

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОСКОВСКИЙ
ГОСУДАРСТВЕННЫЙ СТРОИТЕЛЬНЫЙ УНИВЕРСИТЕТ**»

РАБОЧАЯ ПРОГРАММА

Шифр	Наименование дисциплины
Б1.Б.8	Объектно-ориентированное программирование

Код направления подготовки	09.03.02
Направление подготовки	Информатика и вычислительная техника
Наименование ОПОП	Системотехника и информационные технологии проектирования и управления в строительстве (Академический бакалавриат)
Год начала реализации ОПОП	2017
Уровень образования	Бакалавриат
Форма обучения	очная
Год разработки/обновления	2016

Разработчики:

должность	ученая степень, учёное звание	ФИО
ст. преподаватель		Садовский Б.С.


Рабочая программа рассмотрена и одобрена на заседании кафедры (структурного подразделения) «ИСТАС», Протокол № 9 от 29.09.2016

Заведующий кафедрой
(руководитель структурного подразделения)

 / Гинзбург А.В. /
Подпись, ФИО

Рабочая программа утверждена методической комиссией, Протокол № __1 от 17.10.2016

Председатель (зам. председателя)
методической комиссии

 / Кузина О.Н. /
Подпись, ФИО

Согласовано:

ЦОСП

_____ / _____ /
дата Подпись, ФИО

1. Цель освоения дисциплины

Целью освоения дисциплины «Объектно-ориентированное программирование» является формирование уровня освоения компетенций обучающегося в области техники программирования на алгоритмическом языке высокого уровня базирующихся на фундаментальных принципах построения программного продукта (ПП) и объектно-ориентированной парадигме программирования (ООП).

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования и с учётом рекомендаций примерной основной образовательной программы высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии» (уровень образования - бакалавриат).

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

Компетенция по ФГОС	Код компетенции по ФГОС	Основные показатели оценивания (показатели достижения результата)	Код показателя оценивания
Способность проектированию базовых прикладных информационных технологий	к и ПК-11	Знает методы структурного разбиения программы на части, описания алгоритмов в объектно-ориентированной парадигме, отладки и тестирования программного обеспечения (ПО)	З1
		Умеет моделировать различные объекты с помощью классов; работать с пользовательскими типами данных; использовать рациональные способы проектирования программного продукта	У1
		Имеет навыки самостоятельного решения конструкторских задач, их алгоритмы и программы реализации с использованием возможностей современной вычислительной техники	Н1
Способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	ПК-12	Знает технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах	З2
		Умеет ставить задачу и разрабатывать алгоритм ее решения	У2
		Имеет навыки разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков высокого уровня	Н2
Способность разрабатывать средства автоматизированного проектирования информационных технологий	ПК-13	Знает основы системного программирования, основы объектно-ориентированного подхода к программированию	З3
		Умеет использовать прикладные системы программирования	У3
		Имеет навыки работы с языками процедурного и объектно-ориентированного программирования	Н3
Способность	ПК-15	Знает архитектуру персональных компьютеров и структуру программного обеспечения ПК	З4

Компетенция по ФГОС	Код компетенции по ФГОС	Основные показатели оценивания (показатели достижения результата)	Код показателя оценивания
участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем		Умеет составлять программы на языках высокого уровня, обрабатывающие входные данные, взятые из различных электронных носителей, имеющих различные форматы.	У4
		Имеет навыки владения базовыми математическими знаниями и информационными технологиями	Н4
Способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений	ПК-23	Знает современные технические и программные средства взаимодействия с ЭВМ	35
		Умеет составлять программы с интерфейсом консольного приложения, составлять программы с графическим интерфейсом пользователя	У5
		Имеет навыки выбора элементной базы для построения различных архитектур вычислительных средств	Н5
Способность формировать новые конкурентоспособные идеи и реализовывать их в проектах	ПК-27	Знает средства поиска информации в Интернет, в различных источниках	36
		Умеет оценивать и отбирать наиболее важную информацию, максимально полезную для решения поставленных задач	У6
		Имеет навыки получения/передачи файлов через Интернет.	Н6

3. Указание места дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» относится к базовой части Блока 1 «Дисциплины/модули» основной профессиональной образовательной программы по направлению подготовки 09.03.02 «Информационные системы и технологии» (уровень образования бакалавриат), направленность/профиль «Системотехника и информационные технологии проектирования и управления в строительстве». Дисциплина является обязательной к изучению.

Изучение дисциплины «Объектно-ориентированное программирование» основано на знаниях, умениях и навыках, полученных при изучении обучающимися следующих дисциплин:

- «Информатика»,
- «Программирование на языке высокого уровня»,
- «Математика».

Для освоения дисциплины «Объектно-ориентированное программирование» обучающийся должен:

Знать:

- современные технические и программные средства взаимодействия с ЭВМ,
- технологию разработки алгоритмов и программ,
- методы отладки и решения задач на ЭВМ в различных режимах,
- методы и средства компьютерной графики и геометрического моделирования,

- фундаментальные принципы обработки машинной и внемашинной информации,
- базовые принципы кодирования информации,
- процедурный подход, язык высокого уровня Си,
- основные способы хранения данных в ОП,
- математические алгоритмы решения задач, понятия матриц и правил их вычисления;

Уметь:

- ставить задачу и разрабатывать алгоритм ее решения,
- работать с современными системами программирования, включая объектно-ориентированные,
- решать экономико-математические задачи с использованием ЭВМ,
- кодировать алгоритмы на языке высокого уровня Си,
- использовать математическую библиотеку Си,
- использовать математические модели;

Иметь навыки:

- языками процедурного и объектно-ориентированного программирования,
- навыками разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня,
- навыками работы с системой исчисления и внутреннего представления данных в ЭВМ,
- навыками работы с системой ввода/вывода в Си,
- навыками использования библиотечных функций в Си.

Дисциплина «Объектно-ориентированное программирование» является предшествующей для освоения следующих дисциплин:

- «Технология программирования»
- «Геометрическое моделирование»
- «Вычислительная математика»;
- «Архитектура прикладного программного обеспечения»;
- «Базы данных».

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

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 академических часов.

(1 зачетная единица соответствует 36 академическим часам)

Структура дисциплины:

Форма обучения - очная

№ п/п	Наименование раздела дисциплины (модуля)	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Контактная работа с обучающимися				Самостоятельная работа		
				Лекции	Практико-ориентированные занятия					
					Лабораторный практикум	Практические занятия	Групповые занятия - комп. практикумы	в период теор. обучения	в сессию	
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования	3	1-2	2			2	2	2	
2	Изучение программирования на языке Си++	3	3-4	2			10	10	2	
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	3	5-6	2			6	8	2	
4	Проектирование и разработка ПО	3	7-8	2			4	4	2	Контрольная работа
5	Тестирование и отладка ПО	3	9-10	1			2	4	2	
6	Сопровождение ПО	3	11-12	1			2	8	2	
7	Коллективная разработка. BUG-трекинг. Управление проектом	3	13-14	2			2	2	2	Устный опрос
8	UML и ГОСТ (ЕСПД).	3	15-16	2			2	2	2	
9	Форматирование, документация и другие средства стилистической слаженности кода	3	17-18	2			2	2	2	

	Итого за 3 семестр			16		32	42	18	Зачет, контрольная работа
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL	4	1-6	6		12	20	13	Устный опрос
11	Создание программ на Си++ с графическим интерфейсом пользователя	4	7-12	6		10	11	16	
12	Межпроцессное взаимодействие. Потоки. Взаимодействие программ с ОС	4	13-16	4		10	20	16	
	Итого за 4 семестр :			16		32	51	45	Экзамен, курсовая работа
	Итого			32		64	93	63	

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

5.1. *Содержание лекционных занятий*

№ п/п	Наименование раздела дисциплины	Тема и содержание занятия	Кол-во акад. часов
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	Основные понятия и определения. Особенности создания ПО. Классификация ПО. Основные понятия и определения. ТП в истории. Как хранятся данные в ЭВМ и системы исчисления.	2
2	Изучение программирования на языке Си++	Особенности объектно-ориентированных языков программирования.	2
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	Определение требований к программным продуктам. Функциональные требования. Эксплуатационные требования. Выбор архитектуры ПО. Структура и формат данных. Модульное программирование. Анализ требований и определение спецификаций при структурном подходе. Словарь терминов. Анализ требований и определение спецификаций при объектно-ориентированном подходе.	2
4	Проектирование и разработка ПО	Проектирование при структурном подходе. CASE-технологии. Функциональная схема. Методология RAD. Проектирование при объектно-ориентированном подходе. Диаграммы кооперации. Экстремальное программирование. Рефакторинг. Программирование и тестирование.	2

		Инструментальные средства разработки. Выбор языка. Выбор среды. Модели разработки открытая и проприетарная.	
5	Тестирование и отладка ПО	Тестирование белого ящика и чёрного ящика. Порядок разработки тестов. Модульное тестирование. Интеграционное тестирование. Системное тестирование. Надёжность ПО. Отладка программ.	1
6	Сопровождение ПО	Виды программной документации. Пояснительная записка. Руководство пользователя. Руководство по API для прикладного программиста.	1
7	Коллективная разработка. BUG-трекинг. Управление проектом.	Организация работ при разработке ПО. Ведение проекта по разработке ПО. Прикладное ПО для организации работ по программированию.	2
8	UML и ГОСТ (ЕСПД).	Использование ГОСТ при составлении документации и пояснительной записки. Введение в UML. Основы структурного моделирования. Основы моделирования поведения. Основы моделирования архитектуры.	2
9	Форматирование, документация и другие средства стилистической слаженности кода.	Привила и договорённости в оформлении программного кода. Документирование кода. Декомпозиция. Присвоение имён. Использование языковых средств. Форматирование.	2
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL.	Эффективность и оптимизация программ. Эффективное управление памятью. Проблемы с низкоуровневыми операциями с памятью. Использование кеш памяти и пула объектов. Профилирование программ. Оптимизация использования памяти.	6
11	Создание программ на Си++ с графическим интерфейсом пользователя	Основные модули в разработке ПО. Проблемы безопасности при разработке программ (функции strcpy, strcat).	6
12	Межпроцессное взаимодействие. Поток. Взаимодействие программ с ОС.	Основные понятия при работе с несколькими потоками и общении процессов между собой. Способы общения Программ с ОС.	4
		Итого	32

5.2. Лабораторный практикум

Лабораторный практикум учебным планом не предусмотрен.

5.3. Перечень практических занятий

Практические занятия учебным планом не предусмотрены.

5.4. Групповые занятия – компьютерные практикумы

№ п/п	Наименование раздела дисциплины (модуля)	Тема и содержание группового занятия – компьютерного практикума	Кол-во акад. часов
1	Основы создание интерфейса графического приложения	Создать проект в Qt Creator. В проекте с помощью дизайнера создать интерфейс диалогового окна, главного окна, контекстного меню. Написать код	2

		позволяющий работать с этими элементами. Реализовать технологию «drag&drop» в вашем проекте.	
2	Графика 2D	Создать проект в Qt Creator. В нём создать элемент «Canvas». На нём изобразить построение двух мерной фигуры круг. Изобразить лесенку. Элемент круг должен двигаться по лестнице сначала поднимаясь потом спускаясь с неё.	10
3	Графика 3D	Создать проект в Qt Creator. Изобразить трёхмерный элемент шар. Размножить его по кругу 12 раз, для образования циферблата часов. Сделать, что бы каждый шар изменял свой цвет и форму на куб отображая таким образом который сейчас час.	10
4	Базы данных	Создать проект в Qt Creator. Создать Несколько записей в базе данных SQLite. Создать в программе подключение к этой базе. Создать интерфейс пользователя для подключения и отображения содержимого базы. Добавить возможность добавлять и удалять записи в БД.	10
5	Работа с сетью	Создать проект в Qt Creator. Создать клиентское приложение. Создать серверное приложение. В клиенте должна быть возможность вводить текст. Клиент должен подключаться к серверу и обмениваться с ним текстовыми сообщениями. А сервер в свою очередь передает сообщение другому клиенту. Таким образом сообщение проходит маршрут клиент-сервер-клиент.	10
6	Работа с XML	Создать проект в Qt Creator. Написать файл в формате XML. В приложении включить драйвер XML парсера. Написать парсер для чтения XML файла. Сделать возможность редактирования XML файла.	6
7	Многопоточность	Используя проект по работе с базой данных, добавить возможность сортировки записей в таблице в отдельном потоке, с отображением хода сортировки на экране.	10
8	Встроенные системы	Создать проект в Qt Creator по работе с мобильными устройствами. Перенести приложение по обмену сообщениями (клиентская часть) с компьютера на коммуникатор. Клиенты должны через сеть wifi обмениваться сообщениями через сервер.	6
		Итого	64

5.5. Самостоятельная работа

№ п/п	Наименование раздела дисциплины (модуля)	Виды самостоятельной работы	Кол-во акад. часов	
			в период теор. обучения	в сессию
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	Самостоятельное изучение отдельных тем модуля	12	
2	Изучение программирования на	Самостоятельное изучение отдельных тем модуля	22	

	языке Си++			
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	Самостоятельное изучение отдельных тем модуля	16	
4	UML и ГОСТ (ЕСПД).	Самостоятельное изучение отдельных тем модуля	16	
5	Создание программ на Си++ с графическим интерфейсом пользователя	Самостоятельное изучение отдельных тем модуля	14	
6	Межпроцессное взаимодействие. Поток. Взаимодействие программ с ОС.	Самостоятельное изучение отдельных тем модуля	13	
		Подготовка к экзамену и его сдача	-	63
		Итого	93	0

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

Основные принципы организации самостоятельной работы обучающихся изложены в Положении об организации самостоятельной работы обучающихся (НИУ МГСУ).

Формами самостоятельной работы обучающегося являются:

- чтение и изучение основной и дополнительной литературы, включая справочные издания, конспект лекций,
- изучение нормативной базы по представлению строительных чертежей,
- ознакомление с терминами и понятиями с помощью энциклопедий, словарей, справочников,
- написание собственного конспекта лекций,
- самостоятельное повторное решение задач на компьютере,
- изучение методической литературы по дисциплине (методических указаний и др.),
- осуществление подготовки к мероприятиям текущего контроля (выполнение домашнего задания) и промежуточной аттестации (зачета с оценкой),
- составление перечня неусвоенных вопросов с последующей консультацией у преподавателя.

В самостоятельной работе используются:

- краткий конспект лекций, составленный самостоятельно;
- учебные материалы, указанные в списке основной и дополнительной литературы;

При самостоятельной работе рекомендуется пользоваться учебными материалами, указанными в Приложении 2.

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

Фонд оценочных средств по дисциплине (модулю) для проведения промежуточной аттестации, а также текущего контроля обучающихся является Приложением 1 к рабочей программе дисциплины (модуля).

Оценочные материалы для проведения промежуточной аттестации, а также текущего контроля по дисциплине (модуля) хранятся на кафедре, ответственной за преподавание данной дисциплины.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

Для освоения дисциплины обучающийся может использовать:

- учебно-методическую литературу, имеющуюся в НТБ НИУ МГСУ,
- учебную литературу, размещённую в Электронных библиотечных системах ЭБС АСВ и IPRbooks,
- методическую литературу, размещённую в ЭБС НИУ МГСУ.

Перечень используемой литературы ежегодно обновляется с учётом уровня развития науки и техники и представлен в Приложении 2 к рабочей программе дисциплины.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	http://www.edu.ru/index.php
Научная электронная библиотека	http://elibrary.ru/defaultx.asp?
Электронная библиотечная система IPRbooks	http://www.iprbookshop.ru/
Федеральная университетская компьютерная сеть России	http://www.runnet.ru/
Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
Научно-технический журнал по строительству и архитектуре «Вестник МГСУ»	http://www.vestnikmgsu.ru/
Научно-техническая библиотека НИУ МГСУ	http://www.mgsu.ru/resources/Biblioteka/
Раздел «Кафедры» на официальном сайте НИУ МГСУ	http://www.mgsu.ru/universityabout/Struktura/Kafedri/

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

Перечень тем по разделам дисциплины (модуля) для самостоятельного изучения обучающимися приведён в таблице.

№ п/п	Наименование раздела дисциплины (модуля)	Темы для самостоятельного изучения (в период теоретического обучения)
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	Основные понятия и определения. Особенности создания ПО. Классификация ПО. Основные понятия и определения. ТП в истории. Как хранятся данные в ЭВМ и системы исчисления.
2	Изучение программирования на языке Си++	Особенности объектно-ориентированных языков программирования.
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	Определение требований к программным продуктам. Функциональные требования. Эксплуатационные требования. Выбор архитектуры ПО. Структура и формат данных. Модульное

		программирование. Анализ требований и определение спецификаций при структурном подходе. Словарь терминов. Анализ требований и определение спецификаций при объектно-ориентированном подходе.
4	Проектирование и разработка ПО	Проектирование при структурном подходе. CASE-технологии. Функциональная схема. Методология RAD. Проектирование при объектно-ориентированном подходе. Диаграммы кооперации. Экстремальное программирование. Рефакторинг. Программирование и тестирование. Инструментальные средства разработки. Выбор языка. Выбор среды. Модели разработки открытая и проприетарная.
5	Тестирование и отладка ПО	Тестирование белого ящика и чёрного ящика. Порядок разработки тестов. Модульное тестирование. Интеграционное тестирование. Системное тестирование. Надёжность ПО. Отладка программ.
6	Сопровождение ПО	Виды программной документации. Пояснительная записка. Руководство пользователя. Руководство по API для прикладного программиста.
7	Коллективная разработка. BUG-трекинг. Управление проектом.	Организация работ при разработке ПО. Ведение проекта по разработке ПО. Прикладное ПО для организации работ по программированию.
8	UML и ГОСТ (ЕСПД).	Использование ГОСТ при составлении документации и пояснительной записки. Введение в UML. Основы структурного моделирования. Основы моделирования поведения. Основы моделирования архитектуры.
9	Форматирование, документация и другие средства стилистической слаженности кода.	Правила и договорённости в оформлении программного кода. Документирование кода. Декомпозиция. Присвоение имён. Использование языковых средств. Форматирование.
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL.	Эффективность и оптимизация программ. Эффективное управление памятью. Проблемы с низкоуровневыми операциями с памятью. Использование кеш памяти и пула объектов. Профилирование программ. Оптимизация использования памяти.
11	Создание программ на Си++ с графическим интерфейсом пользователя	Основные модули в разработке ПО. Проблемы безопасности при разработке программ (функции <code>strcpy</code> , <code>strcat</code>).
12	Межпроцессное взаимодействие. Поток. Взаимодействие программ с ОС.	Основные понятия при работе с несколькими потоками и общении процессов между собой. Способы общения Программ с ОС.

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

Организация учебной работы обучающихся на аудиторных занятиях осуществляется в соответствии с п. 4.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

11.1. Перечень информационных технологий, используемых при осуществлении образовательного процесса

№ п/п	Наименование раздела дисциплины (модуля)	Информационные технологии
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	Использование слайд-презентаций при проведении лекционных, практических занятий
2	Изучение программирования на языке Си++	Использование слайд-презентаций при проведении лекционных, практических занятий
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	Использование слайд-презентаций при проведении лекционных, практических занятий
4	Проектирование и разработка ПО	Использование слайд-презентаций при проведении лекционных, практических занятий
5	Тестирование и отладка ПО	Использование слайд-презентаций при проведении лекционных, практических занятий
6	Сопровождение ПО	Использование слайд-презентаций при проведении лекционных, практических занятий
7	Коллективная разработка. BUG-трекинг. Управление проектом.	Использование слайд-презентаций при проведении лекционных, практических занятий
8	UML и ГОСТ (ЕСПД).	Использование слайд-презентаций при проведении лекционных, практических занятий
9	Форматирование, документация и другие средства стилистической слаженности кода.	Использование слайд-презентаций при проведении лекционных, практических занятий
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL.	Использование слайд-презентаций при проведении лекционных, практических занятий
11	Создание программ на Си++ с графическим интерфейсом пользователя	Использование слайд-презентаций при проведении лекционных, практических занятий
12	Межпроцессное взаимодействие. Потоки. Взаимодействие программ с ОС.	Использование слайд-презентаций при проведении лекционных, практических занятий

11.2. Перечень программного обеспечения, используемого при осуществлении образовательного процесса

При освоении дисциплины используется лицензионное и открытое программное обеспечение, указанное в Приложении 3 к рабочей программе.

11.3. *Перечень информационных справочных систем*

Информационно-библиотечные системы

Наименование ИБС	Электронный адрес ресурса
Научная электронная библиотека	http://elibrary.ru/defaultx.asp?
Электронная библиотечная система IPRbooks	http://www.iprbookshop.ru/
Научно-техническая библиотека НИУ МГСУ	http://www.mgsu.ru/resources/Biblioteka/

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю):

Учебные занятия по дисциплине проводятся в оборудованных учебных кабинетах, оснащенных соответствующим оборудованием и программным обеспечением. Перечень материально-технического обеспечения дисциплины (модуля) приведен в Приложении 4 к рабочей программе.

Приложение 1 к рабочей программе

Шифр	Наименование дисциплины
<i>Б1.Б.8</i>	<i>Объектно-ориентированное программирование</i>

Код направления подготовки	09.03.02
Направление подготовки	Информатика и вычислительная техника
Наименование ОПОП	Системотехника и информационные технологии проектирования и управления в строительстве (Академический бакалавриат)
Год начала реализации ОПОП	2017
Уровень образования	Бакалавриат
Форма обучения	очная
Год разработки/обновления	2016

Фонд оценочных средств для проведения промежуточной аттестации и текущего контроля обучающихся по дисциплине (модулю)

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

Формирование компетенций при изучении дисциплины (модуля) происходит поэтапно, по мере освоения обучающимися разделов дисциплины (модуля).

Код компетенции по ФГОС	Этапы формирования компетенций (разделы теоретического обучения)											
	1	2	3	4	5	6	7	8	9	10	11	12
ПК-11	+		+	+			+	+	+	+	+	+
ПК-12	+	+	+	+	+	+	+	+	+	+	+	+
ПК-13		+	+	+	+	+	+	+	+	+	+	+
ПК-15	+		+	+	+	+	+	+	+	+	+	+
ПК-23	+	+		+	+	+	+	+	+	+	+	+
ПК-27		+		+		+		+	+	+	+	+

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Оценивание формирования компетенций осуществляется посредством прохождения обучающимися форм промежуточной аттестации и текущего контроля.

Оценивание формирования компетенций производится на основе показателей оценивания, указанных в п.2. рабочей программы.

2.1. Описание показателей и форм оценивания компетенций

Формами оценивания компетенций являются мероприятия промежуточной аттестации и текущего контроля по дисциплине (модулю), указанные в учебном плане и в п.4 рабочей программы.

Взаимосвязь форм и показателей оценивания компетенций приведена в таблице.

Код компетенции по ФГОС	Показатели освоения (Код показателя оценивания)	Формы оценивания					Обеспеченность оценивания компетенции
		Устный опрос	Текущий контроль		Промежуточная аттестация		
			Контрольная работа	Защита курсовой работы	Зачет	Экзамен	
1	2		4	5	6	7	8
ПК-11	З1		+	+	+	+	
	У1	+		+	+	+	+
	Н1		+	+	+	+	+
ПК-12	З2	+		+	+	+	
	У2		+	+	+	+	+
	Н2	+		+	+	+	+
ПК-13	З3			+		+	
	У3		+	+	+	+	+
	Н3	+	+	+	+	+	+
ПК-15	З4	+		+	+	+	+
	У4		+	+		+	+
	Н4	+		+		+	+
ПК-23	З5		+	+		+	
	У5	+		+	+	+	+
	Н5			+		+	+
ПК-27	З6			+		+	
	У6			+		+	+
	Н6			+		+	+
ИТОГО		+	+	+	+	+	+

2.2. Описание шкалы и критериев оценивания

При проведении промежуточной аттестации в форме экзамена, дифференцированного зачёта, защиты курсовых работ/курсовых проектов используется четырёх балльная шкала оценивания:

Уровень освоения	Оценка
Минимальный	«2» (неудовлетворительно)
Пороговый	«3» (удовлетворительно)
Углубленный	«4» (хорошо)
Продвинутый	«5» (отлично)

При проведении промежуточной аттестации в форме зачёта используется бинарная шкала:

Уровень освоения	Оценка
Ниже порогового	Не зачтено
Пороговый	Зачтено

Критериями оценивания уровня освоения компетенций являются:

Показатели оценивания	Критерии оценивания
Знания	Знание терминов и определений, понятий
	Знание основных закономерностей и соотношений, принципов
	Объём освоенного материала, усвоение всех дидактических единиц (разделов)
	Полнота ответов
	Правильность ответов
Умения	Чёткость изложения и интерпретации знаний
	Освоение методик - умение решать (типовые) практические задачи, выполнять (типовые) задания
	Умение использовать теоретические знания для выбора методики решения задач, выполнения заданий
	Умение проверять решение и анализировать результаты
Навыки	Умение качественно оформлять (презентовать) решение задач и выполнения заданий
	Навыки решения стандартных/нестандартных задач
	Быстрота выполнения трудовых действий оина Е.Б. Математический
	Качество выполнения трудовых действий
	Самостоятельность планирования выполнения трудовых действий

3. *Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций*

3.1. Промежуточная аттестация

Промежуточная аттестация проводится в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся в НИУ МГСУ.

Промежуточная аттестация проводится в форме защиты курсовой работы и зачёта.

К зачету допускаются студенты, успешно выполнившие домашнее задание, задания и упражнения компьютерного практикума.

Перечень типовых вопросов/заданий для проведения экзамена в 4 семестре (очная форма обучения):

№ п/п	Наименование раздела дисциплины (модуля)	Вопросы / задания
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	<ol style="list-style-type: none"> 1. Парадигмы программирования (процедурное, функциональное, логическое и т. д.). 2. Этапы преобразования программы с языка высокого уровня в машинные команды (компиляция, трансляция, интерпретация, связывание, ассемблер). 3. Организация проекта по программированию (файлы *.h, *.c), компиляция (gcc), связывание (статическое, динамическое). 4. Понятие потоков ввода/вывода и их использование (stdin, stdout, stderr).

		5. Запуск внешних приложений из программы (функция <code>system()</code>).
2	Изучение программирования на языке Си++	6. Тип данных <code>string</code> . Основные операции над строками типа <code>string</code> . 7. Тип данных <code>bool</code> . Тип данных <code>void</code> . 8. Понятие пространства имён (<code>namespace</code>) и директивы/объявления <code>using</code> . 9. Препроцессор (<code>#include</code> , <code>#define</code> , <code>#ifdef</code> / <code>#ifndef</code> , <code>#endif</code>). 10. Объявление и инициализация указателей. Опасность при работе с указателем. Указатель на указатель. Указатель на статическую переменную. Указатель на динамическую переменную. Связь массива и указателя. 11. Объявление и инициализация ссылок. Особенности работы ссылок. 12. Динамическое выделение и освобождение памяти (Си и Си++).
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	13. Операторы членства <code>“.”</code> и <code>“->”</code> используемые при работе с составными типами данных. 14. Общий вид функции. Прототипирование и вызов функции. Передача параметров в функцию. Аргументы функции по умолчанию. 15. Встраиваемые функции. Возвращаемое значение. 16. Спецификаторы, квалификаторы памяти.
4	Проектирование и разработка ПО	17. Описание и реализация функций-членов. 18. Конструктор. Объявление и определение конструктора. Использование конструктора. Конструктор по умолчанию. Список инициализаторов. 19. Деструктор. Объявление и использование деструктора. 20. Классы и структуры.
5	Тестирование и отладка ПО	21. Предварительное объявление. 22. Спецификаторы доступа: <code>public</code> , <code>private</code> , <code>protected</code> . 23. Понятие инкапсуляции. Процедурный и объектно-ориентированный подходы в программировании. 24. Модели отношений <code>is-a</code> и <code>has-a</code> . 25. Механизм наследование в классах. Понятия базового и производного классов. Передача данных из базового класса в производный.
6	Сопровождение ПО	26. Отношения между базовым и производными классами. 27. Переопределение функций-членов в классах потомках. Спецификатор доступа: <code>protected</code> .
7	Коллективная разработка. BUG-трекинг. Управление проектом.	28. Перегрузка операций. 29. Полиморфное наследование. 30. Виртуальные функции понятия и реализация. 31. Чисто виртуальные функции. Абстрактный

		базовый класс.
8	UML и ГОСТ (ЕСПД).	32. Автоматическая, статическая и динамическая продолжительность хранения переменных. 33. Внутренние и внешние связывание. Область видимости. 34. Понятие интерфейса. 35. Понятие класса и объекта. 36. Данные и методы.
9	Форматирование, документация и другие средства стилистической слаженности кода.	37. Возвращаемое значение и аргументы функции main() (использование). 38. Объекты cin и cout. 39. Файловый ввод/вывод в Си++.
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL.	40. Множественное наследование. Проблема множественного наследования. 41. Приватное наследование. 42. Механизм включения. 43. Перегрузка функции.
11	Создание программ на Си++ с графическим интерфейсом пользователя	44. Функции scanf(), printf(). 45. Некоторые функции по работе со строками (strcpy(), strcmp(), strlen(), strcat(), atoi(), sprintf()). 46. Файловый ввод/вывод в Си.
12	Межпроцессное взаимодействие. Поток. Взаимодействие программ с ОС.	47. Использование констант с объектами. 48. Неявные функции-члены/операции. 49. Указатель this. 50. Дружественные функции. 51. Дружественные классы. 52. Дружественные функции-члены.

Перечень типовых вопросов/заданий для проведения зачёта в 3 семестре (очная форма обучения):

№ п/п	Наименование раздела дисциплины (модуля)	Вопросы / задания
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования.	1. Парадигмы программирования (процедурное, функциональное, логическое и т. д.). 1. Этапы преобразования программы с языка высокого уровня в машинные команды (компиляция, трансляция, интерпретация, связывание, ассемблер). 2. Организация проекта по программированию (файлы *.h, *.c), компиляция (gcc), связывание (статическое, динамическое). 3. Понятие потоков ввода/вывода и их использование (stdin, stdout, stderr). 4. Запуск внешних приложений из программы (функция system()). 5. Возвращаемое значение и аргументы функции main() (использование). 6. Функции scanf(), printf(). 7. Некоторые функции по работе со строками (strcpy(), strcmp(), strlen(), strcat(), atoi(), sprintf()).
2	Изучение программирования	8. Объявление и инициализация ссылок.

	на языке Си++	<p>Особенности работы ссылок.</p> <p>9. Динамическое выделение и освобождение памяти (Си и Си++).</p> <p>10. Операторы членства “.” и “->” используемые при работе с составными типами данных.</p> <p>11. Общий вид функции. Прототипирование и вызов функции. Передача параметров в функцию. Аргументы функции по умолчанию.</p> <p>12. Встраиваемые функции. Возвращаемое значение.</p>
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	<p>13. Дружественные классы.</p> <p>14. Дружественные функции-члены.</p> <p>15. Предварительное объявление.</p> <p>16. Спецификаторы доступа: public, private, protected.</p> <p>17. Понятие инкапсуляции. Процедурный и объектно-ориентированный подходы в программировании.</p>
4	Проектирование и разработка ПО	<p>18. Файловый ввод/вывод в Си.</p> <p>19. Объекты cin и cout.</p> <p>20. Файловый ввод/вывод в Си++.</p> <p>21. Тип данных string. Основные операции над строками типа string.</p> <p>22. Тип данных bool. Тип данных void.</p> <p>23. Понятие пространства имён (namespace) и директивы/объявления using.</p>
5	Тестирование и отладка ПО	<p>24. Программирование и разработка ПО. Основная трудность оценки программы.</p> <p>25. Структурное программирование, и его основные операторы.</p> <p>26. Метод проектирования сверху вниз. Недостатки указанного метода. Класс решаемых задач.</p> <p>27. Жизненный цикл ПО.</p> <p>28. Каскадная модель разработки ПО.</p> <p>29. Тестирование, отладка и верификация.</p>
6	Сопровождение ПО	<p>30. Внутренние и внешние связывание. Область видимости.</p> <p>31. Понятие интерфейса.</p> <p>32. Понятие класса и объекта.</p> <p>33. Данные и методы.</p>
7	Коллективная разработка. BUG-трекинг. Управление проектом.	<p>34. Препроцессор (#include, #define, #ifdef/#ifndef, #endif).</p> <p>35. Объявление и инициализация указателей. Опасность при работе с указателем. Указатель на указатель. Указатель на статическую переменную. Указатель на динамическую переменную. Связь массива и указателя.</p>
8	UML и ГОСТ (ЕСПД).	<p>36. Спецификаторы, квалификаторы.</p> <p>37. Автоматическая, статическая и динамическая продолжительность хранения.</p> <p>38. Классы и структуры.</p>

		<p>39. Использование констант с объектами.</p> <p>40. Неявные функции-члены/операции.</p> <p>41. Указатель this.</p> <p>42. Дружественные функции.</p>
9	Форматирование, документация и другие средства стилистической слаженности кода.	<p>43. Модели отношений is-a и has-a.</p> <p>44. Механизм наследование в классах. Понятия базового и производного классов. Передача данных из базового класса в производный.</p> <p>45. Отношения между базовым и производными классами.</p> <p>46. Переопределение функций-членов в классах потомках. Спецификатор доступа: protected.</p> <p>47. Описание и реализация функций-членов.</p> <p>48. Конструктор. Объявление и определение конструктора. Использование конструктора. Конструктор по умолчанию. Список инициализаторов.</p> <p>49. Деструктор. Объявление и использование деструктора.</p> <p>50. Множественное наследование. Проблема множественного наследования.</p> <p>51. Приватное наследование.</p> <p>52. Механизм включения.</p> <p>53. Перегрузка функции.</p> <p>54. Перегрузка операций.</p> <p>55. Полиморфное наследование.</p> <p>56. Виртуальные функции понятия и реализация.</p> <p>57. Чисто виртуальные функции. Абстрактный базовый класс.</p> <p>58. Механизм исключений. Блоки try, catch, throw.</p> <p>59. Шаблоны функций и классов.</p> <p>60. Библиотека STL (контейнеры, итераторы).</p> <p>61. Обобщённое программирование.</p> <p>62. Метод проектирования снизу-вверх.</p> <p>63. Недостатки указанного метода. Класс решаемых задач.</p>

Тематика курсовых работ:

Состав типового задания на выполнение курсовых работ/курсовых проектов.
Изучение материала курса завершается выполнением и защитой курсовой работы на тему «Разработка ПО для подсистемы предприятия строительной отрасли».

Возможно написание работ на темы: учёта, расчёта, хранения, прогнозирования и планирования.

1. Календарный график производства работ на объекте.
2. Расчёт нагрузок ЖБ конструкции.

Ход выполнения работы включает следующие этапы:

- получение задания,
- Программирование модулей
- Тестирование программы

- Оформление пояснительной записки. Курсовая работа оформляется в виде программы на ЭВМ и пояснительной записки в электронном виде в формате PDF.

- Защита.

Защита проводится в устной форме, с демонстрацией, работающей и удовлетворяющей поставленным условиям программы на ЭВМ, предполагается открытая защита в учебной группе. Курсовая работа оценивается следующими оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Вопросы к защите курсовой работы:

1. Объяснить каким образом в программе используется ООП подход.
1. Показать в программе где используется инкапсуляция.
2. Рассказать какие трудности были при проектировании и реализации программы.
3. Где хранятся данные программы после её закрытия?
4. Какая связь существует между типом данных и классом?
5. Для чего в классе используется конструктор и деструктор?
6. Используется в программе процедурный подход и где?
7. Сколько времени понадобилось на разработку ПО и в чём были основные трудности?
8. Каким образом хранятся данные в памяти, во время работы ПО?
9. Какие существуют особенности при написании ПО с графическим интерфейсом?
10. В чём преимущество хранения данных в виде списка или дерева?
11. В чём преимущества использования программы по сравнению с расчётами вручную или с использованием офисных пакетов?
12. Какие программы использовались (кроме IDE) для облегчения написания ПО?
13. Применялись ли какие-либо модели разработки при проектировании и реализации ПО?

3.2. Текущий контроль

Перечень проводимых мероприятий текущего контроля: контрольная работа, устный опрос.

Текущий контроль успеваемости представляет собой проверку усвоения учебного материала, регулярно осуществляемый на протяжении семестра.

Типовые контрольные задания мероприятий текущего контроля:

Контрольная работа.

Тема: UML и ГОСТ (ЕСПД).

Вариант задания (пример)

1. Разработать схему алгоритма сортировки используя графические обозначения и правила ЕСПД.
2. Разработать схему алгоритма сортировки используя графические обозначения и правила UML.
3. Разработать концептуальную схему БД используя графические обозначения и правила UML сущность-связь.
4. Рассказать про модели программирования.

Перечень проводимых мероприятий текущего контроля: *устный опрос*

Типовые вопросы для проведения текущего контроля в виде устного опроса:

3 семестр:

1. Что определяет класс? Чем отличается класс от объекта?
2. Можно ли объявлять массив объектов? А массив классов?
3. Разрешается ли объявлять указатель на объект? А указатель на класс?
4. Допускается ли передавать объекты в качестве параметров, и какими способами? А возвращать как результат?
5. Как называется использование объекта одного класса в качестве поля другого класса?
6. Является ли структура классом? Чем класс отличается от структуры?
7. Какие ключевые слова в C++ обозначают класс?
8. Объясните принцип инкапсуляции.
9. Что такое композиция?
10. Для чего используются ключевые слова `public` и `private`?
11. Можно ли использовать ключевые слова `public` и `private` в структуре?
12. Существуют ли ограничения на использование `public` и `private` в классе? А в структуре?
13. Обязательно ли делать поля класса приватными?
14. Что такое метод? Как вызывается метод?
15. Может ли метод быть при
33. Что означает выражение `*this`? В каких случаях оно используется?
34. Какие операции не рекомендуется перегружать как методы класса? Почему?
35. Какие операции разрешается перегружать только как методы класса?
36. Дайте определение дружественной функции. Как объявляется дружественная функция? А как определяется?
37. Дайте определение конструктора. Каково назначение конструктора? Перечислите отличия конструктора от метода.
- ватный?
16. Как определить метод непосредственно внутри класса? А вне класса? Чем эти определения отличаются?
17. Можно в методах присваивать параметрам значения по умолчанию?
18. Что обозначается ключевым словом `this`?
19. Зачем нужны константные методы? Чем отличается определение константного метода от обычного?
20. Может ли константный метод вызываться для объектов-переменных? А обычный метод — для объектов-констант?
21. Объясните принцип полиморфизма.
22. Сколько места в памяти занимает объект класса? Как это узнать?
23. Каков размер «пустого» объекта?
24. Влияют ли методы на размер объекта?
25. Одинаков ли размер класса и аналогичной структуры?
26. Какие операции нельзя перегружать? Как вы думаете, почему?
27. Можно ли перегружать операции для встроенных типов данных?
28. Можно ли при перегрузке изменить приоритет операции?
29. Можно ли определить новую операцию?
30. Перечислите особенности перегрузки операций как методов класса. Чем отличается перегрузка внешним образом от перегрузки как метода класса?
31. Какой результат должны возвращать операции с присваиванием?
32. Как различаются перегруженная префиксная и постфиксная операции инкремента и декремента?
33. Что означает выражение `*this`? В каких случаях оно используется?
34. Какие операции не рекомендуется перегружать как методы класса? Почему?
35. Какие операции разрешается перегружать только как методы класса?

36. Дайте определение дружественной функции. Как объявляется дружественная функция? А как определяется?

37. Дайте определение конструктора. Каково назначение конструктора? Перечислите отличия конструктора от метода.

4 семестр:

Сколько конструкторов может быть в классе? Допускается ли перегрузка конструкторов? Какие виды конструкторов создаются по умолчанию?

39. Может ли конструктор быть приватным? Какие последствия влечет за собой объявление конструктора приватным?

40. Приведите несколько случаев, когда конструктор вызывается неявно.

41. Как проинициализировать динамическую переменную?

42. Как объявить константу в классе? Можно ли объявить дробную константу?

43. Каким образом разрешается инициализировать константные поля в классе?

44. В каком порядке инициализируются поля в классе? Совпадает ли этот порядок с порядком перечисления инициализаторов в списке инициализации конструктора?

45. Какие конструкции C++ разрешается использовать в списке инициализации в качестве инициализирующих выражений?

46. Какой вид конструктора фактически является конструктором преобразования типов?

47. Для чего нужны функции преобразования? Как объявить такую функцию в классе?

48. Как запретить неявное преобразование типа, выполняемое конструктором инициализации?

49. Какие проблемы могут возникнуть при определении функций преобразования?

50. Для чего служит ключевое слово `explicit`?

51. Влияет ли наличие целочисленных констант-полей на размер класса?

52. Разрешается ли объявлять массив в качестве поля класса. Как присвоить элементам массива начальные значения?

53. Сколько операндов имеет операция индексирования `[]`? Какой вид результата должна возвращать эта операция?

54. Для чего нужны статические поля в классе? Как они определяются?

55. Как объявить в классе и проинициализировать статический константный массив?

56. Что такое выравнивание и от чего оно зависит? Влияет ли выравнивание на размер класса?

57. Дайте определение контейнера.

58. Какие виды встроенных контейнеров в C++ вы знаете?

59. Какие виды доступа к элементам контейнера вам известны?

60. Чем отличается прямой доступ от ассоциативного?

61. Перечислите операции, которые обычно реализуются для последовательного доступа к элементам контейнера.

62. Дайте определение итератора.

63. Можно ли реализовать последовательный доступ без итератора? В чем преимущества реализации последовательного доступа с помощью итератора?

64. Что играет роль итератора для массивов C++?

65. Что такое деструктор? Может ли деструктор иметь параметры?

66. Почему для классов-контейнеров деструктор надо писать явным образом?

67. Допускается ли перегрузка деструкторов?

68. Что такое «глубокое копирование» и когда в нем возникает необходимость?

69. Какое копирование осуществляет стандартный конструктор копирования?

70. Чем отличается копирование от присваивания?

71. Объясните, почему в операции присваивания требуется проверка присваивания самому себе?
72. Можно ли в качестве операции индексирования использовать операцию вызова функции ()? В чем ее преимущества перед операцией []?
73. Почему необходимо писать два определения операции индексирования? Чем они отличаются?
74. Дайте определение вложенного класса.
75. Можно ли класс-итератор реализовать как внешний класс? А как вложенный? В чем отличия этих методов реализации?
76. Может ли объемлющий класс иметь неограниченный доступ к элементам вложенного класса? А вложенный класс — к элементам объемлющего?
77. Ограничена ли глубина вложенности классов?
78. Можно ли определить вложенный класс внешним образом? Зачем это может понадобиться?
79. Каким образом вложенный класс может использовать методы объемлющего класса? А объемлющий — методы вложенного?
80. Что такое «запредельный» элемент, какую роль он играет в контейнерах?
81. Объясните, по каким причинам трудно написать универсальный контейнер, элементы которого могут иметь произвольный тип.
82. Назовите ключевые слова C++, которые используются для обработки исключений.
83. Исключение — это:
- 1) событие;
 - 2) ситуация;
 - 3) объект;
 - 4) ошибка в программе;
 - 5) прерывание;
84. Каким образом исключение генерируется?
85. Каковы функции контролируемого блока?
86. Что обозначается ключевым словом `catch`?
- 1) контролируемый блок;
 - 2) блок обработки исключения;
 - 3) секция-ловушка;
 - 4) генератор исключения;
 - 5) обработчик прерывания;
87. Какого типа может быть исключение?

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

Процедура промежуточной аттестации регламентируется с Положением о текущем контроле и промежуточной аттестации обучающихся в НИУ МГСУ.

4.1. Процедура оценивания при проведении промежуточной аттестации обучающихся по дисциплине (модулю) в форме экзамена зачёта

Промежуточная аттестация по дисциплине (модулю) проводится в форме экзамена в 4 семестре.

Используется четырёх балльная шкала оценивания освоения, указанная в п.2.2.

Используются критерии оценивания, указанные п.2.2.

Оценка выставляется преподавателем интегрально по всем показателям и критериям оценивания.

Код показателя оценивания	Оценка			
	«2» (неудовлетв.)	Пороговый уровень освоения	Углубленный уровень освоения	Продвинутый уровень освоения
		«3» (удовлетвор.)	«4» (хорошо)	«5» (отлично)
31 32 33 34 35 36	Обучающийся не знает значительной части программного материала в части формирования основных принципов программирования на языке высокого уровня. Допускает существенные ошибки. Не ориентируется в материале.	Обучающийся имеет знания только основного материала в части формирования основных принципов программирования на языке высокого уровня, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.	Обучающийся твердо знает материал программирования на языке высокого уровня. Грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, уверенно ориентируется в материале.	Обучающийся глубоко и прочно усвоил программный материал в части программирования на языке высокого уровня. Исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал из литературы, правильно обосновывает принятое решение.
У1 У2 У3 У4 У5 У6	Обучающийся неуверенно, с большими затруднениями выполняет практические работы, необходимые компетенции не сформированы, не умеет применить теоретические знания при решении практических задач.	Обучающийся имеет знания только основного материала в части программирования на языке высокого уровня, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике при ответе на практические вопросы экзаменатора	Обучающийся правильно применяет теоретические положения при решении практических вопросов и задач.	Обучающийся свободно справляется с задачами, вопросами и другими видами применения знаний. Умеет анализировать полученные результаты; проявляет самостоятельность при выполнении практических заданий.

N1 N2 N3 N4 N5 N6	Обучающийся не показывает практических навыков в выполнении практических заданий, требуемых составом компетенций.	Обучающийся имеет знания только основного материала в части программирования на языке высокого уровня, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике, при ответе на практические вопросы экзаменатора.	Обучающийся имеет прочные знания программирования на языке высокого уровня, уверенно применяет теоретические знания.	Обучающийся не только имеет прочные знания программирования на языке высокого уровня, но свободно оперирует материалом из литературы.
----------------------------------	---	--	--	---

4.2. Процедура оценивания при проведении промежуточной аттестации обучающихся по дисциплине (модулю) в форме зачёта

Промежуточная аттестация по дисциплине (модулю) проводится в форме Зачёта в 3 семестре.

Для оценивания знаний, умений и навыков используются критерии, указанные п.2.2.

Код показателя оценивания	Оценка	
	Не зачтено	Зачтено
31 32 33 34 35 36	Обучающийся не знает значительной части программного материала в части программирования на языке высокого уровня. Допускает существенные ошибки. Не может проиллюстрировать полученные знания в процессе выполнения курсовой работы	Обучающийся имеет знания только основного материала в части программирования, но не усвоил его деталей, допускает неточности, что приводит к удовлетворительному уровню выполнения курсовой работы
У1 У2 У3 У4 У5 У6	Обучающийся неуверенно, с большими затруднениями выполняет практические работы, необходимые практические компетенции не сформированы, не умеет применить теоретические знания при решении практических задач, возникающих при выполнении курсовой работы.	Обучающийся имеет знания только основного материала, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике при ответе на практические вопросы
N1 N2 N3 N4 N5 N6	Обучающийся не показывает практических навыков в выполнении практических заданий, требуемых составом компетенций. Не может выполнить курсовую работу.	Обучающийся имеет знания только основного материала в части программирования на языке высокого уровня, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике при выполнении курсовой работы

4.3. Процедура оценивания при проведении промежуточной аттестации обучающихся по дисциплине (модулю) в форме защиты курсовой работы/проекта

Процедура защиты курсовой работы (проекта) определена Положением о курсовом проекте (работе) обучающихся НИУ МГСУ:

Промежуточная аттестация по дисциплине (модулю) проводится в форме защиты курсовой работы/курсового проекта в 4 семестре.

Используется четырех балльная шкала оценивания освоения, указанная в п.2.2.

Используются критерии оценивания, указанные п.2.2.

Код показателя оценивания	Оценка			
	«2» (неудовлетв.)	Пороговый уровень освоения	Углубленный уровень освоения	Продвинутый уровень освоения
		«3» (удовлетвор.)	«4» (хорошо)	«5» (отлично)
31 32 33 34 35 36	Обучающийся не знает значительной части программного материала в части программирования на языке высокого уровня. Допускает существенные ошибки. Не может проиллюстрировать полученные знания в процессе выполнения курсовой работы	Обучающийся имеет знания только основного материала в части программирования, но не усвоил его деталей, допускает неточности, что приводит к неудовлетворительному уровню выполнения курсовой работы	Обучающийся твердо знает материал в части программирования на языке высокого уровня, что позволяет успешно справиться с задачами, встающими в процессе выполнения курсовой работы.	Обучающийся глубоко и прочно усвоил программный материал в части технологий программирования на языке высокого уровня. Умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний в курсовой работе, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал из литературы, правильно обосновывает принятое решение.
У1 У2 У3 У4 У5 У6	Обучающийся неуверенно, с большими затруднениями выполняет практические работы, необходимые компетенции не сформированы, не умеет применить теоретические знания при решении практических задач, возникающих при выполнении курсовой работы.	Обучающийся имеет знания только основного материала, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике при ответе на практические вопросы	Обучающийся правильно применяет теоретические положения при решении практических вопросов и задач, возникающих в процессе выполнения курсовой работы.	Обучающийся свободно справляется с задачами, вопросами и другими видами применения знаний в процессе выполнения курсовой работы. Умеет анализировать полученные результаты; проявляет самостоятельность при выполнении курсовой работы
Н1 Н2 Н3 Н4	Обучающийся не показывает практических навыков в выполнении	Обучающийся имеет знания только основного материала в части программирования	Обучающийся имеет прочные навыки проектной работы в области программирования	Обучающийся не только имеет прочные навыки работы в области программирования, но свободно оперирует

<p>H5 H6</p>	<p>практических заданий, требуемых составом компетенций. Не может выполнить курсовую работу.</p>	<p>на языке высокого уровня, но не усвоил его деталей, испытывает затруднения в применении теоретических положений на практике при выполнении курсовой работы</p>	<p>на языке высокого уровня, для чего уверенно применяет теоретические знания в процессе выполнения курсовой работы.</p>	<p>объемом необходимых знаний в собственной курсовой работе</p>
------------------	--	---	--	---

Приложение 2 к рабочей программе

Шифр	Наименование дисциплины
Б1.Б.8	Объектно-ориентированное программирование
Код направления подготовки	09.03.02
Направление подготовки	Информатика и вычислительная техника
Наименование ОПОП	Системотехника и информационные технологии проектирования и управления в строительстве (Академический бакалавриат)
Год начала реализации ОПОП	2017
Уровень образования	Бакалавриат
Форма обучения	очная
Год разработки/обновления	2016

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

№ п/п	Наименование дисциплины (модуля) в соответствии с учебным планом	Автор, название, место издания, издательство, год издания учебной и учебно-методической литературы, количество страниц	Количество экземпляров печатных изданий в библиотеке НИУ МГСУ	Число обучающихся, одновременно изучающих дисциплину (модуль)
1	2	3	4	5
<i>Основная литература:</i>				
		НТБ НИУ МГСУ		
1	Объектно-ориентированное программирование	Казанский, А. А. Объектно-ориентированное программирование на Visual Basic 2010 и Visual C# 2010 в среде разработки Microsoft Visual Studio [Текст] : учебное пособие и практикум / А. А. Казанский ; [рец.: В. В. Серов, В. С. Варников] ; Моск. гос. строит. ун-т. - Москва : МГСУ, 2012. - 399 с.	25	60
2	Объектно-ориентированное программирование	Хорев, П. Б. Объектно-ориентированное программирование [Текст] : учебное пособие для студентов, обучающихся по направлению "Информатика и вычислительная техника" / П. Б. Хорев. - 4-с изд., стернотип. - Москва : Академия, 2012. - 447 с.	15	60
		ЭБС АСВ		

3	Объектно-ориентированное программирование	Объектно-ориентированное программирование и программная инженерия [Электронный ресурс]/ Мейер Б.— Электрон.текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 285 с.	http://www.iprbookshop.ru/39552	60
---	---	--	---	----

Согласовано:

НТБ

26.10.2016

дата



НТБ МГСУ

Подпись, ФИО

Приложение 3 к рабочей программе

Шифр	Наименование дисциплины
Б1.Б.8	Объектно-ориентированное программирование

Код направления подготовки	09.03.02
Направление подготовки	Информатика и вычислительная техника
Наименование ОПОП	Системотехника и информационные технологии проектирования и управления в строительстве (Академический бакалавриат)
Год начала реализации ОПОП	2017
Уровень образования	Бакалавриат
Форма обучения	очная
Год разработки/обновления	2016

Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине (модулю)

№	Наименование раздела дисциплины (модуля)	Тема	Наименование программного обеспечения	Тип лицензии
1	История ТП и программирования. Основные понятия и определения. Парадигмы программирования	Основные понятия и определения. Особенности создания ПО. Классификация ПО. Основные понятия и определения. ТП в истории. Как хранятся данные в ЭВМ и системы исчисления.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
2	Изучение программирования на языке Си++	Особенности объектно-ориентированных языков программирования.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
3	Анализ проблемы (задачи) и синтез спецификаций к ПО	Определение требований к программным продуктам. Функциональные требования. Эксплуатационные требования. Выбор архитектуры ПО. Структура и формат данных. Модульное программирование. Анализ требований и определение спецификаций при структурном подходе. Словарь терминов. Анализ требований и определение спецификаций при объектно-ориентированном подходе.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
4	Проектирование	Проектирование при структурном	Microsoft	DreamSpark

	и разработка ПО	подходе. CASE-технологии. Функциональная схема. Методология RAD. Проектирование при объектно-ориентированном подходе. Диаграммы кооперации. Экстремальное программирование. Рефакторинг. Программирование и тестирование. Инструментальные средства разработки. Выбор языка. Выбор среды. Модели разработки открытая и проприетарная.	Windows MinGW Code::Blocks Qt5 Toolkit 5	subscription Свободное ПО
5	Тестирование и отладка ПО	Тестирование белого ящика и чёрного ящика. Порядок разработки тестов. Модульное тестирование. Интеграционное тестирование. Системное тестирование. Надёжность ПО. Отладка программ.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
6	Сопровождение ПО	Виды программной документации. Пояснительная записка. Руководство пользователя. Руководство по API для прикладного программиста.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
7	Коллективная разработка. BUG-трекинг. Управление проектом	Организация работ при разработке ПО. Ведение проекта по разработке ПО. Прикладное ПО для организации работ по программированию.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
8	UML и ГОСТ (ЕСПД).	Использование ГОСТ при составлении документации и пояснительной записки. Введение в UML. Основы структурного моделирования. Основы моделирования поведения. Основы моделирования архитектуры.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
9	Форматирование, документация и другие средства стилистической слаженности кода	Правила и договорённости в оформлении программного кода. Документирование кода. Декомпозиция. Присвоение имён. Использование языковых средств. Форматирование.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
10	Особенности работы с памятью в Си++ (кеш паять). Эффективность производительность. Использование STL	Эффективность и оптимизация программ. Эффективное управление памятью. Проблемы с низкоуровневыми операциями с памятью. Использование кеш памяти и пула объектов. Профилирование программ. Оптимизация использования памяти.	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
11	Создание программ на Си++ с графическим интерфейсом пользователя	Основные модули в разработке ПО. Проблемы безопасности при разработке программ (функции strcpy, strcat).	Microsoft Windows MinGW Code::Blocks Qt5 Toolkit 5	DreamSpark subscription Свободное ПО
12	Межпроцессное взаимодействие.	Основные понятия при работе с несколькими потоками и общение	Microsoft Windows	DreamSpark subscription

Потоки. Взаимодействие программ с ОС	процессов между собой. Способы общения Программ с ОС.	MinGW Code::Blocks Qt5 Toolkit 5	Свободное ПО
--	--	--	--------------

Приложение 4 к рабочей программе

Шифр	Наименование дисциплины
<i>Б1.Б.8</i>	<i>Объектно-ориентированное программирование</i>

Код направления подготовки	09.03.02
Направление подготовки	Информатика и вычислительная техника
Наименование ОПОП	Системотехника и информационные технологии проектирования и управления в строительстве (Академический бакалавриат)
Год начала реализации ОПОП	2017
Уровень образования	Бакалавриат
Форма обучения	очная
Год разработки/обновления	2016

Перечень материально-технического обеспечения по дисциплине (модулю):

№ п/п	Вид учебного занятия	Наименование оборудования	№ и наименование оборудованных учебных кабинетов, объектов для проведения практических занятий
1	2	3	4
1	Лекции	стационарные / мобильные (переносные) наборы демонстрационного мультимедийного оборудования (проектор, микрофон, экран, компьютер)	Учебные аудитории для проведения занятий лекционного типа (129337, г. Москва, ш. Ярославское, д.26, корп. 7, помещение 8 комн.14, 17,63,64.)
2	Групповые занятия – компьютерные практикумы	Компьютерный класс: 26 персональных компьютеров с конфигурацией: 3,06 ГГц, HDD 500 Гб, RAM 4 Гб, Video RAM 512 Мб, DVD-R/RW, монитор 22 "; экран проекционный (Projecta ELPRO EL) (1 шт.); Компьютерный класс: 24 персональных компьютеров с конфигурацией: 3,06 ГГц, HDD 500 Гб, RAM 4 Гб, Video RAM 512 Мб, DVD-R/RW, монитор 22 "; экран проекционный (Projecta ELPRO EL) (1 шт.); Компьютерный класс: 27 персональных компьютеров с конфигурацией: 3,06 ГГц, HDD 500 Гб, RAM 4 Гб, Video RAM 512 Мб, DVD-R/RW, монитор 22 "; экран проекционный ELPRO EL 168*220 MW VID (1 шт.), проектор/тип №3 Epson (1 шт.); Компьютерный класс: 18 персональных компьютеров с конфигурацией: 3,06 ГГц, HDD 500 Гб, RAM 4 Гб, Video RAM 512 Мб, DVD-R/RW, монитор 22 "; экран проекционный Projecta Professional (2 шт.).	Аудитории / аудитория для проведения занятий семинарского типа в соответствии с перечнем аудиторного фонда (129337, г. Москва, ш. Ярославское, д.26, корп. 20, помещение 1, комн. 14,15,16,17.)
3	Самостоятельная работа	29 персональных компьютеров с конфигурацией: 1,6 ГГц, HDD 80 Гб, RAM 1 Гб, Video RAM 128 Мб, DVD-R/RW, монитор 17 "	Помещение для самостоятельной работы обучающихся (129337, г. Москва, ш. Ярославское, д.26, корп. 2, помещение 6, комн. 5.)