МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Томский государственный университет систем управления и радиоэлектроники

УТВЕРЖДАЮ

Проректор по учебной работе

_____ **П.Е. Троян**

«_ _» _____ 2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Уровень основной образовательной программы: бакалавриат

Направление(я) подготовки (специальность): 09.03.03 – Прикладная информатика

Профиль: Прикладная информатика в экономике

Форма обучения: очная

Факультет: ФСУ, Факультет систем управления

Кафедра: АСУ, Кафедра автоматизированных систем управления

Курс 2 Семестр 3, 4

Учебный план набора 2014, 2015 г.

Распределение рабочего времени:

Виды учебной работы	Семестр 3	Семестр 4	Всего	Единицы
Лекции	**18**	**18**	**36**	**часов**
Лабораторные работы	**54**		**54**	**часов**
Практические занятия		**18**	**18**	**часов**
Курсовой проект/работа (КСР) (аудиторная)		**8**	**8**	**часов**
Всего аудиторных занятий	**72**	**44**	**116**	**часов**
Из них в интерактивной форме	**14**	**8**	**22**	**часов**
Самостоятельная работа студентов (СРС)	**36**	**100**	**136**	**часов**
Всего (без экзамена)	**108**	**144**	**252**	**часов**
Самост. работа на подготовку и сдачу экзамена		**36**		**часов**
Общая трудоемкость	**108**	**180**	**288**	**часов**
(в зачетных единицах)	**3**	**5**	**8**	**ЗЕТ**

Зачет 3 семестр

Курсовой проект 4 семестр

Экзамен 4 семестр

Томск 2017

Согласована на портале № 15935

Рассмотрена и одобрена на заседании кафедры
протокол № 1 от « 12 » 1 2017 г.

Рабочая программа составлена с учетом требований Федерального Государственного образовательного стандарта высшего профессионального образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика (квалификация (степень) "бакалавр"), утвержденного Приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. № 207, рассмотрена и утверждена на заседании кафедры 24 января 2017 г., протокол № 2.

Разработчик ассистент каф. АСУ _____ С.М. Алфёров

Зав. обеспечивающей кафедрой АСУ
д.т.н., профессор _____ А.М. Кориков

Рабочая программа согласована с факультетом, профилирующей и выпускающей кафедрами специальности.

Кандидат технических наук, доцент
Декан, к.т.н., доцент _____ П.В. Сенченко

Заведующий профилирующей и выпускающей
кафедрой АСУ, д.т.н., профессор _____ А.М. Кориков

Эксперты:

Кафедра АСУ, доцент _____ А.И. Исакова

Согласована на портале № 15935

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Целью **дисциплины** «Объектно-ориентированное программирование является обучение студентов основам объектно-ориентированного программирования и проектирования.

Основные задачи дисциплины «Объектно-ориентированное программирование**:**

1) изучение языка UML для проектирования объектной структуры программы;
2) изучение средств языка C++ для создания объектной структуры программы;
3) получение знаний и практических навыков в области проектирования и разработки объектно-ориентированных программ.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Дисциплина «Объектно-ориентированное программирование» относится к числу дисциплин профессионального цикла. Успешное овладение дисциплиной предполагает предварительные знания, полученные в предыдущих дисциплинах: «Информатика и программирование»; «Основы алгоритмизации и языки программирования»; «Дискретная математика». Знания, полученные студентами в этой дисциплине, будут использоваться при изучении дисциплин: «Операционные системы», «Программная инженерия».

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование следующих

профессиональных компетенций (ПК):

– способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение (**ПК**-2)

– способностью программировать приложения и создавать программные прототипы решения прикладных задач (**ПК**-8).

В результате изучения дисциплины студент должен:

Знать: следующие понятия и определения: объектно-ориентированная парадигма, класс, объект, доступ, поля и методы, проектирование объектно-ориентированных программ, объектно-ориентированные языки, архитектура.

Уметь: программировать, используя объектно-ориентированный подход на языках программирования C++, а также умение описывать и читать архитектуру классов и объектов на языке UML.

Владеть: практическими навыками по проектированию и созданию объектно-ориентированной структуры программы, созданию новых классов, эффективному использованию готовых классов.

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общая трудоемкость дисциплины составляет 8 зачетных единиц.

Вид учебной работы	Всего часов	Семестры	
		3	4
Аудиторные занятия (всего)	**116**	**72**	**44**
В том числе:	**-**	**-**	**-**
Лекции	**36**	**18**	**18**
Лабораторные работы (ЛР)	**54**	**54**	
Практические занятия (ПЗ)	**18**		**18**
Семинары (С)			
Коллоквиумы (С)			
Курсовой проект/(работа) (аудиторная нагрузка)	**8**		**8**
Другие виды аудиторной работы			
Самостоятельная работа (всего)	**136**	**36**	**100**
В том числе:			
Курсовой проект (работа) (самостоятельно)	**64**		**64**
Реферат			
Другие виды самостоятельной работы:			
Проработка лекционного материала	**18**	**9**	**9**
Лабораторные работы и подготовка отчетов	**18**	**18**	
Подготовка к практическим занятиям	**18**		**18**
Самостоятельное изучение тем	**18**	**9**	**9**
Подготовка к экзамену	**36**		**36**
Вид промежуточной аттестации (зачет, экзамен)			
Общая трудоемкость час	**288**	**108**	**180**
зач. ед.(до сотых долей)	**8**	**3**	**5**

Согласована на портале № 15935

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1. Разделы дисциплин и виды занятий

№ п/п	Наименование раздела дисциплины	Лекц.	Лаб. зан.	Практ. зан.	СРС	Всего час. (без экзам)	Формируемые компетенции (ПК)
Семестр 3							
1.	Введение	1			2	3	ПК-2, ПК-8
2.	Ввод/вывод на основе ООП	1	6		2	9	ПК-2, ПК-8
3.	Обработка исключений: работа с ошибками	1	2		2	5	ПК-2, ПК-8
4.	Скрытие реализации, инкапсуляция	2	6		4	12	ПК-2, ПК-8
5.	Повторное использование классов, наследование	2	6		4	12	ПК-2, ПК-8
6.	Полиморфизм	2	6		4	12	ПК-2, ПК-8
7.	Использование объектов как атрибутов класса. Интерфейсы и внутренние классы, агрегирование	2	6		4	12	ПК-2, ПК-8
8.	Шаблоны функции и шаблоны классов	2	6		4	12	ПК-2, ПК-8
9.	Модульное программирование	1	4		2	7	ПК-2, ПК-8
10.	Создание окон и приложений	2	6		4	12	ПК-2, ПК-8
11.	Многопоточность, таймеры (win API)	2	6		4	12	ПК-2, ПК-8
ИТОГО		**18**	**54**		**36**	**108**	
Семестр 4							
12.	Создание пользовательского интерфейса	1		2	2	4	ПК-2, ПК-8
13.	Шаблон проекта	1			2	4	ПК-2, ПК-8
14.	Компоненты интерфейса, их свойства и события	6		6	12	24	ПК-2, ПК-8
15.	Контейнеры, примеры наследования	2		2	4	8	ПК-2, ПК-8
16.	Компонент таймер, класс-поток	2		2	4	8	ПК-2, ПК-8
17.	Программное формирование интерфейса, примеры наследования от компонентов	3		3	6	12	ПК-2, ПК-8
18.	Сообщения, создание собственных событий и их обработка	3		3	6	12	ПК-2, ПК-8
ИТОГО		**18**		**18**	**36**	**72**	

5.2. Содержание разделов дисциплины (по лекциям)

№ п/п	Наименование разделов	Содержание разделов	Трудо-емкость (час.)	Формируемые компетенции (ПК)
Семестр 3				
1.	Введение	Развитие абстракции. Основные понятия и определения. Элементы языка UML.	1	ПК-2, ПК-8
2.	Ввод/вывод на основе ООП	Классы ввода/вывода, работа с консолью, с файлами. Способ ввода, вывода в оконных приложениях. Создание объектов (экземпляров класса). Способы создания объектов (статическое и динамическое) и массивов объектов (статическое и динамическое)	1	ПК-2, ПК-8
3.	Обработка исключений: работа с ошибками	Операторы обработки исключений. Средства отладки в средах программирования.	1	ПК-2, ПК-8
4.	Скрытие реализации, инкапсуляция	Объявление простого класса. Объявление класса, члены класса (атрибуты, методы), модификаторы доступа, способы описания методов внутри класса и вне класса. Необходимость в скрытии некоторых членов класса, понятие целостности объекта и её подержание. Конструкторы с параметрами, копирования и по умолчанию, деструктор, указатель this, статические атрибуты и методы. Отображение класса в UML.	2	ПК-2, ПК-8

Согласована на портале № 15935

5.	Повторное использование классов, наследование	Наследование. Вызов конструктора и прочих методов родительского класса из методов дочернего класса. Вызов методов родительского и дочернего класса из объектов (из вне). Динамическое создание объектов родительского и дочернего классов.	2	ПК-2, ПК-8
6.	Полиморфизм	Динамическое создание объектов родительского и дочернего классов. Виртуальные методы. Механизм позднего связывания.	2	ПК-2, ПК-8
7.	Использование объектов как атрибутов класса (агрегирование и композиция). Интерфейсы и внутренние классы.	Организация конструкторов и деструкторов при агрегировании и композиции. Способы передачи объектов в/из подпрограммы, параметры const. Понятие интерфейса (полностью абстрактного класса) и создания интерфейса. Отображение интерфейсов в UML.	2	ПК-2, ПК-8
8.	Шаблоны функций. Шаблоны классов	Необходимость в шаблонах. Описание и использование шаблонов функций. Область применения шаблонов классов. Описание и использование шаблонов классов. Отображение шаблонов в UML. «Подводные камни» шаблонов.	2	ПК-2, ПК-8
9.	Модульное программиро-вание	Пространства имен. Модули. Заголовочные файлы. Файлы исходного кода. Отображение в UML модулей и пространств имен.	1	ПК-2, ПК-8
10.	Создание окон и приложений	Создание окон и элементов управления различными средствами. События и обработка событий.	2	ПК-2, ПК-8
11.	Многопоточность	Создание потоков. Разделяемые ресурсы. Организация разделяемых ресурсов. Использование разделяемых ресурсов несколькими потоками. Распределение потоков по ядрам.	2	ПК-2, ПК-8
ИТОГО			**18**	
Семестр 4				
12.	Создание пользовательского интерфейса	Описание среды программирования, создание заготовки проекта.	1	ПК-2, ПК-8
13.	Проектирование .Шаблон проекта	Оформление технического задания. Рекомендации к построению объектной структуры.	1	ПК-2, ПК-8
14.	Компоненты интерфейса, их свойства и события	Метки, кнопки, поля ввода, поля выбора, таблицы.	6	ПК-2, ПК-8
15.	Контейнеры, примеры наследования	Использование класса List в качестве базового для класса хранения табличной информации определенного формата, построение производного класса для удобного использования.	2	ПК-2, ПК-8
16.	Компонент таймер, класс-поток	Создание таймеров, примеры использования таймеров. Создание потоков, примеры использования потоков. «подводные камни» при использовании таймеров и потоков.	2	ПК-2, ПК-8
17.	Программное формирование интерфейса, примеры наследования от компонентов	Создание объектов. Методы для установки объектов интерфейса на форму. Примеры создания интерфейсов с массивами объектов. Программное задание обработчика события. Использование общего обработчика события для всех элементов массива.	3	ПК-2, ПК-8
18.	Сообщения, создание собственных событий и их обработка	Функции WinAPI: SendMessage, PostMessage, GetMessage, PeekMessage.	3	ПК-2, ПК-8
ИТОГО			**18**	

Согласована на портале № 15935

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

№ п/п	Наименование обеспечивающих (предыдущих) дисциплин	№ № разделов данной дисциплины из табл.5.1, для которых необходимо изучение обеспечивающих (предыдущих) дисциплин																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1.	Дискретная математика	+			+			+	+										
2.	Информатика и программирование		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
3.	Основы алгоритмизации и языки программирования		+	+	+	+	+	+	+	+	+	+		+	+	+	+	+	+

№ п/п	Наименование обеспечивающих (последующих) дисциплин	№ № разделов данной дисциплины из табл.5.1, для которых необходимо изучение обеспечиваемых (последующих) дисциплин																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1.	Программная инженерия	+			+			+	+										
2.	Операционные системы		+	+					+	+	+	+	+	+	+	+	+	+	+

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

Перечень компетенций	Виды занятий					Формы контроля
	Л	Лаб	Пр.	КР/КП	СРС	
ПК-2	+	+	+	+	+	Устный опрос на лекции, защита отчета по лаб., проверка задний по курсовому проекту, контрольная работа
ПК-8	+			+	+	Опрос на лекции (устно), отчет по лаб. работе, защита ПЗ по курсовому проекту, тест по теме для сам. изучения

Л – лекция, Пр – практические и семинарские занятия, Лаб – лабораторные работы, КР/КП – курсовая работа/проект, СРС – самостоятельная работа студента.

6. МЕТОДЫ И ФОРМЫ ОРГАНИЗАЦИИ ОБУЧЕНИЯ

Технологии интерактивного обучения при разных формах занятий в часах[1]

Формы / Методы	Лекции (час)	Лабораторные работы	Всего
Обратная связь	**2**		**2**
Исследовательский метод	**4**		**4**
Решение ситуационных задач		**5**	**5**
Диалог	**2**		**2**
Лекция на проекторе	**9**		**9**
Итого интерактивных занятий	**17**	**5**	**22**

Примечание.

Обратная связь: в конце лекции дается простая задача, преподаватель может проверить уровень освоения материала.

Исследовательский метод: перед началом лекции преподаватель дает задачу, в процессе решения которой, студент уясняет проблему, решение которой будет рассказываться на лекции.

Решение ситуационных задач: студенту дается задача, имеющая практическое значение, например, в лаб. работе №1:

Диалог происходит со студентами на лекции (задаются вопросы в обе стороны).

Лекция на проекторе: лекция в компьютерном классе на проекторе с показом примеров программ, их объяснением, запуском и показом результатов работы. Студенты видят как те или иные структуры программы влияют на успешность компиляции и на работу программы.

[1] Используются смешанные методы, указана доля использования метода

Согласована на портале № 15935

7. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Методические указания к выполнению лабораторных работ и задания по ним по дисциплине содержатся в разделе 12.3 [2].

№ п/п	№ раздела дисциплины из табл. 5.1	Наименование лабораторных работ (семестр 3)	Трудо-емкость (час.)	Компетенции ПК
1.	2	Потоковые классы ввода/вывода для работы с консолью и файлами.	4	ПК-2, ПК-8
2.	3	Обработка исключений.	4	ПК-2, ПК-8
3.	4	Инкапсуляция	6	ПК-2, ПК-8
4.	5	Наследование	6	ПК-2, ПК-8
5.	6	Полиморфизм.	6	ПК-2, ПК-8
6.	7	Агрегирование и композиция.	6	ПК-2, ПК-8
7.	8	Шаблоны.	6	ПК-2, ПК-8
8.	9	Модульное программирование.	4	ПК-2, ПК-8
9.	10	Создание оконных приложений.	6	ПК-2, ПК-8
10.	11	Многопоточность.	6	ПК-2, ПК-8
ИТОГО			**54**	

8. ПРАКТИЧЕСКИЕ ЗАНЯТИЯ (СЕМИНАРЫ)

Методические указания к выполнению практических работ и задания по ним по дисциплине содержатся в разделах 12.2 [5] и 12.3 [3].

№ п/п	№ раздела дисциплины из табл. 5.1	Наименование практических занятий (семестр 4)	Трудо-емкость (час.)	Компетенции ПК
1.	12	Создание пользовательского интерфейса	2	ПК-2, ПК-8
2.	14	Компоненты интерфейса, их свойства и события	2	ПК-2, ПК-8
3.	15	Контейнеры, примеры наследования	4	ПК-2, ПК-8
4.	16	Компонент таймер, класс-поток	4	ПК-2, ПК-8
5.	17	Программное формирование интерфейса, примеры наследования от компонентов	4	ПК-2, ПК-8
6.	18	Сообщения, создание собственных событий и их обработка	2	ПК-2, ПК-8
ИТОГО			**18**	

9. САМОСТОЯТЕЛЬНАЯ РАБОТА

№ п/п	№ раздела дисциплины из табл. 5.1	Тематика самостоятельной работы *(детализация)*	Трудо-емкость (час.)	Компетенции ПК	Контроль выполнения работы
1	1÷18	Проработка лекционного материала	18	ПК-2, ПК-8	Опрос (устно) на лекции
2	2÷11	Подготовка к лабораторным работам и написание отчетов	18	ПК-2, ПК-8	Защита ЛР, отчет
	12÷18	Подготовка к практическим занятиям	18	ПК-2, ПК-8	Контрольная работа
4	10	Самостоятельное изучение тем теоретической части	18	ПК-2, ПК-8	Тест
5	1÷18	Подготовка и сдача экзамена	36	ПК-2, ПК-8	Экзамен
6	1÷18	Выполнение заданий по курсовому проекту (КП)	64		Защита КП
ИТОГО (вместе с экзаменом)			**172**		

Темы для самостоятельного обучения (18 часов)

1) Перегрузка унарных и бинарных операций (6 часа).
2) Создание различных элементов управления оконного интерфейса приложений средствами API, MFC, VCL. Обработка событий мыши, клавиатуры, воздействий на элементы управления, создание пользовательских событий (12 часов).

10. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ)

Методические указания к выполнению курсового проекта содержатся в разделе 12.3 [1].

Примерные темы курсовых работ перечислены ниже:

1) Телефонный справочник.
2) Записная книжка.
3) Видеотека.

Согласована на портале № 15935

4) Автомастерская.
5) Простой графический редактор (векторный или растровый).
6) Шашки.
7) Городки.
8) Морской бой на поле 10х10.
9) Морской бой. Торпедирование движущихся кораблей.
10) Теннис.
11) Тетрис.
12) Гонки.
13) Футбол
14) Экономическая игра с компьютером.

Тема утверждается преподавателем и разрабатывается студентом самостоятельно. Во время практического занятия студент может получить индивидуальную консультацию по проблемам, возникшим на соответствующем этапе проекта.

Контроль самостоятельной работы над проектом осуществляется по следующей схеме.

1. Утверждение темы проекта — 1 час.
2. Утверждение задания на проектирование — 1 час.
3. Утверждение промежуточных отчётов — 6 часов.

Всего — 8 часов.

11. РЕЙТИНГОВАЯ СИСТЕМА ДЛЯ ОЦЕНКИ УСПЕВАЕМОСТИ СТУДЕНТОВ

Курс 2, семестр 3 Контроль обучения – Зачет.

Максимальный семестровый рейтинг – **100 баллов.**

Таблица 11.1 – Балльные оценки для элементов контроля.

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
Посещение занятий	3	3	3	**9**
Тестовый контроль	12	12	12	**36**
Лабораторные работы	0	14	28	**42**
Компонент своевременности	0	6	7	**13**
Итого максимум за период:	**15**	**35**	**50**	**100**
Нарастающим итогом	**15**	**50**	**100**	

Студент, набравший менее 60 баллов в течение семестра, считается неуспевающим. Студент, выполнивший все запланированные лабораторные работы и набравший сумму 60 и более баллов, получает зачёт «автоматом».

Курс 2, семестр 4 Контроль обучения – Дифференцированный зачет.

Максимальный семестровый рейтинг – **100 баллов.**

Таблица 11.2 – Балльные оценки для элементов контроля.

Элементы учебной деятельности	Макс. балл на КТ1 с начала семестра	Макс. балл за период между КТ1 и КТ2	Макс. балл за период от КТ2 до конца семестра	Всего за семестр
Разработка технического задания	6			6
Выполнение заданий по КП		18	18	36
Оформление ПЗ			10	10
Компонент своевременности	6	6	6	18
Итого максимум за период	**12**	**24**	**34**	**70**
Защита проекта (макс.)				**30**
Нарастающим итогом	**12**	**36**	**70**	**100**

Курс 2, семестр 4 Контроль обучения – Экзамен.

Максимальный семестровый рейтинг – 100 баллов.

Согласована на портале № 15935

Таблица 11.3 – Балльные оценки для элементов контроля.

Элементы учебной деятельности	Макс. балл на КТ1 с начала семестра	Макс. балл за период между КТ1 и КТ2	Макс. балл за период от КТ2 до конца семестра	Всего за семестр
Посещение занятий	3	3	3	9
Тестовый контроль	8	8	8	24
Выполнение и защита промежуточных отчётов	5	10	10	25
Компонент своевременности	4	4	4	12
Итого максимум за период	**20**	**25**	**25**	**70**
Экзамен (максимум)				**30**
Нарастающим итогом	**20**	**46**	**70**	**100**

Таблица 11.4 – Пересчёт баллов в оценки за контрольные точки

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

Таблица 11.5 – Пересчёт суммы баллов в традиционную и международную оценку

Оценка (ГОС)	**Итоговый рейтинг**	**Оценка (ECTS)**
5 (отлично)	90 - 100	A (отлично)
4 (хорошо)	85 – 89	B (очень хорошо)
	75 – 84	C (хорошо)
	70 - 74	D (удовлетворительно)
3 (удовлетворительно)	65 – 69	
	60 - 64	E (посредственно)
2 (неудовлетворительно)	Ниже 60 баллов	F (неудовлетворительно)

Согласована на портале № 15935

12. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

12.1. Основная литература

1. Павловская, Т.А. C/C++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. - СПб.: Питер, 2013. - 461 с. (36 экз. библиотека ТУСУР)

12.2. Дополнительная литература

1. Лаптев, В.В. C++. Объектно-ориентированное программирование : учебное пособие / В. В. Лаптев. - СПб. : Питер, 2008. - 464 с. (3 экз.)
2. Катаев, М.Ю. Объектно-ориентированное программирование : Учебное пособие / М. Ю. Катаев, А. Я. Суханов ; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. - Томск : ТМЦДО, 2007. - 160 с. (12 экз.)
3. Объектно-ориентированное программирование: Учебное пособие / Романенко В. В. – 2014. 475 с. – [Электронный ресурс]. – https://edu.tusur.ru/training/publications/4872
4. Объектно-ориентированное программирование: Курс лекций / Ганджа Т. В., Панов С. А. – 2015. 110 с. [Электронный ресурс]. – https://edu.tusur.ru/training/publications/5009
5. Лаптев, В.В. C++. Объектно-ориентированное программирование. Задачи и упражнения : учебное пособие для вузов / В. В. Лаптев, А. В. Морозов, А. В. Бокова. - СПб. : Питер, 2007. - 288 с. (4 экз.)

12.3. Перечень пособий, методических указаний и материалов, используемых в учебном процессе

1. Алфёров С.М. Объектно-ориентированное программирование: Методические указания по курсовой, самостоятельной и индивидуальной работе студентов всех форм обучения для направления бакалавриата 230700 – Прикладная информатика/ С.М. Алфёров. – Томск: ТУСУР, 2015. – 12 с. – [Электронный ресурс]. –
http://asu.tusur.ru/learning/bak230700/d43/b230700_d43_work.doc
2. Объектно-ориентированное программирование: Методические указания к лабораторным работам / Ганджа Т.В., Панов С.А., 2015. – 102 с. – [Электронный ресурс]. – https://edu.tusur.ru/training/publications/5010
3. Павловская, Татьяна Александровна. C++. Объектно-ориентированное программирование. Практикум: учебное пособие для вузов / Т. А. Павловская, Ю. А. Щупак. - СПб.: Питер, 2005. – 464 с. (15 экз. библиотека ТУСУР)
4. Катаев, М.Ю. Объектно-ориентированное программирование: Учебное методическое пособие / М. Ю. Катаев ; Федеральное агентство по образованию, Томский государственный университет систем управления и радиоэлектроники, Кафедра автоматизированных систем управления. - Томск : ТМЦДО, 2007. - 45 с. (8 экз.)

12.4. Базы данных, информационно-справочные и поисковые системы

Программное обеспечение: ОС Windows, Среда программирования Visual Studio C++.

12.5 Учебно-методические пособия для лиц с ограниченными возможностями здоровья

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;

в печатной форме

Согласована на портале № 15935

13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

13.1. Общие требования к материально-техническому обеспечению дисциплины

13.1.1. Материально-техническое обеспечение для лекционных занятий

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используется учебная аудитория, с количеством посадочных мест не менее 22-24, оборудованная доской и стандартной учебной мебелью. Имеются наглядные пособия в виде презентаций по лекционным разделам дисциплины.

13.1.2. Материально-техническое обеспечение для лабораторных работ

Для проведения лабораторных занятий используется учебно-исследовательская вычислительная лаборатория, расположенная по адресу 634034, Томская область, г. Томск, Вершинина улица, д. 74, 4 этаж, ауд. 437, 438, 439. Состав оборудования: Учебная мебель; Экран с электроприводом DRAPER BARONET – 1 шт.; Мультимедийный проектор TOSHIBA – 1 шт.; Компьютеры класса не ниже Intel Pentium G3220 (3.0GHz/4Mb)/4GB RAM/ 500GB с широкополосным доступом в Internet, с мониторами типа Samsung 18.5" S19C200N– 10 шт.; Используется лицензионное программное обеспечение, пакеты версией не ниже: Microsoft Windows XP Professional with SP3; Visual Studio 2008 EE with SP1; Microsoft Office Visio 2010; Microsoft SQL-Server 2005; Matlab v6.5.

13.1.3. Материально-техническое обеспечение для самостоятельной работы

Для самостоятельной работы используется учебная аудитория (компьютерный класс), расположенная по адресу 634034, г. Томск, ул. Вершинина, 74, 1 этаж, ауд. 100. Состав оборудования: учебная мебель; компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 4 шт.; компьютеры подключены к сети ИНТЕРНЕТ и обеспечивают доступ в электронную информационно-образовательную среду университета.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья

Освоение дисциплины лицами с ОВЗ осуществляется с использованием средств обучения общего и специального назначения.

При обучении студентов **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах для студентов с нарушениями слуха, мобильной системы обучения для студентов с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой обучаются студенты с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При обучении студентов **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для удаленного просмотра.

При обучении студентов **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах для студентов с нарушениями опорно-двигательного аппарата, мобильной системы обучения для людей с инвалидностью.

13. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

14.1. Основные требования к фонду оценочных средств и методические рекомендации

Фонд оценочных средств и типовые контрольные задания, используемые для оценки сформированности и освоения закрепленных за дисциплиной компетенций при проведении текущей, промежуточной аттестации по дисциплине приведен в приложении к рабочей программе.

14.2 Требования к фонду оценочных средств для лиц с ограниченными возможностями здоровья

Для студентов с инвалидностью предусмотрены дополнительные оценочные средства, перечень которых указан в таблице 14.1.

Таблица 14.1 – Дополнительные средства оценивания для студентов с инвалидностью

Категории студентов	Виды дополнительных оценочных средств	Формы контроля и оценки результатов обучения

Согласована на портале № 15935

С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно- двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами, исходя из состояния обучающегося на момент проверки

14.3 Методические рекомендации по оценочным средствам для лиц с ограниченными возможностями здоровья

Для студентов с ОВЗ предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Студентам с инвалидностью увеличивается время на подготовку ответов на контрольные вопросы. Для таких студентов предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:
- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:
- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:
- в форме электронного документа;
- в печатной форме.

При необходимости для обучающихся с инвалидностью процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 15935

Приложение к рабочей программе

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

УТВЕРЖДАЮ

Проректор по учебной работе

_____ **П. Е. Троян**

«___» _____ 2017 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

Уровень основной образовательной программы бакалавриат

Направление подготовки 09.03.03 – Прикладная информатика

Профиль(и) Прикладная информатика в экономике

Форма обучения очная

Факультет систем управления

Кафедра автоматизированных систем управления

Курс 2

Семестр 3, 4

Учебный план набора 2014, 2015 гг.

Зачет 3 семестр

Экзамен 4 семестр

Курсовой проект 4 семестр

Томск 2017

Согласована на портале № 15935

1. ВВЕДЕНИЕ

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

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

Перечень закрепленных за дисциплиной «**Объектно-ориентированное программирование»** компетенций приведен в таблице 1.

Таблица 1 – Перечень закрепленных за дисциплиной компетенций

Код	Формулировка компетенции	Этапы формирования компетенции
ПК-2	Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение	***Знать:*** элементы унифицированного языка моделирования (UML); ***Уметь:*** построить объектно-ориентированную модель предметной области; ***Владеть:*** навыками разработки и отладки программ в современных средах.
ПК-8	Способностью программировать приложения и создавать программные прототипы решения прикладных задач	***Знать:*** конструкции и средства современных языков программирования относящихся к объектно-ориентированному стилю программирования; ***Уметь:*** писать код программы по заданным UML диаграммам; ***Владеть:*** современными технологиями построения приложений, грамотно модифицировать имеющие средства (модули программ, структуры данных) для решения прикладных задач.

2. РЕАЛИЗАЦИЯ КОМПЕТЕНЦИЙ

2.1 Компетенция ПК-2

ПК-2: способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение.

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания, представлены в таблице 2.

Таблица 2 – Этапы формирования компетенции ПК-2 и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	**Благодаря способности разрабатывать, внедрять и адаптировать прикладное программное обеспечение знать:** – элементы унифицированного языка моделирования (UML)	**Благодаря способности разрабатывать, внедрять и адаптировать прикладное программное обеспечение уметь:** – строить объектно-ориентированную модель предметной области	**Благодаря способности разрабатывать, внедрять и адаптировать прикладное программное обеспечение владеть:** – навыками разработки и отладки программ в современных средах
Виды занятий	– Лекции. – Лабораторные работы. – Курсовая работа. – Самостоятельная работа студентов.	– Лабораторные работы. – Практические занятия. – Курсовая работа. – Самостоятельная работа студентов.	– Лабораторные работы. – Практические занятия. – Курсовая работа. – Самостоятельная работа студентов.
Используемые средства оценивания	– Опрос; – Контрольная работа; – Зачет; – Экзамен.	– Устная защита лабораторных работ; – Модификация программы по заданию преподавателя и под его непосредственным наблюдением	– Устная защита лабораторных работ; – Модификация программы по заданию преподавателя и под его непосредственным наблюдением

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 3.

Таблица 3 – Общие характеристики показателей и критериев оценивания компетенции по этапам

Показатели и критерии	Знать	Уметь	Владеть

Согласована на портале № 15935

ОТЛИЧНО (высокий уровень)	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем	Контролирует работу, проводит оценку, совершенствует действия работы
ХОРОШО (базовый уровень)	Знает факты, принципы, процессы, общие понятия в пределах изучаемой области	Обладает диапазоном практических умений, требуемых для решения определенных проблем в области исследования	Берет ответственность за завершение задач в исследовании, приспосабливает свое поведение к обстоятельствам в решении проблем
УДОВЛЕТВОРИТЕЛЬНО (низкий уровень)	Обладает низким уровнем общих знаний	Обладает умениями на низком уровне, которые не достаточны для выполнения даже простых задач	Работает только при прямом наблюдении

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания представлены в таблице 4.

Таблица 4 – Этапы формирования компетенции и используемые средства оценивания

Показатели и критерии	Знать	Уметь	Владеть
ОТЛИЧНО (высокий уровень)	– На высоком уровне знает принципы проектирования, объектной декомпозиции.	– На высоком уровне умеет грамотно разбивать задачу на параллельные процессы; – Умеет эффективно использовать виртуальные методы, абстрактные классы.	– На высоком уровне владеет UML для объектно-ориентирования анализа и проектирования.
ХОРОШО (базовый уровень)	– Знает типовые приемы используемые при создании классов и иерархии классов; – Знает типовые шаблоны проектирования.	– Умеет хорошо строить иерархию классов; – Умеет эффективно использовать модификаторы доступа.	– Способен хорошо модифицировать готовые классы для эффективного решения задач; – Способен использовать свои и готовые классы для решения типовых задач.
УДОВЛЕТВО-РИТЕЛЬНО (низкий уровень)	– Знает общие представления об объектно-ориентированном программировании; – Понимает важность роли объектно-ориентированного программирования.	– Умеет создавать простые структуры, классы; – Умеет создавать простые объекты своих классов и готовых классов.	– Способен использовать готовые классы для решения простых задач.

2.2 Компетенция ПК-8

ПК-8: способностью программировать приложения и создавать программные прототипы решения прикладных задач.

Согласована на портале № 15935

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

Таблица 5 – Этапы формирования компетенции ПК-8 и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	**Благодаря способности программировать приложения и создавать программные прототипы знает:** – конструкции и средства современных языков программирования относящихся к объектно-ориентированному стилю программирования;	– **Благодаря способности программировать приложения и создавать программные прототипы умеет:** – писать код программы по заданным UML диаграммам;	**Благодаря способности программировать приложения и создавать программные прототипы владеет:** – современными технологиями построения приложений, грамотно модифицировать имеющие средства (модули программ, структуры данных) для решения прикладных задач
Виды занятий	– Лекции. – Курсовая работа. – Самостоятельная работа студентов.	– Лабораторные работы. – Практические занятия. – Курсовая работа. – Самостоятельная работа студентов.	– Лабораторные работы. – Практические занятия. – Курсовая работа. – Самостоятельная работа студентов.
Используемые средства оценивания	– Опрос; – Контрольная работа; – Зачет. – Экзамен.	– Устная защита лабораторных работ. – Модификация программы по заданию преподавателя и под его непосредственным наблюдением.	– Устная защита лабораторных работ. – Модификация программы по заданию преподавателя и под его непосредственным наблюдением.

Общие характеристики показателей и критериев оценивания компетенции на всех этапах приведены в таблице 3.

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

Таблица 6 – Показатели и критерии оценивания компетенции на этапах

Показатели и критерии	Знать	Уметь	Владеть
ОТЛИЧНО (высокий уровень)	– На высоком уровне знает принципы проектирования, объектной декомпозиции.	– На высоком уровне умеет грамотно разбивать задачу на параллельные процессы; – Умеет эффективно использовать виртуальные методы, абстрактные классы.	– На высоком уровне владеет UML для объектно-ориентирования анализа и проектирования.
ХОРОШО (базовый уровень)	– Знает хорошо типовые приемы используемые при создании классов и иерархии классов; – Хорошо знает типовые шаблоны проектирования.	– Умеет хорошо строить иерархию классов; – Умеет эффективно использовать модификаторы доступа.	– Способен хорошо модифицировать готовые классы для эффективного решения задач; – Способен использовать свои и готовые классы для решения типовых задач.
УДОВЛЕТВО-РИТЕЛЬНО (низкий уровень)	– Знает общие представления об объектно-ориентированном программировании;	– Умеет создавать простые структуры, классы; – Умеет создавать	– Способен использовать готовые классы для решения простых задач.

Согласована на портале № 15935

	– Понимает важность роли объектно-ориентированного программирования.	объекты своих классов и готовых классов.	

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ

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

3.1 Темы лабораторных работ

3-й семестр

1) Классы и объекты.
2) Написание класса по правила предметной области.
3) Дружественные функции.
4) Наследование.
5) Виртуальные методы.
6) Параллельные потоки.
7) Абстрактные классы. Интерфейсы.

3.2 Темы практических заданий

4-й семестр

1) Создание пользовательского интерфейса.
2) Компоненты интерфейса, их свойства и события.
3) Контейнеры, примеры наследования.
4) Компонент таймер, класс-поток.
5) Программное формирование интерфейса, примеры наследования от компонентов.
6) Сообщения, создание собственных событий и их обработка.

3.3 Примеры вариантов лабораторных работ

Лабораторная работа №1. Вариант 1.

Цель: научиться создавать классы и объекты. Знакомство с модификаторами доступа public и private.

Общее задание. Создать класс, имеющий заданные поля, обеспечить доступ к полям через методы (запись и чтение значений). Написать программу, демонстрирующую значения полей объектов (не менее двух экземпляров созданного класса).

Класс	Поля
Монстр	Здоровье, сила

Лабораторная работа №2. Вариант 1.

Цель: научиться создавать классы с учетом правил предметной области. Уяснить необходимость скрытия некоторых полей класса модификаторами private и protected.

Общее задание

Создать класс, имеющий заданные свойства, обеспечить доступ к свойствам (запись и чтение значений свойств) с учетом заданных ограничений. Все конструкторы класса так же должны обеспечивать создание объекта с учетом заданных ограничений. Написать программу, демонстрирующую сохранение ограничений при любых действиях внешней программы над объектом.

Класс	Поля	Правила ПО
Монстр	Здоровье, сила	Каждое поле должно лежать в диапазоне от 1 до 100. Суммарное значение здоровья и силы должно быть не более 150. Если задаваемое значение не будет соответствовать условию, то автоматически присвоить ближайшее разрешенное.

Лабораторная работа №3. Вариант 1.

Цель: познакомиться со способами применения дружественных функций.

Общее задание

Создать указанный класс. Создать указанные дружественные функции для обработки объектов классов.

№	Класс	Свойства	Дружественные функции
1	Монстр	Здоровье, сила	Функция сравнения двух монстров. Возвращает истину, если первый монстр «больше» второго. Считается, что больше тот монстр, у которого больше здоровья. При равном количестве здоровья, больше

Согласована на портале № 15935

			тот монстр, у которого больше силы.

Лабораторная работа №4. Вариант 1.

Цель: познакомится с понятием наследования, научиться создавать иерархию классов, научиться создавать объекты в динамической памяти.

Общее задание

Создать базовый класс, имеющий заданные поля, обеспечить доступ к полям с помощью методов (запись и чтение значений свойств) с учетом заданных ограничений. При помощи наследования, на основе базового класса создать производный класс с учетом своих ограничений. Все конструкторы каждого класса так же должны обеспечивать создание объекта с учетом заданных ограничений. Написать программу, демонстрирующую сохранение ограничений при любых действиях внешней программы над объектом. Программа должна создавать не менее двух объектов каждого класса, хотя бы один объект должен быть создан в динамической памяти (при помощи оператора new).

Класс базовый/ дочерний	**Поля/ дополнительные поля**	**Правила ПО**
Монстр	Здоровье, сила	Суммарное значение здоровья и силы не должно превышать 150, каждое поле должно лежать в диапазоне от 1 до 100. Если задаваемое значение не будет соответствовать условию, то автоматически присвоить ближайшее разрешенное.
Демон	Разум	Разум лежит в диапазоне от 1 до 100. Суммарное значение всех полей не должно превышать 200.

Лабораторная работа №5. Вариант 1.

Цель: уяснить пользу от виртуальных методов.

Общее задание

Модифицировать программу по предыдущей лабораторной работе. Для каждого класса создать метод вывода на экран всех свойств класса, названия методов должны быть одинаковыми, например show(). В основной программе (функции main) объявить массив указателей на базовый (родительского) класс. По некоторым указателям массива создать объекты базового (родительского) класса, по остальным указателям массива создать объекты производного (дочернего) класса. С помощью цикла, для каждого указателя массива, вызвать метод show(). Запустить программу и определить не корректность вывода некоторых объектов. Задать метод show() у родительского класса как virtual. Запустить программу и определить отличие работы программы без слова virtual у метода show().

Лабораторная работа №6. Вариант 1.

Цель: получить навыки создания параллельных потоков и распределение задачи обработки данных на параллельные потоки.

Общее задание. Написать программу для обработки матрицы двумя способами: обычным (последовательным), параллельным (создать написать подпрограмму потока, для обработки строки матрицы) и обработать матрицу с помощью массива потоков, передав на обработку каждому экземпляру потока (элементу массива) свою строку или несколько строк матрицы. Программа должна замерять время обработки матрицы последовательным и параллельным способом. Запустить программу при разном количестве столбцов матрицы. Построить графики зависимости времени обработки матрицы от количества столбцов (сложности задачи). Выявить сложность задачи, при которой параллельный способ обрабатывает матрицу быстрее последовательного.

Дана матрица целых чисел размером *m* x *n*. В каждой строке матрицы определить количество нулевых элементов.

Лабораторная работа №7. Вариант 1.

Цель: получить навыки создания иерархии классов с использованием интерфейсов; уяснить пользу от их использования.

Общее задание. Создать указанный абстрактный класс. Создать несколько производных классов, реализующие поведение абстрактного класса. Продемонстрировать возможность работы с объектами разных классов через методы интерфейсного класса.

Интерфейс	**Дочерние/производные классы (драйверы)**
Стек	1) Стек на основе статического массива 2) Стек на основе динамического массива

3.4 Вопросы для подготовки к теоретическому зачету (для студентов, которые не выполнили все задания семестра)

1) Понятия класс и объект. Отображение класса на UML диаграмме. Пример объявления класса.
2) Понятия класс и объект. Отображение класса на UML диаграмме. Способы создания объектов.
3) Инкапсуляция, уровни доступа к членам класса. Причины закрытия доступа к членам класса.
4) Конструкторы и деструкторы. Типы конструкторов. Объявление объектов с использованием различных типов конструкторов.

Согласована на портале № 15935

5) Конструкторы и деструкторы. Типы конструкторов. Объявление массивов объектов.
6) Понятия поле и метод. Отличие понятия поля класса от переменной программы.
7) Понятия поле и метод. Отличие понятия метода класса от функции программы.
8) Статические члены класса.
9) Наследование. Отображение обобщения или генерализации в UML.
10) Дружественные функции.
11) Использование объектов как полей классов. Агрегация и композиция в UML.
12) Передача объектов через параметры методов. Отношение зависимости между классами в UML.
13) Виртуальные методы.
14) Абстрактные классы.
15) Шаблоны функций и классов. Параметризованные классы в UML.
16) Потоки. Диаграмма деятельности в UML. Отображение параллельных потоков на диаграмме деятельности.
17) Обработка исключений.

3.5 Вопросы для подготовки к экзамену

1) Понятия класс и объект. Отображение класса на UML диаграмме. Пример объявления класса в C++ или C#.
2) Понятия класс и объект. Отображение класса на UML диаграмме. Способы создания объектов в C++ или C#.
3) Инкапсуляция, уровни доступа к членам класса. Причины закрытия доступа к членам класса.
4) Конструкторы и деструкторы. Типы конструкторов. Объявление объектов с использованием различных типов конструкторов в C++ или C#.
5) Конструкторы и деструкторы. Типы конструкторов. Объявление массивов объектов в C++ или C#.
6) Понятия поле и метод. Отличие понятия поля класса от переменной программы.
7) Понятия поле и метод. Отличие понятия метода класса от функции программы.
8) Статические члены класса.
9) Наследование. Отображение обобщения или генерализации в UML.
10) Дружественные функции.
11) Использование объектов как полей классов. Агрегация и композиция в UML.
12) Передача объектов через параметры методов. Отношение зависимости между классами в UML.
13) Виртуальные методы.
14) Абстрактные классы.
15) Шаблоны функций и классов. Параметризованные классы в UML.
16) Потоки. Диаграмма деятельности в UML. Отображение параллельных потоков на диаграмме деятельности.
17) Обработка исключений.
18) Компоненты интерфейса, их свойства и события в C# (WindowsForms)
19) Контейнеры в C#, примеры наследования.
20) Компонент таймер, класс поток в C#.
21) Программное формирование интерфейса, примеры наследования от компонентов в C#.

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Для обеспечения процесса обучения и решения задач обучения используются следующие материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, в составе:

1. Учебник для вузов «Объектно-ориентированное программирование» приведено в рабочей программе в разделе 12.1 [1].
2. Методические указания по курсовой, самостоятельной и индивидуальной работе студентов всех форм обучения приведены в рабочей программе в разделе 12.3 [1].
3. Методические указания к лабораторным работам в разделе 12.3 [2].

Согласована на портале № 15935