуальный базовый класс Обрезка объектов		Традиционная с презентацией								
8		ПЗ 4: Динамическое приведение типов. Оператор dynamic_cast Вывод объектов классов через оператор вывода			1	Практическая работа						
8		Защита практических работ. (К.т.№2)			0,5	Защита практических работ						
8		Тестирование. (К.т.№2)			0.5	Тестирование						
8	5 Блок. Шаблоны в С++	Шаблоны функций Экземпляры шаблонов функций	1	Традиционная с презентацией							40	При решении задач во время семинарских занятий описание решений доводится до проекта программы (схем алгоритмов,
8		Шаблоны классов Параметр non-type		Традиционная с								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		шаблона		презентацией								диаграмм классов и объектов). Составление терминологического словаря. Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям. Подготовка к 1-ой контрольной точке, в виде защиты практических работ. Подготовка ко 2-ой контрольной точке, в виде теста.
8		Явная специализация шаблона функции Явная специализация шаблона класса Частичная специализация шаблона Частичная специализация шаблонов		Традиционная с презентацией								
8		ПЗ 5: Указатели			1	Практическая работа						
8		Защита практических работ. (К.т.№3)			0.5	Защита практических работ						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
8		Тестирование. (К.т.№3)			0.5	Тестирование						
8	6 Блок. Стандартная библиотека шаблонов (STL) в C++. Ввод/Вывод в C++, Область видимости и другие типы переменных в C++. Ведение в программирование на Python. Основные алгоритмические конструкции.	Стандартная библиотека шаблонов (STL) Контейнеры STL Итераторы STL Алгоритмы STL	1	Традиционная с презентацией							40	Самостоятельная проработка теоретического материала из рекомендованных источников. Подготовка к практическим занятиям. Подготовка к 3-ой контрольной точке, в виде защиты практических работ. Подготовка к 4-ой контрольной точке, в виде теста.
8		Потоки ввода/вывода Функционал класса istream Функционал классов ostream и ios. Форматирование вывода Потоковые классы и Строки Состояния потока и валидация пользовательского ввода Базовый файловый ввод/вывод		Традиционная с презентацией								

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
	Библиотеки	Рандомный файловый ввод/вывод										
8		std::string в C++ std::string и std::wstring Создание, уничтожение и конвертация std::string Длина и ёмкость std::string Доступ к символам std::string. Конвертация std::string в строки C-style Присваивание и перестановка значений с std::string Добавление к std::string Вставка символов и строк в std::string, Программа на Python, Оператор условия и			1	Практическая работа						

Номер недели, семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения									
			Лекции, акад.	Форма проведения лекции	Практические занятия, акад.часов	Форма проведения практического занятия	Семинары, акад. часов	Форма проведения	Лабораторные работы, акад.часов	Консультация, аттестация	СРС, акад.часов	Форма проведения СРС
		выбора, циклы, функция										
8		Защита практических работ. (К.т.№4)			0,5	Защита практических работ						
8		Тестирование (К.т.№4)			0.5	Тестирование.						
Консультация – 2 часа												
Промежуточная аттестация – экзамен – 2 часа												

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Для самостоятельной работы по дисциплине обучающиеся используют следующее учебно-методическое обеспечение:

№ п/п	Тема, трудоемкость в акад.ч.	Учебно-методическое обеспечение
1	**Основы алгоритмизации и программирование. Порядок выполнения кода в программе. Циклы, ветвления в С++, 28 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914
2	**Структурные типы данных Основы С++. Массивы, Строки, Указатели и Ссылки в С++.Функции в С++, 28 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015.

		http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914
3	**Директивы препроцессора С++ Введение в отношения между объектами в С++, 30 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914

4	**Переменные и основные типы данных в С++. Виртуальные функции в С++, 30 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914
5	**Операторы в С++. 5 Блок. Шаблоны в С++, 30 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа:

		http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914
6	Область видимости и другие типы переменных в С++. Блок. Стандартная библиотека шаблонов (STL) в С++. Основы языка Python. Программа на Python, Оператор условия и выбора, циклы, функция. Библиотеки. Ввод/Вывод в С++, **30 часа** Заочная форма - 54 часа	1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428 2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194 3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046 4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098 5. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008 6. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914

7. Фонд оценочных средств для проведения текущей и промежуточной аттестации обучающихся по дисциплине (модулю)

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п п	Индекс компе-тенции, индикатора	Содержание компетенции, индикатора	Раздел дисциплины, обеспечивающий этапы формирование компетенции, индикатора	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции, индикатора обучающийся должен:		
				знать	уметь	владеть
1	ПК УВ-4	Способен производить выбор и использовать прикладное программное обеспечение в профессиональной деятельности				
		ПКУВ -4.1. Выбирает современные информационные технологии и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	Все разделы	Знает современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы для решения профессиональных задач.	Умеет обосновывать выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывать оригинальные программные средства для решения профессиональных задач.	Владеет навыками подбора программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий, для решения профессиональных задач
		ПКУВ -4.2. Использует средства информационных, компьютерных и сетевых технологий, прикладное программное обеспечение при решении задач профессиональной деятельности	Все разделы	-Принципы построения современных информационных компьютерных и сетевых систем; -аппаратно-техническое и программное обеспечение глобальных компьютерных сетей и корпоративных информационных систем	Уметь применять современное программное обеспечение, позволяющее решать основные задачи в профессиональной деятельности	Владеет инструментальными средствами, применяемыми для контроля принимаемых решений

7.2 Описание показателей и критериев оценивания компетенций на разных этапах их формирования, описание шкал оценивания

Результат обучения по дисциплине	Показатель оценивания	Критерий оценивания	Этап освоения компетенции
Знать основы прикладного программного обеспечения в профессиональной деятельности; принципы разработки программных приложений. Уметь разрабатывать программные	Защита практической работы, тестирование	Студент демонстрирует знание основы прикладного программного обеспечения в профессиональной деятельности; принципы разработки программных приложений.	закрепление способности производить выбор и использовать прикладное программное обеспечение в профессиональной деятельности

приложения с учетом схем алгоритма, декомпозиции объектов. Владеть навыками применения методологии объектно-ориентированного анализа и конструирования объектов геоинформационного сервиса.		Студент демонстрирует умение разрабатывать программные приложения с учетом схем алгоритма, декомпозиции объектов. Студент демонстрирует владение навыками применения методологии объектно-ориентированного анализа и конструирования объектов геоинформационного сервиса	

Критерии и шкала оценивания освоения этапов компетенций на промежуточной аттестации

Контроль промежуточной успеваемости студентов по дисциплине строится на бально-рейтинговой системе и заключается в суммировании баллов, полученных студентом по результатам текущего контроля и итоговой работы.

Текущий контроль реализуется в формах тестирования, оценки качества и активности работы на практических занятиях, анализа добросовестности и самостоятельности при написании творческих работ, решения задач, посещаемости занятий и т.д. В семестре по дисциплине устанавливается мероприятия текущего контроля успеваемости (4 контрольных точки). Выполнение всех заданий текущего контроля является обязательным для студента и является основанием для допуска к промежуточной аттестации.

К критериям выставления рейтинговых оценок текущего контроля относятся:

Основные критерии:

- оценка текущей успеваемости по итогам работы на семинарах;
- оценки за письменные работы (рефераты, доклады, решение задач и др.);
- оценки текущей успеваемости по итогам интерактивных форм практических занятий (деловые игры, дискуссии и др.);
- посещение учебных занятий.

Дополнительные критерии:

- активность на лекциях и семинарских занятиях, интерес к изучаемому предмету;
- владение компьютерными методами изучения предмета, умение готовить презентации для конференций, использование Интернета, профессиональных баз данных при подготовке к занятиям и написании письменных работ;
- обязательное посещение учебных занятий;
- оценка самостоятельной работы студента;
- участие студента в работе организуемых кафедрой (филиалом) круглых столов, конференций и пр.;
- общий уровень правовой культуры, эрудиция в области правовых проблем.

Результаты промежуточной аттестации определяются оценками "отлично", "хорошо", "удовлетворительно", "неудовлетворительно" (форма промежуточной

аттестации – экзамен или дифференцированный зачет) и "зачтено", "не зачтено" (форма промежуточной аттестации – зачет).

В соответствии с Положением «о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата и программам магистратуры, реализуемым по федеральным государственным образовательным стандартам» рейтинговая оценка студентов по каждой учебной дисциплине независимо от ее общей трудоемкости, определяется по 100-балльной шкале в каждом семестре. Распределение баллов рейтинговой оценки между видами контроля рекомендуется устанавливать в следующем соотношении:

Посещаемость – посещение занятий лекционного типа (за исключением поточных) и занятий семинарского типа оценивается накопительно следующим образом: максимальное количество баллов, отводимых на учет посещаемости (30 баллов), делится на количество лекций (за исключением поточных) и практических занятий по дисциплине. Полученное значение определяет количество баллов, набираемых студентом за посещение одного занятия. По решение Ученого совета Высшей школы бизнеса, менеджмента и права посещаемость учебных занятий может не учитываться при оценивании результатов освоения дисциплин.

Успеваемость – оценка успеваемости выставляется за выполнение заданий текущего контроля по дисциплине. Как правило, в семестре 4 мероприятия текущего контроля (4 «контрольных точки»), причем выполнение всех 4 заданий текущего контроля является обязательным для студента. При обнаружении преподавателем в выполненном студентом задании плагиата данное задание оценивается 0 баллов и считается не выполненным.

Практические занятия (между «контрольными точками») проводятся в активной и интерактивной форме (дискуссии по изученному материалу, разбор ситуаций и т.п.), в аудитории или вне аудитории (на выставке, например). Несмотря на то, что преподаватель не оценивает в баллах студента на каждом занятии, в тоже время преподаватель фиксирует активность на занятии и при подведении итогов за семестр начисляет от 0 до 5 рейтинговых бонусных баллов за активность на занятиях.

Результаты текущего контроля успеваемости учитываются при выставлении оценки в ходе промежуточной аттестации.

Для допуска к промежуточной аттестации обучающийся должен выполнить все мероприятия текущего контроля по дисциплине (не иметь задолженностей по текущей контролю успеваемости) и набрать в общей сложности не менее 51 балла.

Перевод рейтинговых баллов в итоговую 5 – балльную шкалу оценку осуществляется в соответствии с таблицей.

Баллы за семестр	Автоматическая оценка		Баллы за зачет	Баллы за экзамен	Общая сумма баллов	Итоговая оценка
	зачет	экзамен				
90-100*	зачет	5 (отлично)	-	-	90-100	5 (отлично)
71-89*	зачет	4 (хорошо)	-	0-20	71-89 90-100	4 (хорошо) 5 (отлично)
51-70*	зачет	3 (удовлетворительно)	-	0-20	51-70 71-89 90	3 (удовлетворительно) 4 (хорошо) 5 (отлично)
50 и менее	недопуск к зачету, экзамену		-	-	50 и менее	2 (неудовлетворительно), незачет

* при условии выполнения всех заданий текущего контроля успеваемости

Виды средств оценивания, применяемых при проведении текущего контроля и шкалы оценки уровня знаний, умений и навыков при выполнении отдельных форм текущего контроля

Средство оценивания – защита практической работы

Шкала оценки уровня знаний, умений и навыков при выполнении контрольного задания

Оценка	Критерии оценивания	Показатели оценивания
«5»	- правильно определены основы теории функционирования объектов геоинформационного сервиса; - корректно раскрыта сущность теории системного представления объектов; - логично изложены преимущества и недостатки системного представления и совершенствования объектов	– Обучающийся показывает всесторонние и глубокие знания программного материала; – последовательно и четко отвечает на дополнительные вопросы; – демонстрирует способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы, проявляет творческие способности в понимании, изложении и использовании программного материала; – подтверждает полное освоение компетенций, предусмотренных программой
«4»	- правильно определены основы теории функционирования объектов геоинформационного сервиса; - корректно раскрыта сущность теории системного представления объектов; - логично изложены преимущества и недостатки системного представления и совершенствования объектов	– Обучающийся способен показать знания программного материала; – четко отвечает на дополнительные вопросы; – демонстрирует способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы, проявляет творческие способности в понимании, изложении и использовании программного материала; - подтверждает полное освоение компетенций, предусмотренных программой
«3»	- допущены неточности в определении теории функционирования объектов геоинформационного сервиса; - допущены неточности в раскрытии сущности теории функционирования	– Обучающийся показывает знания меньшей части программного материала; – отвечает не на все дополнительные вопросы;

	объектов геоинформационного сервиса; - допущены неточности в изложении преимуществ и недостатков системного представления и совершенствования объектов.	– Демонстрирует частичную способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы; – подтверждает полное освоение компетенций, предусмотренных программой.
«2»	- неверно определена теории функционирования объектов геоинформационного сервиса; - некорректно раскрыта сущность теории функционирования объектов геоинформационного сервиса; - некорректно изложены преимущества и недостатки системного представления и совершенствования объектов..	– Обучающийся не демонстрирует знания программного материала; – не отвечает на дополнительные вопросы; – Не демонстрирует способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы; – компетенции, предусмотренные программой, не освоены.

Средство оценивания – тестирование

Шкала оценки уровня знаний, умений и навыков при решении тестовых заданий

Критерии оценки	**оценка**
выполнено верно заданий	«5», если (90 –100)% правильных ответов
	«4», если (70 – 89)% правильных ответов
	«3», если (50 – 69)% правильных ответов
	«2», если менее 50% правильных ответов

Виды средств оценивания, применяемых при проведении промежуточной аттестации и шкалы оценки уровня знаний, умений и навыков при их выполнении

Устный опрос

Шкала оценки уровня знаний, умений и навыков при устном ответе

оценка	**Критерии оценивания**	**Показатели оценивания**
	– полно раскрыто содержание материала; – материал изложен грамотно, в определенной логической последовательности; – продемонстрировано системное и глубокое знание программного материала; – точно используется	– Обучающийся показывает всесторонние и глубокие знания программного материала, – знание основной и дополнительной литературы; – последовательно и четко отвечает на вопросы билета и дополнительные вопросы; – уверенно ориентируется в

«5»	терминология; – показано умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации; – продемонстрировано усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость компетенций, умений и навыков; – ответ прозвучал самостоятельно, без наводящих вопросов; – продемонстрирована способность творчески применять знание теории к решению профессиональных задач; – продемонстрировано знание современной учебной и научной литературы; – допущены одна – две неточности при освещении второстепенных вопросов, которые исправляются по замечанию	проблемных ситуациях; – демонстрирует способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы, проявляет творческие способности в понимании, изложении и использовании программного материала; – подтверждает полное освоение компетенций, предусмотренных программой
«4»	– вопросы излагаются систематизировано и последовательно; – продемонстрировано умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер; – продемонстрировано усвоение основной литературы. – ответ удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков: – а) в изложении допущены небольшие пробелы, не исказившие содержание ответа; – б) допущены один – два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя; – в) допущены ошибка или более двух недочетов при освещении второстепенных вопросов, которые легко исправляются по замечанию преподавателя	– обучающийся показывает полное знание – программного материала, основной и – дополнительной литературы; – дает полные ответы на теоретические вопросы билета и дополнительные вопросы, допуская некоторые неточности; – правильно применяет теоретические положения к оценке практических ситуаций; – демонстрирует хороший уровень освоения материала и в целом подтверждает освоение компетенций, предусмотренных программой
	– неполно или непоследовательно раскрыто содержание материала, но	– обучающийся показывает знание основного

«3»	показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения материала; – усвоены основные категории по рассматриваемому и дополнительным вопросам; – имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, исправленные после нескольких наводящих вопросов; – при неполном знании теоретического материала выявлена недостаточная сформированность компетенций, умений и навыков, студент не может применить теорию в новой ситуации; – продемонстрировано усвоение основной литературы	– материала в объеме, необходимом для предстоящей профессиональной деятельности; – при ответе на вопросы билета и дополнительные вопросы не допускает грубых ошибок, но испытывает затруднения в последовательности их изложения; – не в полной мере демонстрирует способность применять теоретические знания для анализа практических ситуаций; – подтверждает освоение компетенций, предусмотренных программой на минимально допустимом уровне
«2»	– не раскрыто основное содержание учебного материала; – обнаружено незнание или непонимание большей или наиболее важной части учебного материала; – допущены ошибки в определении понятий, при использовании терминологии, которые не исправлены после нескольких наводящих вопросов. – не сформированы компетенции, умения и навыки.	– обучающийся имеет существенные пробелы в знаниях основного учебного материала по дисциплине; – не способен, аргументировано и последовательно его излагать, допускает грубые ошибки в ответах, неправильно отвечает на задаваемые вопросы или затрудняется с ответом; – не подтверждает освоение компетенций, предусмотренных программой

Решение задач

Шкала оценки уровня знаний, умений и навыков при решении кейсов (ситуационных задач)

Предел длительности контроля	30 мин.
Критерии оценки	– было сформулировано и проанализировано большинство проблем, заложенных в кейсе (задаче); – были продемонстрированы адекватные аналитические методы при работе с информацией; – были использованы дополнительные источники информации для решения

	кейса(задачи); – были выполнены все необходимые расчеты; – подготовленные в ходе решения кейса документы соответствуют требованиям к ним по смыслу и содержанию; – выводы обоснованы, аргументы весомы; – сделаны собственные выводы, которые отличают данное решение кейса от других решений
Показатели оценки	max 10 баллов
«5», если (9 – 10) баллов	полный, обоснованный ответ с применением необходимых источников
«4», если (7 – 8) баллов	неполный ответ в зависимости от правильности и полноты ответа: - не были выполнены все необходимые расчеты; - не было сформулировано и проанализировано большинство проблем, заложенных в кейсе;
«3», если (5 – 6) баллов	неполный ответ в зависимости от правильности и полноты ответа: - не были продемонстрированы адекватные аналитические методы при работе с информацией; - не были подготовленные в ходе решения кейса документы, которые соответствуют требованиям к ним по смыслу и содержанию; - не были сделаны собственные выводы, которые отличают данное решение кейса от других решений

7.3 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.

Номер недели семестра	Раздел дисциплины обеспечивающий формирование компетенции (или ее части)	Вид и содержание контрольного задания	Требования к выполнению контрольного задания и срокам сдачи
6/7	1Блок Основы алгоритмизации и программирование. Порядок	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2- полностью правильный).
6/7		**Тест по блоку** «Основы	Проводится в письменном виде,

	выполнения кода в программе. Циклы, ветвления в С++	алгоритмизации и программирование. Порядок выполнения кода в программе. Циклы, ветвления в С++». Задание состоит из 10 вопросов.	ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.
17/7	2 Блок. Структурные типы данных Основы С++. Массивы, Строки, Указатели и Ссылки в С++ Функции в С++	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2- полностью правильный).
17/8		Тест по блоку «Процессы жизненного цикла геоинформационных систем». Задание состоит из 10 вопросов.	Проводится в письменном виде, ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.
10/8	1. Блок Директивы препроцессора С++ . 3 Введение в отношения между объектами в С++.	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2- полностью правильный).
10/8		Тест по блоку « Директивы препроцессора С++ . 3 Введение в отношения между объектами в С++». Задание состоит из 10 вопросов.	Проводится в письменном виде, ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.
17/8	4 Блок. Переменные и основные типы данных в С++. Виртуальные функции в С++	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2- полностью правильный).
17/8		Тест по блоку «Переменные и основные типы данных в С++. Виртуальные функции в С++». Задание состоит из 10 вопросов.	Проводится в письменном виде, ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.
10/8	5 Блок. Операторы в С++. Шаблоны в С++	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2-

			полностью правильный).
10/8		Тест по блоку «Операторы в С++. Шаблоны в С++». Задание состоит из 10 вопросов.	Проводится в письменном виде, ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.
17/8	6 Блок. Область видимости и другие типы переменных в С++. Стандартная библиотека шаблонов (STL) в С++. Ввод/Вывод в С++.	Защита практических работ	Каждый студент имеет уникальное задание, состоящее из -5 контрольных вопросов. Каждый правильный ответ оценивается от 0 до 2 баллов (0- неправильный ответ, 1 – правильный частично, 2- полностью правильный).
17/8		Тест по блоку «Область видимости и другие типы переменных в С++. Стандартная библиотека шаблонов (STL) в С++. Ввод/Вывод в С++». Задание состоит из 10 вопросов.	Проводится в письменном виде, ответы заносятся в бланк ответов. Каждый правильный ответ на вопрос оценивается в 1 балл. Максимальное количество баллов – 10.

Блок первый Основы алгоритмизации и программирование.
1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Перечень тем рефератов:

1. Синтаксис и семантика универсального языка программирования высокого уровня.
2. Структура программы. Описание данных, константы и переменные. Типы переменных. Выражения. Оператор присваивания. Процедуры ввода-вывода. Построение вычислительных программ линейной структуры.
3. Основные и дополнительные структурные конструкции управления процессом вычислений и их реализация операторами языка: условной передачи управления, выбора, конструкции циклов.
4. Организация программ разветвленной и циклической структуры на примере решения задач вычислительной математики: приближенное вычисление корня функции, приближенное вычисление суммы сходящегося бесконечного ряда и др

2 контрольная точка: **Вид контрольного задания - тесты**

1. Что называется алгоритмом:
а) протокол вычислительной сети
б) описание последовательности действий, строгое исполнение которых приводит к **решению поставленной задачи за конечное число шагов +**
в) правила выполнения определенных действий
2. Линейным называется алгоритм, если:
а) его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий +
б) он включает в себя вспомогательный алгоритм
в) он представим в табличной форме

3. Цикличным называется алгоритм, если:
а) он представим в табличной форме
б) ход его выполнения зависит от истинности тех или иных условий
в) он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий +
4. Алгоритм включает в себя ветвление, если:
а) ход его выполнения зависит от истинности тех или иных условий +
б) он включает в себя вспомогательный алгоритм
в) он представим в табличной форме
5. Что является свойством алгоритма:
а) цикличность
б) простота записи на языках программирования
в) результативность +
6. Как называется свойство алгоритма, заключающееся в том, что каждое действие и алгоритм в целом должны иметь возможность завершения:
а) результативность
б) конечность +
в) дискретность
7. Как называется свойство алгоритма, заключающееся в том, что алгоритм должен состоять из конкретных действий, следующих в определенном порядке:
а) массовость
б) детерминированность
в) дискретность +
8. Как называется свойство алгоритма, заключающееся в отсутствие ошибок, алгоритм должен приводить к правильному результату для всех допустимых входных значениях:
а) результативность +
б) детерминированность
в) массовость
9. Как называется свойство алгоритма, заключающееся в том, что один и тот же алгоритм можно использовать с разными исходными данными:
а) дискретность
б) массовость +
в) детерминированность
10. Как называется алгоритм, записанный на "понятном" компьютеру языке программирования:
а) текстовка
б) программа +
в) протокол алгоритма

Блок второй «Структурные типы данных Основы С++. Массивы, Строки, Указатели и Ссылки в С++. Функции в С++»

3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Структурные типы данных и модульное программирование.
2. Создание консольных приложений. Создание схем алгоритмов средствами Microsoft Visio и OpenOffice Draw.
3. Структурные типы данных: массивы, строки и записи (структуры).

4. Программирование с использованием структурных типов данных.
5. Представление обработки массивов, матриц и текстов.

4 контрольная точка: Вид контрольного задания - тесты.

Тесты:

1. В какой из следующих строк выполняется обращение к седьмому элементу массива, размер массива равен 10?

1. **mas[7];**
2. mas[6];
3. mas(7);
4. mas;

2. Укажите статическую строку!

1."Статическая строка"
2.char string[100];
3.'Статическая строка'

3. Код, указанный ниже объявляет массив ссылок. Правда это или нет?

```
int main()
{
        int& x[50];

  return 0;
}
```

1.нет
2. да

4. Как правильно высвободить память, после выполнения этого кода?

char *a; a = new char[20];
delete a;
delete a[];
delete [] a;

5. Какая из следующих записей возвращает значение переменной a, хранящееся в памяти по адресу на который указывает указатель?

1. val(a);
2. a ;
3.*a;
4. &a;

6. Укажите правильное объявление указателя в C++

1.int
2.***x**;
3.int &x;
4. int x;
5.ptr x;

7. Строковый типы данных в C++

1.строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением (‘/0’).

2.строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением '0'.

3. **строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением (‘’).**

8. Объявлена переменная char a; Какое из следующих выражений не верно?

1. **a = '3';**
2. a = 3;
3. a = "3";

9. Массив - это ...

1. Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и адресу

2. Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу и индексу

3. Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и индексу

10. Что такое ссылка?

1.нет правильного ответа
2.используется для переименования объектов
3.ссылка является псевдонимом для объекта
4. оператор

11.Укажите зарезервированное ключевое слово для динамического выделения памяти!

1.malloc
2.value
3. new
4.create

2. Блок третий «Директивы С++ . Введение в отношения между объектами в С++.»

1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Объектно-ориентированное программирование;
2. Языки интернет-программирования;
3. Базы данных;
4. Технология разработки программных систем.

2 контрольная точка: Вид контрольного задания - тесты

Тесты:

1. Выберите все правильные ответы. В переменной типа unsigned char можно хранить число

1.213
2.-213
3.-13
4.13
5.1213

2. Алфавит языка С++ включает в себя:

1. символы <, >, =, !
2. прописные латинские буквы
3. цифры
4.строчные латинские буквы

3. Элементарные конструкции (лексемы) языка С++ включают в себя:

1. операторы присваивания
2. **ключевые слова**
3. константы

4. Выберите все правильные утверждения.

1.в выражении можно использовать операнды булевского типа
2.результат операции сложения действительного и целого числа - действительный
3.результат операций сравнения действительных чисел - целый
4.результат стандартных функций sin и cos с аргументом целого типа - целый

*5. Чему равно числовое значение выражения e/2*a-abs(e)*1e0 при e = 4, a = 2?*

1. 0
2. **3**
3. 1

6.Выберите правильные утверждения:

1. целой переменной можно присвоить вещественную константу
2. **целой константе можно присвоить целую переменную**
3. целой переменной можно присвоить целую константу
4. целой константе можно присвоить вещественную переменную

7. Чему равно значение выражения (a && ! b || c), где a, b и c -величины типа bool, имеющие значения false, true и true соответственно?

1. True

1. false

7. Какие выражения не содержат синтаксических ошибок?

1. a*exp(t)\(2t)
2. **((cos(3*a+1.*abs(x))))**
3. **0XCC00*.34E-4/_do/k-2**
4. sin(abs(0.6(e*3))

8. Чему равно значение выражения (a || b && a || c), где a, b ис — величины типа bool, имеющие значения false, true и true соответственно?

1. **True**

1. false

*9. Чему равно числовое значение выражения sqrt(4)+142/20*2?*

1.12
2.16
3. 5

10. Какая из следующих операций языка C выполняется справа налево?

1.->
2.^
3.=
4.[]
5. ,

Блок четвертый «Переменные и основные типы данных в C++. Виртуальные функции в C++»

3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.

Темы рефератов:

1. Понятие Строки

2. Процедуры и функции
3. Подпрограммы. Средства отладки C++
4. Файловая система
5. Создание модулей. Процедурный тип параметров

4 контрольная точка: Вид контрольного задания - тесты:

1. Что называется наследованием?

1. * это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы

2. это механизм переопределения методов базового класса

3. это механизм, посредством которого производный класс получает все поля базового класса

4. это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

2. Выберите правильное объявление производного класса

1. class MoreDetails:: Details;

2. class MoreDetails: public class Details;

3. * class MoreDetails: public Details;

4. class MoreDetails: class(Details);

3. Выберите правильные утверждения:

1. если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям

2. * если элементы класса объявлены как private, то они недоступны ни наследникам класса, ни внешним функциям

3. если элементы объявлены как public, то они доступны наследникам класса, но не внешним функциям

4. * если элементы объявлены как public, то они доступны и наследникам класса, и внешним функциям

4. Возможность и способ обращения производного класса к элементам базового определяется

1. ключами доступа: private, public, protected в теле производного класса

2. только ключом доступа protected в заголовке объявления производного класса

3. * ключами доступа: private, public, protected в заголовке объявления производного класса

4. ключами доступа: private, public, protected в теле базового класса

5. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа прозводного класса к элементам базового

1. ключ доступа - public; в базовом классе: private; права доступа в производном классе - protected

2. * ключ доступа - любой; в базовом классе: private; права доступа в производном классе - нет прав

3. * ключ доступа - protected или public ; в базовом классе: protected; права доступа в производном классе - protected

4. ключ доступа - private; в базовом классе: public; права доступа в производном классе - public

5 * ключ доступа – любой; в базовом классе: public; права доступа в производном классе – такие же, как ключ доступа

6. Дружественная функция - это
1. функция другого класса, среди аргументов которой есть элементы данного класса
2. * функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса;
3. функция, являющаяся членом класса и объявленная с атрибутом friend;
4. функция, которая в другом классе объявлена как дружественная данному
7. Выберите правильные утверждения:
1. * одна функция может быть дружественной нескольким классам
2. дружественная функция не может быть обычной функцией, а только методом другого класса
3. * дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ
4. дружественная функция не может быть методом другого класса
8. Шаблон функции - это...
1. * определение функции, в которой типу обрабатываемых данных присвоено условное обозначение
2. прототип функции, в котором вместо имен параметров указан условный тип
3. определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
4. определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров
9. Выберите правильные утверждения:
1. * по умолчанию члены класса имеют атрибут private
2. по умолчанию члены класса имеют атрибут public;
3. члены класса имеют доступ только к элементам public;
4. * элементы класса с атрибутом private доступны только членам класса
10. Переопределение операций имеет вид:
1. имя_класса, ключевое слово operation, символ операции
2. * имя_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы
3. имя_класса, ключевое слово operator, список аргументов
4. имя_класса, два двоеточия, ключевое слово operator, символ операции

Блок пятый «Операторы в С++. Шаблоны в С++»
1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Процедуры и функции.
2. Формальные и фактические параметры.
3. Передача параметров по значению и ссылке.
4. Рекурсия.
5. Модули.
6. Файловая система.
7. Файлы.

2 контрольная точка: Вид контрольного задания - тесты:

1. Для доступа к элементам объекта используются:
1. * при обращении через имя объекта – точка, при обращении через указатель – операция «->»

2. при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
3. при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
4. при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

2. Полиморфизм – это :

1. * средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
2. средство, позволяющее в одном классе использовать методы с одинаковыми именами;
3. средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
4. средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

3. Полиморфизм реализован через механизмы:

1. * перегрузки функций, виртуальных функций, шаблонов
2. перегрузки функций, наследования методов, шаблонов;
3. наследования методов, виртуальных функций, шаблонов
4. перегрузки функций, наследования, виртуальных функций.

4. Виртуальными называются функции:

1. * функции базового класса, которые могут быть переопределены в производном классе
2. функции базового класса, которые не используются в производном классе;
3. функции базового класса, которые не могут быть переопределены в базовом классе;
4. функции производного класса, переопределенные относительно базового класса

5. Выберите правильный вариант выделения динамической памяти под переменную X типа float:

1. * float *ptr = new float; X = *ptr;
2. float & ptr = new float; X = & ptr;
3. float * ptr = &X; X = new float;

6. Полиморфизм в объектно-ориентированном программировании реализуется:

1. * через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны
2. через механизмы перегрузки (функций и операций) и шаблоны;
3. через виртуальные функции и шаблоны;
4. через механизмы перегрузки (функций и операций) и виртуальные функции

7. Дано определение класса

```
class monstr {
int health, armo;
monstr(int he, int arm);
public:
monstr(int he=50, int arm=10);
int color;
}
```

Укажите свойства и методы, доступные внешним функциям
- health, armo

```
monstr(int he, int arm);
monstr(int he=50, int arm=10);
```

- * int color;
monstr(int he=50, int arm=10);
- health, armo, color
monstr(int he=50, int arm=10);
- int color;
monstr(int he, int arm);
6. Переменная типа signed char может принимать значения

1. только символов английского алфавита, цифр и символа подчеркивания
2. **из первой половины кодовой таблицы**
3. **от -128 до 127**
4. только из алфавита языка С++

Блок шестой «Область видимости и другие типы переменных в С++. Стандартная библиотека шаблонов (STL) в С++. Ввод/Вывод в С++.»
3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Типизированные и текстовые файлы. Создание односвязных списков по типу очереди и по типу стека. Удаление элементов из списков и добавление элементов в них.
2. Простые классы и композиция. Наследование. Наполнение. Полиморфное наследование.
3. Создание однооконных приложений Windows.
4. Создание MDI многооконных приложений Windows.

4 контрольная точка: Вид контрольного задания - тесты:

1. Класс - это:
- любой тип данных, определяемый пользователем
- * тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- структура, для которой в программе имеются функции работы с нею
2. Членами класса могут быть
- * как переменные, так и функции, могут быть объявлены как private и как public
- только переменные, объявленные как private
- только функции, объявленные как private
- только переменные и функции, объявленные как private
- только переменные и функции, объявленные как public
3. Что называется конструктором?
- * метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса
4. Объект - это
- переменная, содержащая указатель на класс

- * экземпляр класса
- класс, который содержит в себе данные и методы их обработки
5. Отметьте правильные утверждения
- * конструкторы класса не наследуются
- конструкторов класса может быть несколько, их синтаксис определяется программистом
- * конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- конструктор возвращает указатель на объект
- * конструктор не возвращает значение
6. Что называется деструктором?
- метод, который уничтожает объект
- метод, который удаляет объект
- * метод, который освобождает память, занимаемую объектом
- системная функция, которая освобождает память, занимаемую объектом
7. Выберите правильные утверждения
- * у конструктора могут быть параметры
- конструктор наследуется, но должен быть перегружен
- конструктор должен явно вызываться всегда перед объявлением объекта
- * конструктор вызывается автоматически при объявлении объекта
- объявление каждого класса должно содержать свой конструктор
- * если конструктор не создан, компилятор создаст его автоматически
8. Выберите правильные утверждения
- деструктор - это метод класса, применяемый для удаления объекта
- * деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом
- деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
- * деструктор не наследуется
- деструктор наследуется, но должен быть перегружен
9. Выберите правильное объявление производного класса
- class MoreDetails:: Details;
- class MoreDetails: public class Details;
- * class MoreDetails: public Details;
- class MoreDetails: class(Details);
10. Выберите правильные утверждения:
- если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям
- * если элементы класса объявлены как private, то они недоступны ни наследникам класса, ни внешним функциям
- если элементы объявлены как public, то они доступны наследникам класса, но не внешним функциям
- * если элементы объявлены как public, то они доступны и наследникам класса, и внешним функциям

Промежуточная аттестация

Тесты для проверки знаний к первой промежуточной аттестации: экзамену

1.Возможность и способ обращения производного класса к элементам базового определяется

1. - ключами доступа: private, public, protected в теле производного класса
2. - только ключом доступа protected в заголовке объявления производного класса
3. **- * ключами доступа: private, public, protected в заголовке объявления производного класса**
4. - ключами доступа: private, public, protected в теле базового класса

2. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа прозводного класса к элементам базового

1. - ключ доступа - public; в базовом классе: private; права доступа в производном классе - protected
2. **- * ключ доступа - любой; в базовом классе: private; права доступа в производном классе - нет прав**
3. **- * ключ доступа - protected или public ; в базовом классе: protected; права доступа в производном классе - protected**
4. - ключ доступа - private; в базовом классе: public; права доступа в производном классе - public
5. **- * ключ доступа – любой; в базовом классе: public; права доступа в производном классе – такие же, как ключ доступа**

3. Дружественная функция - это

1. - функция другого класса, среди аргументов которой есть элементы данного класса
2. **- * функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса;**
3. - функция, являющаяся членом класса и объявленная с атрибутом friend;
4. - функция, которая в другом классе объявлена как дружественная данному
5. 17. Выберите правильные утверждения:
6. **- * одна функция может быть дружественной нескольким классам**
7. - дружественная функция не может быть обычной функцией, а только методом другого класса
8. **- * дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ**
9. - дружественная функция не может быть методом другого класса

4. Шаблон функции - это...

1. **- * определение функции, в которой типу обрабатываемых данных присвоено условное обозначение**
2. - прототип функции, в котором вместо имен параметров указан условный тип
3. - определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
4. - определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

5. Выберите правильные утверждения:

1. **- * по умолчанию члены класса имеют атрибут private**
2. - по умолчанию члены класса имеют атрибут public;

3. - члены класса имеют доступ только к элементам public;
4. **- * элементы класса с атрибутом private доступны только членам класса**

6. Переопределение операций имеет вид:
1. - имя_класса, ключевое слово operation, символ операции
2. **- * имя_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы**
3. - имя_класса, ключевое слово operator, список аргументов
4. - имя_класса, два двоеточия, ключевое слово operator, символ операции

7. Для доступа к элементам объекта используются:
1. **- * при обращении через имя объекта – точка, при обращении через указатель – операция «->»**
2. - при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
3. - при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
4. - при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

8. Полиморфизм – это :
1. **- * средство, позволяющее использовать одно имя для обозначения** действий, общих для родственных классов
2. - средство, позволяющее в одном классе использовать методы с одинаковыми именами;
3. - средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
4. - средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

9. Полиморфизм реализован через механизмы:
1. **- * перегрузки функций, виртуальных функций, шаблонов**
2. - перегрузки функций, наследования методов, шаблонов;
3. - наследования методов, виртуальных функций, шаблонов
4. - перегрузки функций, наследования, виртуальных функций.

10. Виртуальными называются функции:
1. **- * функции базового класса, которые могут быть переопределены в производном классе**
2. - функции базового класса, которые не используются в производном классе;
3. - функции базового класса, которые не могут быть переопределены в базовом классе;
4. - функции производного класса, переопределенные относительно базового класса

11. Выберите правильный вариант выделения динамической памяти под переменную X типа float:
1. **- * float *ptr = new float; X = *ptr;**
2. - float & ptr = new float; X = & ptr;
3. - float * ptr = &X; X = new float;

12. Полиморфизм в объектно-ориентированном программировании реализуется:
1. **- * через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны**
2. - через механизмы перегрузки (функций и операций) и шаблоны;

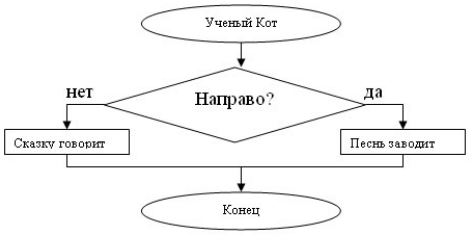
3. - через виртуальные функции и шаблоны;
4. - через механизмы перегрузки (функций и операций) и виртуальные функции

Тесты для проверки знаний ко второй промежуточной аттестации: экзамену

1. Как выглядит знак присваивания в программировании?
1. =>
2. =
3. :=
4. <= :

2.Как называется алгоритмическая конструкция, которая состоит из последовательных действий, В строго упорядоченном порядке друг за другом следующих?
1. Следование
2.Ветвление
3. Повторение

3. Какой алгоритм изображен на картинке?:



1. Следование
2. Ветвление
3. Повторение

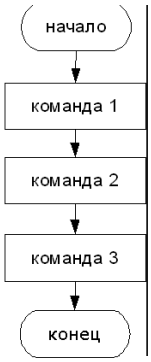
4.Выберите правильную запись программы :
1. programm Yakov_petrovich;
2. program Yakov petrovich; program Yakov Petrovich;
3. program_Yakov_Petrovich;

5.С помощью какой команды можно узнать остаток деления числа a на b:
1.read
2.div
3.var
4.mod

6.Запишите значение переменной b после выполнения фрагмента алгоритма:
```
 a:=4;
 x:=10;
a:= x-a*2;
x:=a*2-x;
```
Ответ: 14

7.Какая запись верна:
write(Задайте переменные a и b);
write('Задайте переменные a и b');
write(' Задайте переменные ', a и b);
write(Задайте переменные , 'a и b');

8.На схеме алгоритм изображена алгоритмическая конструкция:



1. **Линейного алгоритма**
2. Алгоритма ветвления
3. Алгоритма повторения

9. Какая запись верна:
var a b c integer;
var a. b. c :integer;
var a, b, c integer;
var a, b, c :integer;

10.При использовании в программе оператора scanf(«%d%d%d»,&a,&b,&c)требуется разделять числовые значения величин при вводе при помощи

1. запятой
2. **одного пробела**
3. **нажатия клавиши ENTER**
4. **произвольного количества пробелов**

11.Укажите основных два типа данных в языке С++

a. Целые и плавающие типы данных
b. Дробные и вещественные типы данных
c. Побитовые и целые типы данных
d. Целые и беззнаковые типы данных
e. Плавающие и вещественные типы данных

12. К целым типам данных в С++ относятся …

a. Char, long, int, short
b. Char, float, int, short
c. Double, long, int, short
d. Double, float, int, short
e. Char, long, int, double

13.К целым типам данных в C++ не относится…

- **a. float**
- **b. char**
- c. int
- d. long
- e. short

14. К целым типам данных в C++ не относится…

- a. double
- b. char
- **c. int**
- **d. long**
- **e. short**

15. В C++ к плавающим типам данных относятся…

- **a. Double, float**
- b. Char, float
- c. Float, int
- **d. Double,short**
- e. Char, int

16. Какое из следующих утверждений неверное?

- a. для диапазона 0…265 лучше всего подходит тип unsigned char
- **b. для диапазона -128…127 лучше всего подходит тип char**
- c. для диапазона 0…255 лучше всего подходит тип unsigned char
- d. для диапазона -32768…32767 лучше всего подходит тип short
- e. значение 32000 входит в тип short

17. Какое из следующих утверждений верно?

- a. для диапазона -128…127 лучше всего подходит тип char
- b. для диапазона 0…265 лучше всего подходит тип unsigned char
- **c. для диапазона 0…255 лучше всего подходит тип char**
- d. для диапазона -32768…32767 лучше всего подходит тип unsigned char
- e. значение 32000 входит в тип char

18.Какой тип не подходит для данных диапазона 0…255?

- a. char
- **b. int**
- c. short
- **d. long**
- **e. unsigned char**

19.Какой тип не подходит для данных диапазона -128…127?

- f. unsigned char
- g. char
- **h. int**
- **i. short**

j. long

20. К переменным какого типа можно применить операцию %
    a. целого
    b. вещественного
    c. логического
    d. символьного
    e. любого типа

21. Какой диапазон значений имеет тип данных char?
a. -128…127
b. -32768…32767
c. 0…255
d. 0…65535
e. 3.14E-38 … 3.14E+38.

22. Какая из операций есть логическое или
    a. ||
    b. &&
    c. !=
    d. ==
    e. <<=

23. Знаком || обозначается в С++ логическая операция
a. или
b. и
c. равно
d. не равно
e. если

Блок первый « Основы алгоритмизации и программирование»
1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Перечень тем рефератов:

1. Синтаксис и семантика универсального языка программирования высокого уровня.
2. Структура программы.
3. Описание данных, константы и переменные. Типы переменных. Выражения.
4. Оператор присваивания. Процедуры ввода-вывода.
5. Построение вычислительных программ линейной структуры.
6. Основные и дополнительные структурные конструкции управления процессом вычислений и их реализация операторами языка: условной передачи управления, выбора, конструкции циклов.
7. Организация программ разветвленной и циклической структуры на примере решения задач вычислительной математики: приближенное вычисление корня функции, приближенное вычисление суммы сходящегося бесконечного ряда и др

Блок второй «Структурные типы данных Основы С++. Массивы, Строки, Указатели и Ссылки в С++. Функции в С++»
3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.

Темы рефератов:

1. Структурные типы данных и модульное программирование.
2. Создание консольных приложений. Создание схем алгоритмов средствами Microsoft Visio и OpenOffice Draw.
3. Структурные типы данных: массивы, строки и записи (структуры).
4. Программирование с использованием структурных типов данных.
5. Представление обработки массивов, матриц и текстов.

Блок третий «Директивы С++ . Введение в отношения между объектами в С++.»
1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Объектно-ориентированное программирование;
2. Языки интернет-программирования;
3. Базы данных;
4. Технология разработки программных систем.

Блок четвертый «Переменные и основные типы данных в С++. Виртуальные функции в С++»

3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.

Темы рефератов:

1. Понятие Строки
2. Процедуры и функции
3. Подпрограммы. Средства отладки С++
4. Файловая система
5. Создание модулей.
6. Процедурный тип параметров

Блок пятый «Операторы в С++. Шаблоны в С++»
1 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Процедуры и функции.
2. Формальные и фактические параметры.
3. Передача параметров по значению и ссылке.
4. Рекурсия.
5. Модули.
6. Файловая система.
7. Файлы.

Блок шестой «Область видимости и другие типы переменных в С++. Стандартная библиотека шаблонов (STL) в С++. Ввод/Вывод в С++.»
3 контрольная точка: Вид контрольного задания – защита практических работ, реферат.
Темы рефератов:

1. Типизированные и текстовые файлы. Создание односвязных списков по типу очереди и по типу стека.
2. Удаление элементов из списков и добавление элементов в них.
3. Простые классы и композиция. Наследование.

4. Наполнение.
5. Полиморфное наследование.
6. Создание однооконных приложений Windows.
7. Создание MDI многооконных приложений Windows.

Ситуационные задачи для проверки умений и навыков к экзаменам

Задания к первому экзамену

Задание 1. IT-собеседований по языку C++

Часто работодатели пытаются ввести нас в заблуждение сложными вопросами и логическими задачами. Это своего рода мозговой штурм, чтобы проверить кандидата не только на умение правильно решать сложные задачи, но и на время, за которое он приходит к правильным ответам.

Чтобы не ударить лицом в грязь, просмотрите нашу подборку из 15 нелегких вопросов с IT-собеседований по C++: она вам обязательно пригодится.

1. Что мы получим на выходе, исходя из условия?

```
#include<iostream>
using namespace std;

f();
int x = 9;
void main()
{
    f();
    cout << x;
}

f()
{
   ::x = 8;
}
```

Ответ: 8.

Думаю, объяснения излишни, но на всякий случай: «чтение» кода в плюсах происходит строго сверху вниз. Так что как в Java, где можно ниже мейна объявлять сколь угодно методов, а они все равно будут воспроизводиться в указанном порядке, сделать не получится.

Задание 2.

2. Что будет выведено и почему?

```
#include <iostream>

int main(int argc, char **argv)
{
    std::cout << 25u - 50;
    return 0;
}
```

Ответ не -25, не надейтесь.

Ответ (который удивит многих): 4294967271, предполагая 32-битные целые числа.

Почему так происходит?

Существует иерархия: long double, double, float, unsigned long int, long int, unsigned int, int. И когда два операнда определены как **25u** (unsigned int) и **50** (int), **50** также будет интерпретироваться как беззнаковое целое число, то есть **50u**.

Кроме того, результат операции также будет иметь тип операндов. Следовательно, результат **25u — 50u** и сам является беззнаковым целым числом. Таким образом,

результат **-25** преобразуется в 4294967271.

Задание 3. Когда используется виртуальное наследование?

Распространенный вопрос с IT-собеседований. Когда есть класс (класс A), который наследуется от 2 родителей (B и C), оба из которых разделяют родителя (класс D):

```cpp
#include <iostream>

class D {
public:
    void foo() {
        std::cout << "Fooooooo" << std::endl;
    }
};


class C:  public D {
};

class B:  public D {
};

class A: public B, public C {
};

int main(int argc, const char * argv[]) {
    A a;
    a.foo();
}
```

Если вы не используете виртуальное наследование, то получите две копии D в классе A: один из B и один из C. Чтобы исправить это, вам нужно изменить объявления классов C и B следующим образом:

```cpp
class C:  virtual public D {
};

class B:  virtual public D {
};
```

Задание 4. Что вообще означает модификатор virtual?

В C++ виртуальные функции позволяют поддерживать полиморфизм – одну из ключевых составляющих ООП. С его помощью в классах-потомках можно переопределять функции класса-родителя. Без виртуальной функции мы получаем «раннее связывание», а с ней – «позднюю привязку». То есть, какая реализация метода используется, определяется непосредственно во время выполнения и основывается на типе объекта с указателем на объект, из которого он построен.

Пример. Приведите пример использования виртуальной функции.

У нас есть 2 класса:

```cpp
class Animal
{
    public:
        void eat() { std::cout << "I'm eating generic food."; }
};

class Cat : public Animal
{
    public:
        void eat() { std::cout << "I'm eating a rat."; }
};
```

В основной функции:

```cpp
Animal *animal = new Animal;
Cat *cat = new Cat;

animal->eat(); // Outputs: "I'm eating generic food."
cat->eat();    // Outputs: "I'm eating a rat."
```

Теперь сделаем так, что **eat()** будет вызываться посредством какой-нибудь промежуточной функции:

```cpp
void func(Animal *xyz) { xyz->eat(); }
```

В основной функции:

```
Animal *animal = new Animal;
Cat *cat = new Cat;

func(animal); // Outputs: "I'm eating generic food."
func(cat);    // Outputs: "I'm eating generic food."
```

Как это исправить, если мы захотим добавить больше животных? Просто делаем **eat()**виртуальной функцией:

Теперь в основной функции:

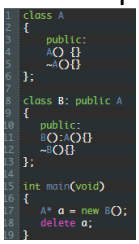
```
func(animal); // Outputs: "I'm eating generic food."
func(cat);    // Outputs: "I'm eating a rat."
```

Все работает!

Задание 6. Есть ли разница между классом и структурой?

Единственное различие между классом и структурой – это модификаторы доступа. Элементы структуры являются общедоступными по умолчанию, а класса – private. Рекомендуется использовать классы, когда вам нужен объект с методами, а в случае с простым объектом – структуры.

В чем проблема следующего фрагмента?



```
class A
{
    public:
    A() {}
    ~A() {}
};

class B: public A
{
    public:
    B():A() {}
    ~B() {}
};

int main(void)
{
    A* a = new B();
    delete a;
}
```

Если статический тип подлежащего удалению объекта отличается от его динамического типа, статический тип должен быть базовым классом динамического типа подлежащего удалению объекта и иметь виртуальный деструктор или поведение **undefined**.

Задание 8. Что такое класс хранения?

Класс, который определяет срок существования, компоновку и расположение переменных/функций в памяти.

В C ++ поддерживаются такие классы хранения: auto, static, register, extern и mutable.

Обратите внимание, что **register** устарел для C++11. Для C++17 он был удален и зарезервирован для будущего использования.

Задание 9. Как вызвать функцию C в программе на C++?

Еще один популярный вопрос с IT-собеседований, рассчитанный на новичков, совершенно не представляющих, как такое возможно. На самом же деле возможно, если использовать **extern «C»**:

```c
//C code
void func(int i)
{
//code
}

void print(int i)
{
//code
}
```

```cpp
//C++ code
extern "C"{
void func(int i);
void print(int i);
}

void myfunc(int i)
{
   func(i);
   print(i);
}
```

Что делает ключевое слово const?

Ответ: задает константность объекта, указателя, а также указывает, что данный метод сохраняет состояние объекта (не модифицирует члены класса).

Пример с неизменяемыми членами класса:

```cpp
class Foo
{
private:
    int i;
public:
    void func() const
    {
        i = 1; // error C3490: 'i' cannot be modified because it is being accessed through a
    }
};
```

Задание 11. Виртуальный деструктор: что он собой представляет?

Во-первых, он объявляется как **virtual** (об этом модификаторе мы писали выше). Он нужен, чтобы с удалением указателя на какой-нибудь объект был вызван деструктор данного объекта. Например, у нас есть 2 класса:

```cpp
class base
{
   public:
    ~base()
    {
      cout << "Вызывается деструктор класса base";
    }
};

class derived: public base
{
  public:
   ~derived()
   {
     cout << "Вызывается деструктор класса derived";
   }
};
```

Выполняем следующее:

```cpp
base *p;  //объявляем указатель на base
derived d_ob;
p=new derived();
delete p;
return 0;
```

В итоге выполнится деструктор базового класса, а не производного. Это может поспособствовать утечке памяти. Если же до объявления деструкторов установить модификатор **virtual**, выполнится деструктор производного класса.

Задание 12. Виртуальный конструктор: что он собой представляет?

Каверзный вопрос с IT-собеседований, который чаще всего задают именно после виртуальных деструкторов, дабы сбить кандидата с толку. Конструктор не может быть виртуальным, поскольку в нем нет никакого смысла: при создании объектов нет такой неоднозначности, как при их удалении.

Сколько раз будет выполняться этот цикл?

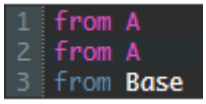
```cpp
unsigned char half_limit = 150;

for (unsigned char i = 0; i < 2 * half_limit; ++i)
{
    //что-то происходит;
}
```

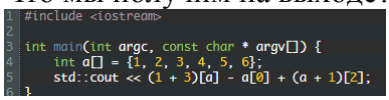
Еще один вопрос с подвохом с IT-собеседований. Если бы вы сказали 300, а **i** был объявлен как **int**, вы были бы правы. Но поскольку **i** объявлен как **unsigned char**, правильный ответ – зацикливание (бесконечный цикл).
Объясняем. Выражение **2 * half_limit** будет повышаться до **int** и заимеет значение 300. Но так как **i** – это **unsigned char**, он пересматривается по 8-битному значению, которое после достижения 255 будет переполняться, поэтому вернется к 0, и цикл будет продолжаться вечно.

Задание 14. Каков результат следующего кода?

Ответ:



Здесь важно отметить порядок уничтожения классов и то, как метод класса **Base** возвращается к своей реализации после удаления **A**.
Что мы получим на выходе?



Ответ: 8.
Объяснение:

- **(1 + 3)[a]** – то же, что и **a[1 + 3] == 5**
- **a[0] == 1**
- **(a + 1)[2]** – то же, что и **a[3] == 4**

Суть вопроса заключается в проверке арифметических знаний и понимании всей магии, которая происходит за квадратными скобками.

Задания ко второму экзамену

Задание 1. Консольные приложения C++ в среде Microsoft Studio 2008 (Visual C++).

Задание 2. Программирование ввода/вывода и выражений. Вычислить: y=(sin(a)-b)/(|b|+cos(b^2)).

Задание 3. Основные операторы передачи управления.

Дано натуральное K. Определить K цифру последовательности (номер K вводится с клавиатуры): 1101001000100001000001000000…. Символьные строки не использовать. На печать вывести фрагмент последовательности, содержащий искомую цифру.

Задание 4. Массивы.

Задание выдается на занятии преподавателем и выполняется с использованием конспектов лекций.

Задание 5. Использование указателей при работе со строками C++.

Дан текст. Слова в тексте разделены пробелами. Текст завершается точкой. Определить количество слов, в которые одновременно входят буквы **a** и **t**. Вывести на экран найденные слова. Пользуясь ***указателями,*** выдать на экран адреса этих слов.

Задание 6. Динамические структуры данных. Списки.

С клавиатуры вводится последовательность символов s_1, s_2, s_3 ...s_n, где n заранее неизвестно и определяется при вводе. Сформировать из введенных символов список и распечатать его. Задано число ***m*** (***m<n***). Используя список, сформировать из символов списка две строки, в которых символы последовательности размещены в следующем порядке: s_m, s_{m+1}, s_{m+2} и s_n, s_1, s_2,..., s_{m-1}.

Из полученных строк удалить символы “а” и “м” или выдать сообщение об их отсутствии. Вывести на печать сформированные строки до и после корректировки.

Задание 7. **Простые объекты.**

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов.

Объект - предложение. Параметры: массив слов (n<10) и их количество. Методы: инициализирующий, выводящий данные об объекте и метод, определяющий количество слов, начинающихся с согласных букв.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Задание 8. **Простые объекты.**

Описать класс, включающий заданные поля и методы, двумя способами: без конструктора и с конструктором. Написать тестирующие программы, создающие массив объектов. Объект - слово. Параметры: слово и его длина. Методы: инициализирующий, выводящий на экран данные об объекте и метод, определяющий количество заданных букв в слове.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Задание 9. Наследование.

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Объект – Прямоугольник, характеризующийся размерами. Объект умеет выводить на экран значения своих полей и отвечать на запрос о площади.

Объект – Прямоугольный параллелепипед, характеризующийся размерами. Объект умеет выводить на экран содержимое своих полей, возвращать по запросу их содержимое и определять объем параллелепипеда.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Задание 10. **Наследование.**

Разработать и реализовать иерархию классов для описанных объектов предметной области, используя механизмы наследования. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Объект – книга. Параметры: автор, название и количество экземпляров. Методы: инициализирующий и функции, возвращающие значения параметров.

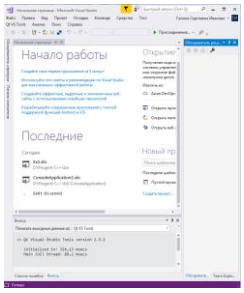
Объект – книга. Параметры: автор, название, количество книг и количество желающих ее прочитать читателей. Методы: конструктор и процедура определения средней длины очереди на чтение экземпляра.

В отчете привести диаграмму разработанных классов и объектную декомпозицию.

Задание 7 Практические задания

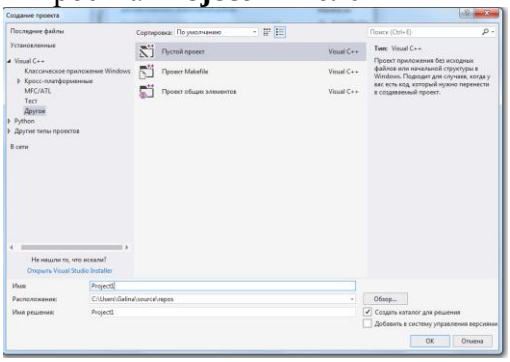
Создание заготовки консольного приложения

Главное окно среды при запуске имеет вид, представленный на рисунке 1. Окно включает меню, панели инструментов, текстовый редактор со стартовой страницей и окно навигатора Обозреватель решений. Причем последнее включает несколько вкладок, позволяющих просматривать различную информацию. Вид главного окна среды при отсутствии открытого проекта



Создание заготовки консольного приложения выполняется либо нажатием кнопки **Создать проект...** стартовой страницы (справа внизу), либо с использованием меню **Файл/Создать/Проект...**.

В появившемся окне **Создание проекта** выбираем тип проекта **Visual C++** и шаблон **Пустой проект**. Вводим имя проекта **Project1** в поле **Имя**



Вид окна выбора типа проекта и задания его имени и местоположения

При желании можно указать отличное от стандартного местонахождение папки будущего проекта в позиции **Расположение** и отличное от имени проекта имя программы **Имя решения**, но особой необходимости в этом нет.

Также можно выбрать и **Классическое приложение Windows,** и шаблон **Консольное приложение**. В этом случае создается приложение «Hello, Windows», которое надо исправлять под свою программу.

По умолчанию Visual Studio вместе с файлами проекта создает файл . pdb. Этот файл содержит информацию для работы отладчика. Например, он содержит имя файла и номер строки оператора, которые включаются в сообщение об ошибке при возникновении

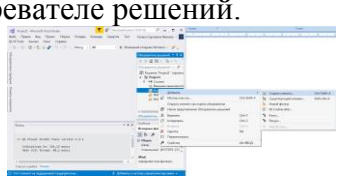
исключений. Формирование этого файла можно исключить, для этого следует сбросить опцию:

Проект->Свойства->Компоновщик->Отладка->Создавать отладочную информацию: Нет.

Для предотвращения закрытия окна консоли вместо ввода в конце программы, как это было в Delphi, устанавливаем опцию:

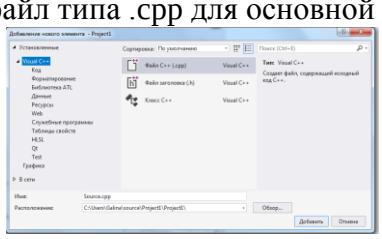
Проект->Свойства->Компоновщик->Система->Подсистема: Консоль.

Пустой проект. Если вы создали пустой проект, то к нему необходимо добавить файлы, содержащие исходный код программы. Для этого можно использовать пункт меню **Проект/Добавить новый элемент** или один раз щелкнуть правой кнопкой мышки по пункту Исходные файлы в Обозревателе решений.



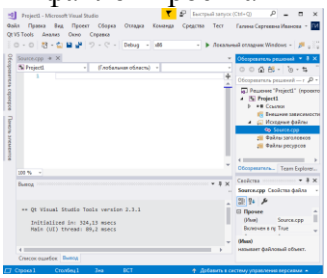
Добавление файлов к проекту

Тип добавляемого файла выбирается в специальном окне (рисунок 5). Первым добавляем к пустому проекту файл типа .cpp для основной программы.



Выбор типа добавляемого файла

Чтобы ввести текст программы необходимо открыть соответствующий файл в окне текстового редактора. Для этого переходим на вкладку **Обозреватель решений** и щелкаем правой кнопкой мыши по добавленному файлу. Если пункт Исходные файлы «закрыт», то его предварительно надо «раскрыть», щелкнув по треугольнику слева от пункта, раскрывая список исходных файлов проекта



Переход в окно текстового редактора для ввода программы

Консольное приложение Windows. При выборе консольного приложения Visual Studio высвечивает в окне редактора приложение «Hello, Windows» и некоторую дополнительную информацию:

```
// ConsoleApplication1.cpp: Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.
//

#include "pch.h"
#include <iostream>
```

```
int main()
{
   std::cout << "Hello World!\n";
}

// Запуск программы: CTRL+F5 или меню "Отладка" > "Запуск без отладки"
// Отладка программы: F5 или меню "Отладка" > "Запустить отладку"

// Советы по началу работы
//   1. В окне обозревателя решений можно добавлять файлы и управлять ими.
//   2. В окне Team Explorer можно подключиться к системе управления версиями.
//   3. В окне "Выходные данные" можно просматривать выходные данные сборки и другие сообщения.
//   4. В окне "Список ошибок" можно просматривать ошибки.
//   5. Последовательно выберите пункты меню "Проект" > "Добавить новый элемент", чтобы создать файлы кода, или "Проект" > "Добавить существующий элемент", чтобы добавить в проект существующие файлы кода.
//   6. Чтобы снова открыть этот проект позже, выберите пункты меню "Файл" > "Открыть" > "Проект" и выберите SLN-файл.
```

Файл pch.h содержит предварительно откомпилированные заголовки (англ. *pre-compiled headers*). Это способ ускорить компиляцию программ за счёт предварительной обработки заголовочных файлов (h-файлов), которые содержат интерфейсы модулей и подключаются к программе путём прямой вставки их текстов в тело основной программы с помощью директивы `#include`. Важно помнить, что директива **`#include pch.h`** должна быть включена *во все*(!) исходные файлы проекта с расширением cpp.

Однако использование **pch**-файлов не обязательно. Его можно отменить, сбросив опцию:

Проект->Свойства->Си/С++->Предварительно откомпилированные заголовки->Предварительно откомпилированный заголовок: Не использовать.

Visual C++ 2017 имеет еще одну особенность: Компилятор настроен на проверку безопасности строковых функций, т.е. он разрешает использовать только функции с постфиксом «**_s**». Для отмены проверки можно отключить опцию контроля:

Проект->Свойства->Си/С++->Создание кода->Проверка безопасности: Отключить проверку безопасности (/GS-).

Ввод программы

В качестве примера введем в окно редактора программу вычисления наибольшего общего делителя двух целых чисел. Причем само вычисление выделим в функцию **nod**, оставив в основной программе ввод данных и вывод результата.

```
#include "pch.h"  // только при создании проекта с заголовочными файлами (!)
#include <locale.h>  // для подключения русского языка
#include <stdio.h>   // подключение процедур ввода вывода
int nod(int x, int y)
{
     while (x!=y)
          if (x>y) x=x-y;
          else y=y-x;
     return y;
}
```

```
void main()
{
      int a,b;
      setlocale(0,"russian");  // подключение русского языка
      puts("Введите два натуральных числа:");
      scanf_s("%d %d",&a,&b);
      printf("НОД %d и %d = %d.\n",a,b,nod(a,b));
}
```

Программа специально настроена на использование при выводе русского языка. Для этого перед программой подключена библиотека **locale.h**, а в самой программе вызвана процедура назначения необходимой кодировки. Консольная программа с такими настройками может выводить (но не вводить!) русские тексты.

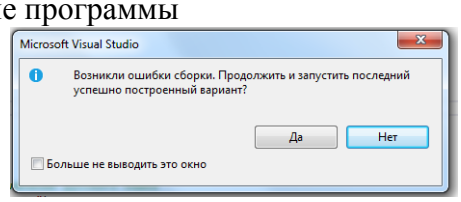
Запуск программы на выполнение

Для запуска программы на выполнение используют команды подменю пункта главного меню **Отладка** или клавиши «быстрого доступа», показанные в скобках:

- **Начать отладку (F5)** – компиляция, компоновка и выполнение с отладкой – позволяет в случае выдачи сообщения об ошибке выполнения просмотреть содержимое интересующих программиста переменных (см. раздел 6)**;**
- **Запуск без отладки (Ctrl +F5)** – компиляция, компоновка и выполнение программы без отладки**.**

В обоих случаях, если готовая программа не существует, среда выдаст соответствующих запрос на выполнение компиляции и компоновки программы перед запуском.

Если при компиляции или компоновке обнаружены ошибки, то среда запрашивает, продолжить или нет создание программы

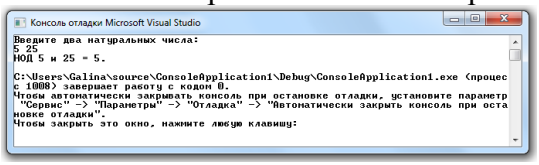


Запрос на запуск на выполнение программы при наличии ошибок

При отрицательном ответе на запрос среда высвечивает сообщения об ошибках в специальном окне **Вывод**, которое появляется в нижней части окна текстового редактора. При положительном – пытается запустить программу с ошибками, что в большинстве случаев обречено на провал.

Для перехода на строку, при обработке которой зафиксирована ошибка, необходимо дважды щелкнуть мышью по соответствующему сообщению. После исправления ошибок программу необходимо вновь скомпилировать и скомпоновать.

Если компиляция и компоновка программы прошли без ошибок, то на экране появляется функционирующее в режиме консоли окно результатов, в которое выводятся сообщения программы и «эхо» вводимых символов Окно результата закрывается по нажатию любой клавиши или кнопки закрытия окна «×» в правом верхнем углу.



Окно результата (изменены цвета фона и символов)

Если ошибка обнаружена уже при выполнении программы, а именно:

- программа выдает сообщение об ошибке выполнения,
- результаты программы отличны от ожидаемых,
- программа «зациклилась»,

то для поиска ошибки следует использовать средства отладки среды Visual C++.

При зацикливании программы ее выполнение можно завершить, используя кнопку закрытия окна «×» в правом верхнем углу окна результата.

Модульное программирование. Файлы заголовков

Среда Microsoft Visual Studio позволяет создавать и отлаживать программы, использующие не только стандартные, но и пользовательские библиотеки (модули).

Модуль языка C++ в отличие от модуля языка Pascal обычно включает два файла: заголовочный файл с расширением **.h** и файл реализации с расширением **.cpp**.

Заголовочный файл играет роль своеобразной интерфейсной секции модуля. В него помещают объявление экспортируемых ресурсов модуля: прототипы (заголовки) процедур и функций, объявления переменных, типов и констант. Этот файл подключают директивой **#include "<Имя модуля>.h"** к файлу реализации модуля, программы или другого модуля, если они используют ресурсы описываемого модуля.

Файл реализации представляет собой аналог секции реализации модуля Pascal. Он должен содержать директивы подключения заголовочных файлов используемых модулей, описания экспортируемых процедур и функций, а также объявления и описания внутренних ресурсов модуля. Если используется проект с прекомпилируемым заголовком, то в начало каждого файла реализации необходимо поместить оператор подключения заголовочного файла pch.h: **#include "pch.h"**. Этот файл осуществляет подсоединение библиотек среды, и при его отсутствии компилятор выдает ошибку «Не найден конец файла».

Для создания файлов заголовка и реализации модуля и *добавления их к проекту* используют пункт меню **Проект/Добавить новый элемент**. Выполнение этого пункта также как и добавление файла через вызов контекстного меню щелчком правой кнопки мыши на пункте **Файлы заголовков** Обозревателя решений вызовет открытие окна добавления файлов различных типов к проекту (рисунок 5).

В этом окне необходимо выбрать и тип подключаемого файла и ввести его имя. По данной схеме создадим два файла, образующих модуль:

- файл заголовка Nod.h – **Файл заголовка (.h)**;
- файл реализации Nod.cpp – **Файл C++ (.cpp)**.

Если используется проект с прекомпилированным файлом, то в начало файла Nod.cpp добавим строку подключения файла **pch.h:**

#include "pch.h" .

Затем перенесем в этот файл текст функции **nod**, вырезав его из файла программы, и добавим строку подключения заголовочного файла модуля. В результате файл Nod.cpp должен содержать:

```
#include "pch.h"     // только для проекта с прекомпилированным файлом
#include "Nod.h"
int nod(int x, int y)
{
     while (x!=y)
          if (x>y) x=x-y;
          else y=y-x;
          return y;
}
```

В заголовочном файле **Nod.h** объявим внешний ресурс – функцию **nod()**:

```
int nod(int x, int y);
```

Вместо функции **nod()** в файл основной программы добавим ссылку на заголовочный файл **Nod.h**:

```
#include "Nod.h"
```

Для навигации по многофайловому проекту используют вкладку **Решение** обозревателя, на которой высвечен список всех исходных файлов проекта.

Примечание. Модуль не всегда включает два файла. Возможно создание модулей, состоящих из одного файла заголовка или файла реализации. Если файл реализации пуст, то он не создается. В программе это никак не отражается. Если не используется файл заголовка, то в проекте или других модулях, обращающихся к ресурсам этого модуля, объявленным в файле реализации, указывается непосредственно подключение файла реализации. Однако в последнем случае нарушается принцип инкапсуляции модулей, что *нетехнологично*.

Если необходимо подключить модуль, файлы которого размещены в других каталогах и не могут быть скопированы в каталог текущего проекта, то в операторе **include** указывают полное имя файла, например:

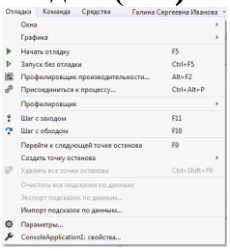
```
#include "C:\primer.cpp\con2\filemod.h"
```

Файл реализации модуля с расширением **.cpp** при этом обязательно должен быть скопирован в папку проекта.

Для удаления файла *из проекта* необходимо выделить этот файл на вкладке **Решение** обозревателя и нажать на клавиатуре **Delete**.

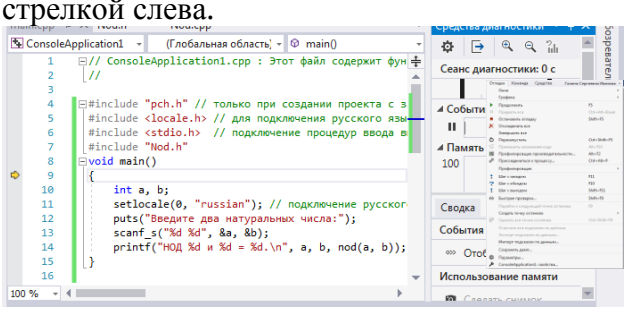
Отладка консольных приложений

Для входа в режим пошагового выполнения используют команды **Отладка/Шаг с заходом (F11)** или **Отладка/Шаг с обходом (F10)**



Подменю отладки

После запуска программы в режиме отладки на экране появляется окно с текстом основной программы, в котором следующий выполняемый оператор отмечен специальной стрелкой слева.



Окно основной программы при запуске в режиме отладки

При входе в режим отладки содержимое пункта меню **Debug** изменяется (рисунок 10). В нем появляются следующие основные команды управления режимом:

Остановить отладку (Shift+F5);

Перезапустить (Ctrl+Shift+F5);

Шаг с заходом (F11) – если на данном шаге осуществляется вызов процедуры или функции, то зайти в них;

Шаг с обходом (F10) – если на данном шаге есть вызов подпрограммы, то не заходить в нее;

Шаг с выходом (Shift+F10) – выполнить процедуру или функцию до конца и вернуться в вызывающую функцию.

После выхода из режима отладки меню среды принимает исходный вид.

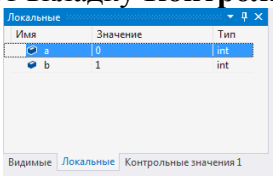
Отладку программы можно выполнять в пошаговом режиме, просматривая изменяющиеся значения переменных, или устанавливая точки останова в ключевых точках программы.

В процессе пошагового выполнения многофайловых проектов переключение между файлами, содержащими исходные тексты программы и модулей, происходит автоматически.

Просмотр значений переменных в режиме отладки

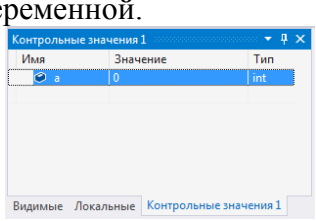
Для просмотра значений переменных в процессе отладки открывают вкладки окна **Вывод**, появляющегося в нижней части окна среды при запуске программы на выполнение: вкладки **Видимые**, **Локальные, Контрольные значения 1.**

На вкладке **Локальные** высвечиваются значения всех переменных выполняемой функции. По мере выполнения операторов программы значения этих переменных меняются. Однако локальных переменных в сложных функциях много, отслеживать их трудно, поэтому обычно используют вкладку **Контрольное значение 1.**



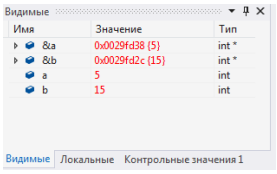
Окно Locals

На вкладке **Контрольное значение 1** высвечиваются значения только переменных, указанных программистом. Добавить переменную в окно **Контрольное значение 1** можно, введя ее имя в поле **Имя**. Также непосредственно в окне, можно удалить идентификатор отслеживаемой переменной.



Окно Контрольное значение 1

Очень удобно при пошаговом выполнении также окно **Видимые**. В нем высвечиваются значения переменных текущего и следующего оператора программы. Особенно это удобно, если отлаживаемый фрагмент сложен и использует большое количество переменных.

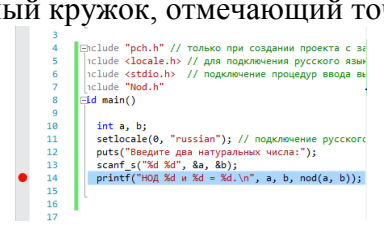


Окно Видимые

Установка и отмена точек останова в программе

Часто при отладке программу целесообразно выполнять пошагово не с самого начала, а с некоторого места, подозрительного с точки зрения наличия ошибки. Для останова программы в этом месте используют точки останова.

Задать точку останова в программе проще всего, щелкнув мышкой по серому полю текстового редактора перед интересующим нас оператором. Это можно сделать в процессе пошагового выполнения программы или перед ее запуском. При таком щелчке на сером поле появляется красный кружок, отмечающий точки останова.



Отметка точки останова в программе

Теперь, если программу запустить на выполнение, то она автоматически остановится перед выполнением отмеченного оператора.

Вопросы к экзамену:

1. Модель памяти и структура программы. Классы памяти. Ссылки.
2. Средства абстракции C++. Структура класса. Статические члены и их инициализация
3. Средства инкапсуляции C++. Инкапсуляция и наследование.
4. Модульность, раздельная компиляция, пространства имен, using директива.
5. Представление иерархических отношений. Наследование.
6. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.
7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм.
8. C++: средства реализации состояния объектов; реализация поведения.
9. Перегрузка операторов.
10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании.
11. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов.
12. Исключения в C++. Обработка исключений. Умные указатели.
13. Шаблоны классов и шаблоны функций. Специализация.
14. Основы STL. Структура и назначение. Контейнеры.
15. Объект с точки зрения ООП. Состояние. Поведение.
16. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов.
17. Объект с точки зрения ООП. Взаимоотношения между объектами.
18. Классы. Природа классов. Метамодель. Инстанцирование.
19. Классы. Структура класса. Абстрактные классы и интерфейсы.
20. Классы. Принцип разделения интерфейсов.
21. Классы. Средства UML для построения диаграмм классов.
22. Классы. Отношения между классами. Ассоциация и агрегация.
23. Классы. Иерархии классов. Зависимость.

7.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

ПРАКТИЧЕСКАЯ РАБОТА №1
Вид практического занятия: Практическая работа.
Тема и содержание занятия Программирование ввода/вывода и выражений
Цель занятия:

1. Ознакомиться с принципом работы ввода/вывода.
2. Разработать код программы.
3. Привести результаты тестирования.
4. Описать схему алгоритм.

По результатам выполненной работы написать отчет и сделать общий вывод.
Продолжительность занятия – 10 часов (2 для заочной формы обучения).
ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)
Продолжительность занятия – 10 часов (1 час для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)
Продолжительность занятия – 7 часов (1 час для заочной формы обучения).

ПРАКТИЧЕСКАЯ РАБОТА № 2
Вид практического занятия: Практическая работа.
Тема и содержание занятия Основные операторы передачи управления.
Цель занятия:
1. Научиться реализовать конструкцию оператора «если…»
Цель занятия:

1. Ознакомиться с принципом работы ввода/вывода.
2. Разработать код программы.
3. Привести результаты тестирования.
4. Описать схему алгоритм.

По результатам выполненной работы написать отчет и сделать общий вывод.
Продолжительность занятия – 10 часов (2 для заочной формы обучения).

ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)
Продолжительность занятия – 10 часов (1 час для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)
Продолжительность занятия – 7 часов (1 час для заочной формы обучения).

ПРАКТИЧЕСКАЯ РАБОТА № 3
Вид практического занятия: Практическая работа.
Тема и содержание занятия Функции
Цель занятия:

1. Ознакомиться с принципом работы подпрограммы функции
2. Разработать код программы.
3. Привести результаты тестирования.
4. Описать схему алгоритм.

По результатам выполненной работы написать отчет и сделать общий вывод.

Продолжительность занятия – 5 часов (1 час для заочной формы обучения).

ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)
Продолжительность занятия – 5 часов (0,5 часа для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)
Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

ПРАКТИЧЕСКАЯ РАБОТА № 4
Вид практического занятия: Практическая работа.
Тема и содержание занятия. Текстовая обработка
Цель занятия:

Ознакомиться с принципом работы программы строк и вывести соответствующее сообщение.

1. Разработать код программы.
2. Привести результаты тестирования.
3. Описать схему алгоритм.

По результатам выполненной работы написать отчет и сделать общий вывод.
Продолжительность занятия – 5 часов (1 час для заочной формы обучения).

ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)
Продолжительность занятия – 5 часов (0,5 часа для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)
Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

ПРАКТИЧЕСКАЯ РАБОТА № 5
Вид практического занятия: Практическая работа.
Тема и содержание занятия Файлы
Цель занятия: *работа с файлами (создание, открытие, закрытие)*

1. Разработать код программы.
2. Привести результаты тестирования.
3. Описать схему алгоритм.

По результатам выполненной работы написать отчет
Продолжительность занятия – 5 часов (1 час для заочной формы обучения)

ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)
Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)
Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

ПРАКТИЧЕСКАЯ РАБОТА № 6
Вид практического занятия: Практическая работа.
Тема и содержание занятия Композиция**.** Основы програмрования Python.

Цель занятия:

Разработать и реализовать диаграмму классов для описанных объектов предметной области, используя механизмы наследования и композиции. Проверить ее на тестовом примере, с демонстрацией всех возможностей разработанных классов на конкретных данных.

Даны:

Объект «Символ», умеющий печатать свое значение и отвечать на запрос о значении своего символьного поля.

Объект «Запись», содержащий 2 поля: символ и строка символов, как массив из 15 символов, умеющий выводить на печать значение своих полей, удалять символ строки по номеру, удалять символ строки по значению.

По результатам выполненной работы в отчете привести диаграмму разработанных классов и объектную декомпозицию.

Продолжительность занятия – 5 часов (1 для заочной формы обучения)

ЗАЩИТА ПРАКТИЧЕСКОЙ РАБОТЫ (К.т.№1)

Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

ТЕСТИРОВАНИЕ (К.т.№2)

Продолжительность занятия – 4 часа (0,5 часа для заочной формы обучения).

8. Перечень основной и дополнительной учебной литературы; перечень ресурсов информационно-телекоммуникационной сети «Интернет», перечень информационных технологий, необходимых для освоения дисциплины

8.1. Основная литература

1. Программирование на СИ#: Учебное пособие / Медведев М.А., Медведев А.Н., - 2-е изд., стер. - М.:Флинта, Изд-во Урал. ун-та, 2017 http://znanium.com/catalog/product/948428
2. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015 http://znanium.com/catalog/product/505194
3. Прикладное программирование/АгафоновЕ.Д., ВащенкоГ.В. - Краснояр.: СФУ, 2015. http://znanium.com/catalog/product/550046
4.Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М. : ИД «ФОРУМ» : ИНФРА-М, 2018. - - Режим доступа: http://znanium.com/catalog/product/918098

8.2. Дополнительная литература

1. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев; под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА-М, 2019. - Режим доступа: http://znanium.com/catalog/product/1000008
2. Программирование графики на С++. Теория и примеры: учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М. : ИД «ФОРУМ» : ИНФРА-М, 2017 http://znanium.com/catalog/product/562914

8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Электронно-библиотечная система «Znanium.com»:http://znanium.com/
Информационная система «Единое окно доступа к образовательным ресурсам»:http://window.edu.ru/
Служба тематических толковых словарей «Глоссарий.ру»:http://www.glossary.ru/

Научная электронная библиотека «КиберЛенинка»:https://cyberleninka.ru/

8.4. Перечень программного обеспечения, современных профессиональных баз данных и информационных справочных системам

1. Microsoft Windows;
2. Microsoft Office;
3. Электронно-библиотечная система (ЭБС) «ZNANIUM.COM» Режим доступа: http://www.znanium.com/
4. Электронно-библиотечная система (ЭБС) «BOOK.ru».Режим доступа: https:// www.book.ru/
5. Реферативно-библиографическая и наукометрическая база данных Web of Science Режим доступа: http://webofscience.com/ Сублицензионный договор № WoS/919 от 02.04.2018.
6. База данных Scopus. Режим доступа: http://www.scopus.com/Сублицензионный договор № SCOPUS / 919 от 10.05.2018.
7. Федеральный перечень туристских объектов (профессиональная база анных)] Режим доступа: http://xn----7sba3acabbldhv3chawrl5bzn.xn--p1ai/ Доступ свободный
8. Инновационные территориальные кластеры Московской области (информационно-справочная система).Режим доступа: http://mii.mosreg.ru/deyatelnost/tehnicheskoe-regulirovanie/Доступ свободный
9. Инновации в России (информационно-справочная система).Режим доступа: http://innovation.gov.ru/taxonomy/term/382/Доступ свободный
10. Единый портал инноваций и уникальных изобретений (профессиональная база данных)Режим доступа: http://innovationportal.ru/inventions/Доступ свободный
11. Федеральный портал по научной и инновационной деятельности. Инновационные проекты (профессиональная база данных).Режим доступа: http://www.sci-innov.ru/catalog_tech/innov_project/Доступ свободный
12. Управление качеством в туризме (информационно-справочная система) Режим доступа: http://tourlib.net/quality.htm. Доступ свободный
13. QUALITY - Менеджмент качества и ISO 9000. Документы и материалы по менеджменту качества, стандартам ISO серии 9000 (профессиональная база данных).Режим доступа: http://quality.eup.ru/Доступ свободный
14. Федеральный информационный фонд стандартов (профессиональная база данных). Режим доступа: http://www.gostinfo.ru/pages/Maintask/fund/Доступ свободный.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Процесс изучения дисциплины предусматривает аудиторную (работа на лекциях и практических занятиях) и внеаудиторную (самоподготовка к лекциям и практическим занятиям) работу обучающегося.

В качестве основной методики обучения была выбрана методика, включающая - совокупность приёмов, с помощью которых происходит целенаправленно организованный, планомерно и систематически осуществляемый процесс овладения знаниями, умениями и навыками.

В качестве основных форм организации учебного процесса по дисциплине «Прикладное программное обеспечение» в предлагаемой методике обучения выступают

лекционные и практические занятия (с использованием интерактивных технологий обучения), а так же самостоятельная работа обучающихся.

- лекции

Лекция представляет собой устное изложение материала по определенной теме. Эта форма учебного процесса применяется при изложении объемного нового материала. Традиционная лекция состоит из трех частей: вступления, основной части и заключения. В первой части обозначается тема, план и цель лекции. В основной части лектор последовательно раскрывает все ключевые вопросы и приводит определение основных терминов. В заключении материал обобщается и суммируется.

Традиционная лекция с презентацией - это визуальная форма подачи лекционного материала. Лекция сводится к комментированию визуальных материалов.

- практические занятия

Практические занятия по дисциплине «Прикладное программное обеспечение» проводятся в форме выполнения практических работ с целью приобретения практических навыков в области конструкция отдельных элементов инженерных систем, обеспечивающих функционирование объектов недвижимости.

Практическая работа заключается в выполнении студентами, под руководством преподавателя, комплекса учебных заданий, направленных приобретение практических навыков и овладения методами практической работы с применением современных информационных и коммуникационных технологий. Выполнения **практической** работы студенты производят в письменном виде, в виде изучения конструкции и технических характеристик элементов инженерных систем. Отчет предоставляется преподавателю, ведущему данный предмет, в электронном и печатном виде.

Практические занятия способствуют более глубокому пониманию теоретического материала учебного курса, а также развитию, формированию и становлению различных уровней составляющих профессиональной компетентности студентов. Основой практикума выступают типовые задачи, которые должен уметь решать специалист в области сервиса.

При изучении дисциплины «Прикладное программное обеспечение» используются практические занятия в форме практических работ:

- самостоятельная работа обучающихся

Целью самостоятельной (внеаудиторной) работы обучающихся является обучение навыкам работы с научно-теоретической, периодической, научно-технической литературой и технической документацией, необходимыми для углубленного изучения дисциплины «Прикладное программное обеспечение» а также развитие у них устойчивых способностей к самостоятельному изучению и изложению полученной информации.

Основными задачами самостоятельной работы обучающихся являются:

- овладение фундаментальными знаниями;
- наработка профессиональных навыков;
- приобретение опыта творческой и исследовательской деятельности;
- развитие творческой инициативы, самостоятельности и ответственности студентов.

Самостоятельная работа студентов по дисциплине «Прикладное программное обеспечение» обеспечивает:

- закрепление знаний, полученных студентами в процессе лекционных и практических занятий;

- формирование навыков работы с периодической, научно-технической литературой и технической документаций;
- приобретение опыта творческой и исследовательской деятельности;
- развитие творческой инициативы, самостоятельности и ответственности студентов.

Самостоятельная работа является обязательной для каждого обучающегося.

Формы самостоятельной работы

Перечень тем самостоятельной работы студентов по подготовке к лекционным и практическим занятиям соответствует тематическому плану рабочей программы дисциплины.

Самостоятельная работа студента предусматривает следующие виды работ:

- Ознакомление с литературой по дисциплине на сайте ЭБС znanium.com.
- Самостоятельное изучение отдельных тем блока;
- Подготовка к практическим занятиям;

10. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю):

Учебные занятия по дисциплине «Прикладное программное обеспечение» проводятся в следующих оборудованных учебных кабинетах:

Вид учебных занятий по дисциплине	Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий с перечнем основного оборудования
Занятия лекционного типа, групповые и индивидуальные консультации, текущий контроль, промежуточная аттестация	учебная аудитория, специализированная учебная мебель ТСО: видеопроекционное оборудование/переносное видеопроекционное оборудование доска
Занятия семинарского типа	интерактивный компьютерный класс, специализированная учебная мебель ТСО: видеопроекционное оборудование, автоматизированные рабочие места студентов с возможностью выхода в информационно-телекоммуникационную сеть "Интернет" доска
Самостоятельная работа обучающихся	помещение для самостоятельной работы, специализированная учебная мебель, ТСО: видеопроекционное оборудование, автоматизированные рабочие места студентов с возможностью выхода в информационно-телекоммуникационную сеть "Интернет", доска; Помещение для самостоятельной работы в читальном зале Научно-технической библиотеки университета, специализированная учебная мебель автоматизированные рабочие места студентов с возможностью выхода информационно-телекоммуникационную сеть «Интернет», интерактивная доска