

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Набережночелнинский институт (филиал)
Экономическое отделение



Утверждаю

Заместитель директора
по образовательной деятельности
НЧИ КФУ Н.Д.Ахметов



« _____ » _____ 20__ г.

подписано электронно-цифровой подписью

Программа дисциплины

Практикум по объектно-ориентированному программированию

Направление подготовки: 09.04.03 - Прикладная информатика

Профиль подготовки: Реинжиниринг бизнес-процессов предприятий, организаций, банков

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
 - 6.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения
 - 6.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания
 - 6.3. 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
- 6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработал(а)(и) доцент, к.н. (доцент) Лысанов Д.М. (Кафедра бизнес-информатики и математических методов в экономике, Экономическое отделение), DMLysanov@kpfu.ru

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-3	Способен анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями;

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

ОПК-3: знать способы анализа профессиональной информации, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями

Должен уметь:

ОПК-3: уметь анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями

Должен владеть:

ОПК-3: владеть навыками анализа профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями

Должен демонстрировать способность и готовность:

ОПК-3: демонстрировать способность и готовность анализировать профессиональную информацию, выделять в ней главное, структурировать, оформлять и представлять в виде аналитических обзоров с обоснованными выводами и рекомендациями

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.05 Дисциплины (модули)" основной профессиональной образовательной программы 09.04.03 "Прикладная информатика (Реинжиниринг бизнес-процессов предприятий, организаций, банков)" и относится к обязательным дисциплинам.

Осваивается на 1 курсе в 1 семестре.

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

Общая трудоемкость дисциплины составляет 6 зачетных(ые) единиц(ы) на 216 часа(ов).

Контактная работа - 36 часа(ов), в том числе лекции - 0 часа(ов), практические занятия - 0 часа(ов), лабораторные работы - 36 часа(ов), контроль самостоятельной работы - 0 часа(ов).

Самостоятельная работа - 144 часа(ов).

Контроль (зачёт / экзамен) - 36 часа(ов).

Форма промежуточного контроля дисциплины: экзамен в 1 семестре.

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

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Методы и классы	1	0	0	3	14
2.	Тема 2. Перегрузка операторов	1	0	0	3	14
3.	Тема 3. Индексаторы и свойства	1	0	0	3	14
4.	Тема 4. Наследование	1	0	0	3	14
5.	Тема 5. Интерфейсы, структуры и перечисления	1	0	0	4	14
6.	Тема 6. Обработка исключительных ситуаций	1	0	0	4	14
7.	Тема 7. Применение средств ввода-вывода	1	0	0	4	15
8.	Тема 8. Делегаты, события и лямбда-выражения	1	0	0	4	15
9.	Тема 9. Пространства имен, препроцессор и сборки	1	0	0	4	15

4.2 Содержание дисциплины (модуля)

Тема 1. Методы и классы

10. Динамическая идентификация типов, рефлексия. Управление доступом к членам класса, Модификаторы доступа, Организация закрытого и открытого доступа, Практический пример организации управления доступом, Передача объектов методам по ссылке, Способы передачи аргументов методу, Использование модификаторов параметров ref и out, Использование модификатора параметра ref, Использование модификатора параметра out, Использование модификаторов ref и out для ссылок на объекты, Использование переменного числа аргументов, Возврат объектов из методов, Возврат массива из метода, Перегрузка методов, Перегрузка конструкторов, Вызов перегружаемого конструктора с помощью ключевого слова this, Инициализаторы объектов, Необязательные аргументы, Необязательные аргументы и перегрузка методов, Необязательные аргументы и неоднозначность, Практический пример использования необязательных аргументов, Именованные аргументы, Метод Main(), Возврат значений из метода Main(), Передача аргументов методу Main(), Рекурсия, Применение ключевого слова static, Статические конструкторы, Статические классы

Тема 2. Перегрузка операторов

Основы перегрузки операторов, Перегрузка бинарных операторов, Перегрузка унарных операторов, Выполнение операций со встроенными в C# типами данных, Перегрузка операторов отношения, Перегрузка операторов true и false, Перегрузка логических операторов, Простой способ перегрузки логических операторов, Как сделать укороченные логические операторы доступными для применения, Операторы преобразования, Рекомендации и ограничения по перегрузке операторов, Пример перегрузки операторов

Тема 3. Индексаторы и свойства

Индексаторы, Создание одномерных индексаторов, Перегрузка индексаторов, Индексаторы без базового массива, Многомерные индексаторы, Свойства, Автоматически реализуемые свойства, Применение инициализаторов объектов в свойствах, Ограничения, присущие свойствам, Применение модификаторов доступа в аксессуарах, Применение индексаторов и свойств

Тема 4. Наследование

Основы наследования, Доступ к членам класса и наследование, Организация защищенного доступа, Конструкторы и наследование, Вызов конструкторов базового класса, Наследование и сокрытие имен, Применение ключевого слова base для доступа к скрытому имени, Создание многоуровневой иерархии классов, Порядок вызова конструкторов, Ссылки на базовый класс и объекты производных классов, Виртуальные методы и их переопределение, Что дает переопределение методов, Применение виртуальных методов, Применение абстрактных классов, Предотвращение наследования с помощью ключевого слова sealed, Класс object, Упаковка и распаковка, Класс object как универсальный тип данных

Тема 5. Интерфейсы, структуры и перечисления

Интерфейсы, Реализация интерфейсов, Применение интерфейсных ссылок, Интерфейсные свойства, Интерфейсные индексы, Наследование интерфейсов, Скрытие имен при наследовании интерфейсов, Явные реализации, Выбор между интерфейсом и абстрактным классом, Стандартные интерфейсы для среды .NET Framework, Структуры, О назначении структур, Перечисления, Инициализация перечисления, Указание базового типа перечисления, Применение перечислений

Тема 6. Обработка исключительных ситуаций

Класс System.Exception, Основы обработки исключительных ситуаций, Применение пары ключевых слов try и catch, Пример обработки исключительной ситуации, Последствия перехвата исключений, Обработка исключительных ситуаций, Применение нескольких операторов catch, Перехват всех исключений, Вложение блоков try, Генерирование исключений вручную, Повторное генерирование исключений, Использование блока finally, Подробное рассмотрение класса Exception, Наиболее часто используемые исключения, Получение производных классов исключений, Перехват исключений производных классов, Применение ключевых слов checked и unchecked

Тема 7. Применение средств ввода-вывода

Организация системы ввода-вывода в C# на потоках, Байтовые и символьные потоки, Встроенные потоки, Классы потоков, Класс Stream, Классы байтовых потоков, Классы-оболочки символьных потоков, Двоичные потоки, Консольный ввод-вывод, Чтение данных из потока ввода с консоли, Применение метода ReadKey(), Запись данных в поток вывода на консоль, Класс FileStream и байтовый ввод-вывод в файл, Открытие и закрытие файла, Чтение байтов из потока файлового ввода-вывода, Запись в файл, Использование класса FileStream для копирования файла, Символьный ввод-вывод в файл, Применение класса StreamWriter, Применение класса StreamReader, Переадресация стандартных потоков, Чтение и запись двоичных данных, Класс BinaryWriter, Класс BinaryReader, Демонстрирование двоичного ввода-вывода, Файлы с произвольным доступом, Применение класса MemoryStream, Применение классов StringReader и StringWriter, Класс File, Копирование файлов с помощью метода Copy(), Применение методов Exists() и GetLastAccessTime(), Преобразование числовых строк в их внутреннее представление

Тема 8. Делегаты, события и лямбда-выражения

Делегаты, Групповое преобразование делегируемых методов, Применение методов экземпляра в качестве делегатов, Групповая адресация, Ковариантность и контравариантность, Класс System.Delegate, Назначение делегатов, Анонимные функции, Анонимные методы, Передача аргументов анонимному методу, Возврат значения из анонимного метода, Применение внешних переменных в анонимных методах, Лямбда-выражения, Лямбда-оператор, Одиночные лямбда-выражения, Блочные лямбда-выражения, События, Пример групповой адресации события, Методы экземпляра в сравнении со статическими методами в качестве обработчиков событий, Применение аксессоров событий, Разнообразные возможности событий, Применение анонимных методов и лямбда-выражений вместе с событиями, Рекомендации по обработке событий в среде .NET Framework, Применение делегатов EventHandler<TEventArgs> и EventHandler, Практический пример обработки событий

Тема 9. Пространства имен, препроцессор и сборки

Пространства имен, Объявление пространства имен, Предотвращение конфликтов имен с помощью пространств имен, Директива using, Вторая форма директивы using, Аддитивный характер пространств имен, Вложенные пространства имен, Глобальное пространство имен, Применение описателя псевдонима пространства имен ::, Препроцессор, Директива #define, Директивы #if и #endif, Директивы #else и #elif, Директива #undef, Директива #error, Директива #warning, Директива #line, Директивы #region и #endregion, Директива #pragma, Сборки и модификатор доступа internal, Модификатор доступа internal

Тема 10. Динамическая идентификация типов, рефлексия и атрибуты

Динамическая идентификация типов, Проверка типа с помощью оператора is, Применение оператора as, Применение оператора typeof, Рефлексия, Класс System.Type - ядро подсистемы рефлексии, Применение рефлексии, Получение сведений о методах, Вызов методов с помощью рефлексии, Получение конструкторов конкретного типа, Получение типов данных из сборок, Полностью автоматизированное обнаружение типов, Атрибуты, Основы применения атрибутов, Сравнение позиционных и именованных параметров, Встроенные атрибуты, Атрибут AttributeUsage, Атрибут Conditional, Атрибут Obsolete

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

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

6.1 Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы и форм контроля их освоения

Этап	Форма контроля	Оцениваемые компетенции	Темы (разделы) дисциплины
Семестр 1			
	<i>Текущий контроль</i>		
1	Лабораторные работы	ОПК-3	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления 6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
2	Устный опрос	ОПК-3	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления 6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
3	Тестирование	ОПК-3	1. Методы и классы 2. Перегрузка операторов 3. Индексаторы и свойства 4. Наследование 5. Интерфейсы, структуры и перечисления 6. Обработка исключительных ситуаций 7. Применение средств ввода-вывода 8. Делегаты, события и лямбда-выражения 9. Пространства имен, препроцессор и сборки 10. Динамическая идентификация типов, рефлексия и атрибуты
	<i>Экзамен</i>	ОПК-3	

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

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Семестр 1					
Текущий контроль					

Форма контроля	Критерии оценивания				Этап
	Отлично	Хорошо	Удовл.	Неуд.	
Лабораторные работы	Оборудование и методы использованы правильно. Проявлена превосходная теоретическая подготовка. Необходимые навыки и умения полностью освоены. Результат лабораторной работы полностью соответствует её целям.	Оборудование и методы использованы в основном правильно. Проявлена хорошая теоретическая подготовка. Необходимые навыки и умения в основном освоены. Результат лабораторной работы в основном соответствует её целям.	Оборудование и методы частично использованы правильно. Проявлена удовлетворительная теоретическая подготовка. Необходимые навыки и умения частично освоены. Результат лабораторной работы частично соответствует её целям.	Оборудование и методы использованы неправильно. Проявлена неудовлетворительная теоретическая подготовка. Необходимые навыки и умения не освоены. Результат лабораторной работы не соответствует её целям.	1
Устный опрос	В ответе качественно раскрыты содержание темы. Ответ хорошо структурирован. Прекрасно освоен понятийный аппарат. Продemonстрирован высокий уровень понимания материала. Превосходное умение формулировать свои мысли, обсуждать дискуссионные положения.	Основные вопросы темы раскрыты. Структура ответа в целом адекватна теме. Хорошо освоен понятийный аппарат. Продemonстрирован хороший уровень понимания материала. Хорошее умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема частично раскрыта. Ответ слабо структурирован. Понятийный аппарат освоен частично. Понимание отдельных положений из материала по теме. Удовлетворительное умение формулировать свои мысли, обсуждать дискуссионные положения.	Тема не раскрыта. Понятийный аппарат освоен неудовлетворительно. Понимание материала фрагментарное или отсутствует. Неумение формулировать свои мысли, обсуждать дискуссионные положения.	2
Тестирование	86% правильных ответов и более.	От 71% до 85 % правильных ответов.	От 56% до 70% правильных ответов.	55% правильных ответов и менее.	3
Экзамен	Обучающийся обнаружил всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой дисциплины, усвоил взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявил творческие способности в понимании, изложении и использовании учебно-программного материала.	Обучающийся обнаружил полное знание учебно-программного материала, успешно выполнил предусмотренные программой задания, усвоил основную литературу, рекомендованную программой дисциплины, показал систематический характер знаний по дисциплине и способен к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.	Обучающийся обнаружил знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справился с выполнением заданий, предусмотренных программой, знаком с основной литературой, рекомендованной программой дисциплины, допустил погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладает необходимыми знаниями для их устранения под руководством преподавателя.	Обучающийся обнаружил значительные пробелы в знаниях основного учебно-программного материала, допустил принципиальные ошибки в выполнении предусмотренных программой заданий и не способен продолжить обучение или приступить по окончании университета к профессиональной деятельности без дополнительных занятий по соответствующей дисциплине.	

6.3 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

Приложение. Развёрнутое содержание оценочных средств - в прикреплённом файле
F_1158314110/B1.O.05._Praktikum_po_obektno_orientirovannomu_programmirovaniyu.pdf

Семестр 1

Текущий контроль

1. Лабораторные работы

Темы 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Тема 1. Методы и классы.

Модификаторы public и private, Алгоритм предотвращения сбоев, Передача методам объектов в качестве параметров, Аргументы простых типов передаются по значению, Передача объектов по ссылке, Модификатор ref для передачи методу значения обычного типа по ссылке, Замена местами значения двух переменных, Модификатор параметра out, Модификатор ref при вызове по ссылке, Модификатор params, Обычный параметр и параметр переменной длины, Возврат объектов, Возврат объектов, определенных пользователем, Перегрузка метода, Автоматическое преобразование типов аргументов, Перегрузка конструктора, Один объект используется для инициализации другого, Ключевое слово this для вызова одного конструктора из другого, Отображение аргументов, заданных в командной строке, Простой автоматизированный телефонный справочник, Рекурсия для вычисления факториала числа, Модификатор static.

Тема 2. Перегрузка операторов.

Перегруженные операторы, Перегрузка унарного ?, Перегрузка унарного ++, Перегрузка бинарного оператора +, Перегрузка операторов "<" и ">".

Тема 3. Индексаторы и свойства.

Индексатор для доступа к элементам массива, Индексаторы, работающие не с массивами, Двухмерный индексатор, Простое свойство, Свойство Length.

Тема 4. Наследование.

Простая иерархия классов, Новая характеристика в классе ? наследнике, Свойства для доступа к закрытым членам класса, Модификатор protected, Конструктор в классе ? наследнике, Конструктор в базовом классе, Скрытие переменной при наследовании, Ключевое слово base для доступа к скрытой переменной, Вызов скрытого метода с помощью ключевого слова base, Многоуровневая иерархия классов, Порядок вызова конструкторов, Ссылка переменной наследуемого класса на объект наследующего класса, Передача ссылочной переменной наследуемого класса ссылки на объект наследующего класса, Виртуальный метод, Виртуальные методы и полиморфизм, Абстрактный класс, Переопределение метода ToString(), Упаковка и распаковка переменной, Упаковка при передаче методу значений в качестве аргументов.

Тема 5. Интерфейсы, структуры и перечисления.

Реализация интерфейса, Интерфейсные ссылки, Свойства в интерфейсе, Индексатор в интерфейсе, Наследование интерфейсов, Явная реализация члена интерфейса, Явная реализация с целью устранения неоднозначности, Возможности структуры, Возможности перечисления.

Тема 6. Обработка исключительных ситуаций.

Обработка исключений, Генерация исключения в одном методе, а перехват его другим методом, Обработка ошибок системой выполнения C#, Обработка ошибок и продолжение выполнения программы, Несколько операторов catch, Оператор catch, перехватывающий все ошибки, Вложенный блок try, Генерирование исключения оператором throw, Повторное генерирование исключения, Блок finally, Члены класса Exception, Заказное исключение, Исключения наследующего класса должны следовать перед исключениями наследуемого класса, Использование ключевых слов checked и unchecked, Использование ключевых слов checked и unchecked при работе с блоками операторов.

Тема 7. Применение средства ввода-вывода.

Ввод с консоли методом ReadLine(), Поток Console.In для чтения массива байтов с клавиатуры, Запись данных в потоки Console.Out и Console.Error, Отображение содержимого файла, Копирование данных из файла в файл, Класс StreamWriter, Открытие файла с помощью класса StreamWriter, Возможности класса StreamReader, Перенаправление Console.Out, Запись и считывание двоичных данных, Произвольный доступ к содержимому файла, Расчет среднего значения введенных чисел.

Тема 8. Делегаты, события и лямбда выражения.

Объявление делегата, Делегаты ссылаются на методы экземпляра, Многоадресность делегата, Использование события, Широковещательное событие, Реагирование обработчиков каждого из объектов класса на событие.

Тема 9. Пространства имен, процессор и сборки.

Объявление пространства имен, Директива using, Использование псевдонима, Аддитивность пространств имен, Вложенное пространство имен, Оператор преобразования, Оператор явного преобразования.

Тема 10. Динамическая идентификация типов, рефлексия и атрибуты.

Директивы препроцессора #if, #endif и #define, Символьное выражение, Директива препроцессора #else, Директива препроцессора #elif, Условный атрибут Conditional, Атрибут Obsolete, Указатель и ключевое слово unsafe,

https://shelly.kpfu.ru/pls/student/docs/F_1158314110/B1.O.05._Praktikum_po_obektno_orientirovannomu_programmirovaniyu.p

2. Устный опрос

Темы 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

1. Переменные (Копирование переменных, Типы переменных, Действия с переменными, Определение типа переменной, Установка типа переменной)
2. Оператор присваивания
3. Ссылочные переменные (Жесткие ссылки, Символические ссылки, Ссылки на объекты)
4. Константы (Предопределенные константы, Определение констант, Проверка существования константы, Константы с динамическими именами)
5. Отладочные функции
6. Выражения (Логические выражения, Строковые выражения)
7. Операции (Арифметические операции, Строковые операции, Операции присваивания, Операции инкремента и декремента, Битовые операции, Операции сравнения, Логические операции, Операция отключения предупреждений, Условные операции)
8. Передача данных командной строки
9. Формы (Трансляция полей формы, Трансляция переменных окружения, Трансляция cookies)
10. Обработка списков
11. Обработка массивов
12. Диагностика
13. Порядок трансляции переменных
14. Инструкция if, else
15. Цикл с предусловием while
16. Цикл с постусловием do, while
17. Универсальный цикл for
18. Инструкции break и continue
19. Цикл foreach
20. Конструкция switch?case
21. Инструкции goto
22. Инструкции require и include
23. Инструкции однократного включения
24. Создание массива. Автомассивы
25. Конструкция list()
26. Списки и ассоциативные массивы
27. Конструкция array() и многомерные массивы
28. Массивы, константы
29. Операции над массивами (Доступ по ключу, Функция count(), Слияние массивов, Косвенный перебор элементов массива, Прямой перебор массива)
30. Списки и строки
31. Сериализация (Упаковка, Распаковка)

Темы 5, 6, 7, 8, 9

1. Класс как тип данных
2. Создание нового класса
3. Работа с классами (Создание объекта некоторого класса, Доступ к свойствам объекта, Доступ к методам, Создание нескольких объектов, Перегрузка преобразования в строку)
4. Инициализация и разрушение (Конструктор, Деструктор, Алгоритм сбора мусора)
5. Права доступа к членам класса (Модификаторы доступа, Неявное объявление свойств, Общие рекомендации)
6. Класс self, объект \$this (счетчик объектов, кэш ресурсов)
7. Константы класса
8. Перехват обращений к членам класса
9. Клонирование объектов (Переопределение операции клонирования, Запрет клонирования)
10. Перехват сериализации

11. Сериализация объектов (Упаковка и распаковка объектов, Методы sleep() и wakeup())
12. Расширение класса (Метод включения, Наследование, Переопределение методов, Константы __CLASS__ и __METHOD__, Позднее статическое связывание)
13. Анонимные классы
14. Полиморфизм (Абстрагирование, Виртуальные методы, Расширение иерархии, Абстрактные классы и методы, Совместимость родственных типов)
15. Интерфейсы (Наследование интерфейсов, Интерфейсы и абстрактные классы)
16. Трейты
17. Объявление пространства имен
18. Иерархия пространства имен
19. Импортирование
20. Автозагрузка классов (Функция __autoload(), Функция spl_autoload_register())
21. Что такое ошибка? (Роли ошибок, Виды ошибок)
22. Контроль ошибок (Директивы контроля ошибок, Установка режима вывода ошибок, Оператор отключения ошибок, Перехват ошибок, Генерация ошибок, Стек вызовов функций)
23. Исключения (Базовый синтаксис, Инструкция throw, Раскрутка стека, Исключения и деструкторы, Исключения и set error handler(), Классификация и наследование, Блоки финализаторы)
24. Трансформация ошибок (Серьезность "несерьезных" ошибок, Преобразование ошибок в исключения)

3. Тестирование

Темы 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

1. Какие парадигмы программирования поддерживает PHP
логическую
процедурную
объектно-ориентированную

2. Написать программу, которая выводит ?жирными? буквами (тег) строку ?Добро пожаловать!? с использованием языка PHP.

```
<?php echo  
"<b> Добро пожаловать! </b>";  
?>  
<?php <b>  
echo "Добро пожаловать!" </b>  
?>  
<?  
echo "<bold>Добро пожаловать! </bold>"  
?>
```

3. Известно, что настройки PHP можно сохранять/изменять не только в php.ini, но и в самих PHP скриптах, в файлах .htaccess и в файле настроек сервера httpd.conf. Где можно установить значение опции session.auto_start?

в php.ini или httpd.conf
в скрипте пользователя
в файле .htaccess

4. Какой оператор обозначает равенство значений в языке PHP?

```
:=  
=  
=:  
==
```

5. Как можно задать массив в языке PHP?

```
$arr[?a?,?b?,?c?] = ?q?;  
$arr (?0"=> ?a?);  
$arr = array(?a?,?b?,?c?);  
$arr[0] = ?a?;
```

6. Какие из утверждений относительно оператора require верны?

require выполняет код указанного в нем файла только один раз
require используется для включения в программу содержимого другого файла
при использовании внутри условных блоков require не нужно заключать в фигурные скобки

7. Каким будет результат выполнения следующей программы?

```
<?php
for ($j = 1;
{
$i = round (9/$j);
switch ($i)
{
case 5: echo "+?";
break;

case 9: echo "-?";
break;

case 3: echo "!?";
break 2;
default: echo $i;
break;
}
$j++;
}
?>
```

+-!
ошибка синтаксиса
бесконечный цикл
-+!

8. Дана команда: `if(!$var) echo "Hello?";` В каком случае на экран будет выведено слово "Hello"??

если `$var == true`
если `$var = 0`
если `$var` преобразуется к логическому `false`

9. Какие из операторов `switch` записаны правильно с точки зрения синтаксиса?

```
switch ($par){ "1": echo "1"; "2": echo 2;
}
switch ($par){
case "1": echo "1"; case 2: echo 2; default: echo 3;
}
switch ($par):
case "1": echo "1"; break;
case 2: echo 2; break; endswitch;
```

Файл `vars.php`: `<?php`

`$a = 1; ?>`

Файл `index.php`: `<?php`

`for ($l=0; $l , 3; $l++)`

```
{
include(?vars.php?);
echo $a++;
} ?>
```

10. Что мы получим в результате обработки интерпретатором файла `index.php`?

12
1234
123
111

11. В каком случае выполняется блок действий цикла

```
for (expr1; expr2; expr3) {
// блок действий
}
```

если второе выражение (expr2) вычисляется как true
 если третье выражение (expr3) вычисляется как true
 если первое выражение (expr1) вычисляется как true

12. Какое из утверждений относительно оператора include верно?
 файл, включаемый с помощью include, должен быть правильным HTML-файлом
 include используется для включения в программу содержимого другого файла
 файл, включаемый с помощью include, может быть любым файлом

13. Как можно узнать виртуальный путь до выполняющегося в данный момент скрипта?
 getenv(?SCRIPT_NAME?)
 \$_SERVER[?SCRIPT_NAME?]
 \$_SERVER[?PHP_SELF?]
 с помощью константы SCRIPT_NAME

14. Чем отличается клиент от сервера?
 клиент посылает запросы, а сервер обрабатывает их
 клиент отображает данные на экране компьютера пользователя, а сервер предоставляет данные
 сервер выполняет запросы клиента, специализируясь на эффективном решении задач определенного класса
 сервер есть часть компьютерной архитектуры сервер-сервер, а клиент ? архитектуры клиент-клиент
 сервер создает соединение, а клиент, пользуясь им, передает свой запрос

15. Как можно получить адрес страницы, с которой был послан запрос?
 \$_SERVER[?HTTP_REFERER?]
 getenv(?REMOTE_HOST?)
 getenv(?HTTP_REFERER?)
 с помощью константы HTTP_REFERER

16. В чем состоят функции сервера?
 обрабатывать запросы клиента
 запускать процессы, запрошенные клиентом, и возвращать клиенту результаты
 отображать данные на экране компьютера пользователя
 инициировать соединение с клиентом

Дана следующая html форма:

```
<form name="my_form" method="get">
Field 1 <input type="text" name="f1"><br>
Select field
<select name="f2">
<option value="1">first
<option value="2">second
</select><br>
<input type="submit">
</form>
```

17. Как можно получить переданные клиентом значения из этой формы?
 \$f1 и \$f2, если register_globals=off
 \$_POST[?f1 ?] и \$_POST[?f2?]
 \$_GET[?f1 ?] и \$_GET[?f2?]
 \$POST[?f1 ?] и \$POST[?f2?]
 \$_REQUEST[?f1 ?] и \$_REQUEST[?f2?]

18. Создать класс A, расширяющий класс B с помощью метода add() и переменной \$a.
 class B{
 var \$b;
 function B(){
 }
 }
 class B extends A {
 var \$a;

```

function add($b){
$this->a += $b;
}
}
class B{
var $b;
function B(){
}
}
class A extends B {
var $a;
function add($b){ $this->a += $b;
}
}
class A extends B {
var $a;
function add($b){
$this-> += $b;
}
}

```

19. Как можно получить и вывести на экран список всех методов класса, представителем которого является объект?

```

get_methods(get_class(объект));
get_class_methods(get_class(объект));
get_class_methods(get_class -> объект);
Класс MyClass задан следующим образом:

```

```

<?php
class MyClass{
var $a;
function MyClass() {
$this->a = "hello";
}
function MyFunc($b) {
return $b. " ". $this->a;
}
}
?>

```

20. Создать представителя класса MyClass. Получить и вывести значения всех свойств этого класса. Вызвать методы данного класса.

```

$myObj = new MyClass();
echo $myObj->a;
$str = $myObj->MyFunc("people");
Конструктор MyClass() вызывается автоматически при создании представителя класса.
new MyClass();
echo $MyClass->a;
$str = $MyClass->MyFunc("people");
Конструктор MyClass() вызывается автоматически при создании представителя класса.
$myObj = new MyClass("test");
$b = "a";
echo $myObj->$b;
MyFunc($b);

```

21. Имеется объект (\$obj) какого-то класса. Требуется получить список всех свойств класса, которому принадлежит объект.

```

$get_class_vars(get_class->$obj);
get_class_var(get_class($obj));
$vars = get_class_vars(get_class($obj));

```

22. Как можно программно узнать имя класса, представителем которого является объект?

```

$(get_class->объект);

```

```
get_class(объект);
class(объект);
```

23. Изменить значения переданных элементов, увеличив их в 10 раз. Использовать функцию `array_walk()`. Вывести значения элементов до и после применения `array_walk`

Имеется следующая форма:

```
<form action=task3.php>
Запись номер 1 <input type=checkbox name=id[] value=10> <br>
Запись номер 2 <input type=checkbox name=id[] value=20> <br>
Запись номер 3 <input type=checkbox name=id[] value=30> <br>
<input type=submit value="Отправить">
</form>
```

```
<?
print_r($_GET);
array_walk($_GET,"test");
function test(&$val,$key){
foreach ($val as $k=> $v)
$val[$k] = $v*10;
}
print_r($_GET);
?>
```

```
<?
print_r($_GET);
array_walk($_GET,"test");
function test(&$val,$key){
$val = $val*10;
}
print_r($_GET)
?>
```

```
<?
print_r($_POST);
array_walk($_POST,"test");
function test(&$val,$key){
foreach ($val as $k=> $v)
$val[$k] = $v*10;
}
print_r($_POST);
?>
```

24. Дана строка `?<h2>Наука ? двигатель прогресса!</h2>?`. Как можно вывести ее в браузер именно в таком виде, без форматирования согласно стандарту HTML?

```
echo htmlspecialchars("<h2>Наука - <font color=red>двигатель</font> прогресса!</h2>");
echo htmlentities("<h2>Наука ? <font color=red>двигатель</font> прогресса!</h2>");
echo htmlspecialchars("<h2>Наука - <font color=red>двигатель</font> прогресса!</h2>", ENT_COMPAT);
```

25. Дана строка текста: `?PHP ? очень простой язык. Я знаю PHP!!!?`. Заменить в ней все вхождения слова `?PHP?` на слово `?Perl?`.

```
<?php
$str = "PHP ? очень простой язык. Я знаю PHP!!!";
echo str_replace($str, "PHP", "Perl");
?>
```

```
<?php
$str = "PHP ? очень простой язык. Я знаю PHP!!!";
echo str_replace("PHP", "Perl", $str);
?>
```

```
<?php
$str = "PHP ? очень простой язык. Я знаю PHP!!!";
echo substr_replace("PHP", "Perl", $str);
?>
```

Экзамен

Вопросы к экзамену:

1. Управление доступом к членам класса
2. Передача объектов методам по ссылке
3. Использование модификаторов параметров ref и out
4. Использование переменного числа аргументов
5. Возврат объектов из методов
6. Перегрузка методов
7. Перегрузка конструкторов
8. Инициализаторы объектов
9. Необязательные аргументы
10. Именованные аргументы
11. Метод Main()
12. Применение ключевого слова static
13. Основы перегрузки операторов
14. Выполнение операций со встроенными в C# типами данных
15. Перегрузка операторов отношения
16. Перегрузка операторов true и false
17. Перегрузка логических операторов
18. Операторы преобразования
19. Индексаторы
20. Свойства
21. Применение модификаторов доступа в аксессорах
22. Применение индексаторов и свойств
23. Основы наследования
24. Доступ к членам класса и наследование
25. Конструкторы и наследование
26. Наследование и сокрытие имен
27. Создание многоуровневой иерархии классов
28. Порядок вызова конструкторов
29. Ссылки на базовый класс и объекты производных классов
30. Виртуальные методы и их переопределение
31. Применение абстрактных классов
32. Предотвращение наследования с помощью ключевого слова sealed
33. Класс object
34. Интерфейсы
35. Применение интерфейсных ссылок
36. Интерфейсные свойства
37. Интерфейсные индексаторы
38. Наследование интерфейсов
39. Сокрытие имен при наследовании интерфейсов
40. Явные реализации
41. Выбор между интерфейсом и абстрактным классом
42. Стандартные интерфейсы для среды .NET Framework
43. Структуры
44. Перечисления

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

В КФУ действует балльно-рейтинговая система оценки знаний обучающихся. Суммарно по дисциплине (модулю) можно получить максимум 100 баллов за семестр, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов.

Для зачёта:

56 баллов и более - "зачтено".

55 баллов и менее - "не зачтено".

Для экзамена:

86 баллов и более - "отлично".

71-85 баллов - "хорошо".

56-70 баллов - "удовлетворительно".

55 баллов и менее - "неудовлетворительно".

Форма контроля	Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций	Этап	Количество баллов
Семестр 1			
Текущий контроль			
Лабораторные работы	В аудитории, оснащённой соответствующим оборудованием, обучающиеся проводят учебные эксперименты и тренируются в применении практико-ориентированных технологий. Оцениваются знание материала и умение применять его на практике, умения и навыки по работе с оборудованием в соответствующей предметной области.	1	20
Устный опрос	Устный опрос проводится на практических занятиях. Обучающиеся выступают с докладами, сообщениями, дополнениями, участвуют в дискуссии, отвечают на вопросы преподавателя. Оценивается уровень домашней подготовки по теме, способность системно и логично излагать материал, анализировать, формулировать собственную позицию, отвечать на дополнительные вопросы.	2	20
Тестирование	Тестирование проходит в письменной форме или с использованием компьютерных средств. Обучающийся получает определённое количество тестовых заданий. На выполнение выделяется фиксированное время в зависимости от количества заданий. Оценка выставляется в зависимости от процента правильно выполненных заданий.	3	10
Экзамен	Экзамен нацелен на комплексную проверку освоения дисциплины. Экзамен проводится в устной или письменной форме по билетам, в которых содержатся вопросы (задания) по всем темам курса. Обучающемуся даётся время на подготовку. Оценивается владение материалом, его системное освоение, способность применять нужные знания, навыки и умения при анализе проблемных ситуаций и решении практических заданий.		50

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями и предоставленных доступов НЧИ КФУ;

- в печатном виде - в фонде библиотеки Набережночелнинского института (филиала) КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра (для обучающихся по ФГОС 3++ - не менее 0,25 экземпляра) каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов библиотеки Набережночелнинского института (филиала) КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Microsoft - <https://www.microsoft.com/ru-ru>

Программирование - <http://professorweb.ru/>

Программирование - <https://metanit.com/>

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

Вид работ	Методические рекомендации
лабораторные работы	Лабораторная работа требует от студентов не только наличия знаний, но еще и умений применять эти знания в новых ситуациях, сообразительности. Лабораторная работа активизирует познавательную деятельность студентов, выполняется на компьютере. Используется лабораторная работа для закрепления определенных навыков с программными средствами, когда кроме алгоритмических предписаний в задании студентов может получить консультации преподавателя. Подготовка к лабораторной работе включает 2 этапа: организационный; закрепление и углубление теоретических знаний. В процессе этой работы студент должен стремиться понять и запомнить основные положения материала, примеры, поясняющие его, а также разобраться в дополнительном материале. Использование дистанционных технологий в образовательном процессе.
самостоятельная работа	В процессе самостоятельной деятельности студент должен научиться выделять познавательные задачи, выбирать способы их решения, выполнять операции контроля за правильностью решения поставленной задачи, совершенствовать навыки реализации теоретических знаний. Формирование умений и навыков самостоятельной работы студентов может протекать как на сознательной, так и на интуитивной основе. Самостоятельная работа - деятельность обучающегося направленная на самостоятельное выполнение заданий. К самостоятельной работе относится: подготовка к практическим, лабораторным и семинарским занятиям, подготовка к устному опросу, подготовка к тестированию. Самостоятельная работа творческого характера позволяет развивать творческие способности студентов. Использование дистанционных технологий в образовательном процессе.
устный опрос	Устный опрос как контроль знаний студентов осуществляется в виде фронтальной и индивидуальной проверки. При фронтальном опросе за короткое время проверяется состояние знаний студентов всей группы по определенному вопросу или группе вопросов. Эта форма проверки используется для: выяснения готовности группы к изучению нового материала; определения сформированности понятий; проверки домашних заданий; поэтапной или окончательной проверки учебного материала, только что разобранный на занятии; при подготовке к выполнению практических и лабораторных работ. Индивидуальный устный опрос позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления, культуру речи студентов. Использование дистанционных технологий в образовательном процессе.
тестирование	Тест представляет собой кратковременное технически сравнительно просто составленное испытание, проводимое в равных для всех условиях. Различают следующие виды тестов. Избирательный тест состоит из системы заданий, к каждому из которых прилагаются как верные, так и неверные ответы. Из них студент выбирает тот, который считает верным для данного вопроса. При этом неверные ответы содержат такую ошибку, которую студент может допустить, имея определенные пробелы в знаниях. Закрытые тесты не содержат вариантов ответов. Студенты предлагают свой вариант ответа. Имеются тесты перекрестного выбора, в которых требуется установить соответствие между элементами множества ответов. Встречаются также тесты идентификации, в которых в качестве ответов приводятся графики, схемы, чертежи. Использование дистанционных технологий в образовательном процессе.
экзамен	Экзаменом заканчивается изучение предмета или его основного раздела. Наиболее распространённые формы экзамена: экзамен по билетам, экзамен-беседа, письменный экзамен, коллоквиум, тестовый контроль, программированный контроль. Успешно сдает экзамен тот, кто умеет периодически повторять материал, который был прослушан на лекциях, законспектирован и закреплен на самостоятельных занятиях. Повторение предполагает обобщение, углубление, а также расширение усвоенных знаний. При подготовке к экзамену необходимо опираться на лекции, а также на источники, которые разбирались на практических, семинарских и лабораторных занятиях. Использование дистанционных технологий в образовательном процессе.

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

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

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

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

Компьютер и принтер для распечатки раздаточных материалов.

Мультимедийная аудитория.

Компьютерный класс.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:
- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 09.04.03 "Прикладная информатика" и магистерской программе "Реинжиниринг бизнес-процессов предприятий, организаций, банков".

*Приложение 2
к рабочей программе дисциплины (модуля)
Б1.О.05 Практикум по объектно-ориентированному
программированию*

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 09.04.03 - Прикладная информатика

Профиль подготовки: Реинжиниринг бизнес-процессов предприятий, организаций, банков

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Основная литература:

1. Бунаков П. Ю. Практикум по решению задач на ЭВМ в среде Delphi : учебное пособие / П.Ю. Бунаков, А.К. Лопатин. - Москва : ФОРУМ : ИНФРА-М, 2018. - 304 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-00091-481-6. - URL: <https://znanium.com/catalog/product/939812> (дата обращения: 27.07.2020). - Текст : электронный.
2. Корнеев В. И. Программирование графики на C++. Теория и примеры : учебное пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. - Москва : ИД 'ФОРУМ' : ИНФРА-М, 2019. - 517 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0837-2. - URL: <https://znanium.com/catalog/product/1018909> (дата обращения: 27.07.2020). - Текст : электронный.
3. Белов В.В. Программирование в Delphi: процедурное, объектно-ориентированное, визуальное : учебное пособие для вузов / В.В. Белов, В.И.Чистякова. - 2-е изд., стереотип. - Москва : Горячая линия - Телеком, 2014. - 240 с. - ISBN 978-5-9912-0412-5. - URL : <https://www.studentlibrary.ru/book/ISBN9785991204125.html> (дата обращения: 27.07.2020). - Текст : электронный.

Дополнительная литература:

1. Медведев М. А. Программирование на СИ#: учебное пособие / М.А. Медведев, А.Н. Медведев. - 2-е изд., стер. - Москва : Флинта, Изд-во Урал. ун-та, 2017. - 64 с. - ISBN 978-5-9765-3169-7. - URL: <https://znanium.com/catalog/product/948428> (дата обращения: 27.07.2020). - Текст : электронный.
2. Кучунова Е. В. Программирование. Процедурное программирование: учебное пособие / Е.В. Кучунова, Б.В. Олейников, О.М. Чередниченко. - Красноярск: СФУ, 2016. - 92 с. - ISBN 978-5-7638-3555-7. - URL: <https://znanium.com/catalog/product/978627> (дата обращения: 27.07.2020). - Текст : электронный.
3. Хорев П. Б. Объектно-ориентированное программирование с примерами на С# : учебное пособие / П.Б. Хорев. - Москва : ФОРУМ : ИНФРА-М, 2020. - 200 с. - (Высшее образование: Бакалавриат). - ISBN 978-5-00091-680-3. - URL: <https://znanium.com/catalog/product/1069921> (дата обращения: 12.03.2021). - Текст : электронный.
4. Шакин В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.NET : учебное пособие / В.Н. Шакин, А.В. Загвоздкина, Г.К. Сосновиков. - Москва : ФОРУМ : ИНФРА-М, 2019. - 398 с. - (Высшее образование. Бакалавриат). - ISBN 978-5-00091-048-1. - URL: <https://znanium.com/catalog/product/1010028> (дата обращения: 06.10.2020). - Текст : электронный.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.О.05 Практикум по объектно-ориентированному
программированию

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 09.04.03 - Прикладная информатика

Профиль подготовки: Реинжиниринг бизнес-процессов предприятий, организаций, банков

Квалификация выпускника: магистр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2020

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7 Профессиональная или Windows XP (Volume License)

Пакет офисного программного обеспечения Microsoft Office 365 или Microsoft Office Professional plus 2010

Браузер Mozilla Firefox

Браузер Google Chrome

Adobe Reader XI или Adobe Acrobat Reader DC

Kaspersky Endpoint Security для Windows