

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМ. И. РАЗАКОВА


ВЫСШАЯ ШКОЛА МАГИСТРАТУРЫ

Кафедра «Прикладная математика и информатика»

«УТВЕРЖДАЮ»

Директор ВШМ, к. т. н.

\_\_\_\_\_ (факультет/институт)  
Омуров Ж.М.  
(подпись)  
« 16 » 09 2022 г.



УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

ПО ДИСЦИПЛИНЕ

М.1.В.1 «Объектно-ориентированные языки и системы  
программирования»

<b><u>Направление:</u></b>	510200 Прикладная математика и информатика
<b><u>Профиль:</u></b>	Математическое моделирование
<b><u>Квалификация:</u></b>	Магистратура
<b><u>Форма обучения:</u></b>	Очная

Бишкек 2022

## Лист согласования

Учебно-методический комплекс по дисциплине «Объектно-ориентированные языки и системы программирования» разработан в соответствии с требованиями ГОС ВПО по подготовке магистров и предназначен для магистрантов, обучающихся по направлению 510200 Прикладная математика и информатика профилю «Математическое моделирование».

Составитель: д.ф. –м.н., профессор Кабаева Г.Дж.

Процесс рассмотрения и утверждения УМКД	№ протокола	Подписи (печать)
Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры <u>ПМИ</u> <u>ИИТ</u> (наименование учебного подразделения)	протокол № <u>1</u> от « <u>16</u> » <u>сентября</u> 2022 г.	Зав. профилирующей кафедры: <u>Джаманбаев М.Дж.</u> (подпись)
* Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры _____ (наименование учебного подразделения)	протокол № _____ от « _____ » _____ 2022 г.	Зав. непрофилирующей кафедры:  <u>Джаманбаев М.Дж.</u> (подпись)
Учебно-методический комплекс дисциплины одобрен руководителем ООП по направлению _____ (наименование учебного подразделения)	Дата: <u>16.09.2022</u>	Руководитель ООП:  <u>Джаманбаев М.Дж.</u> (подпись)
Учебно-методический комплекс дисциплины согласован на заседании Учебно-методической комиссии факультета/института <u>ИИТ</u> (наименование учебного подразделения)	протокол № <u>1</u> от « <u>19</u> » <u>октября</u> 2022 г.,	Председатель УМК:  <u>Зорнда Е.И.</u> (подпись)
** Учебно-методический комплекс дисциплины согласован (или обсуждался/рецензирован)  (указать наименование предприятия/ учреждения/ организации)	Дата: согласования/ обсуждения/ рецензия	(должность) _____  (подпись) <u>Ф.И.О.</u> _____

\* УМК дисциплины непрофилирующей кафедры обязательно согласовывается с выпускающей кафедрой, реализующей соответствующее направление/специальность

\*\* УМК должен пройти согласование или обсуждение на соответствие требованиям заинтересованных сторон (отраслевой совет, «круглый стол», совещание, заседание кафедры/методический совет с представителями производства, рецензирование (рецензия должна быть приложена) и др.)

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ  
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМ. И. РАЗЗАКОВА

ВЫСШАЯ ШКОЛА МАГИСТРАТУРЫ

Кафедра «Прикладная математика и информатика»

«УТВЕРЖДАЮ»

Директор ВШМ, к. т.н.



(Факультет/институт)

Омуров Ж.М

(подпись)

« 16 » 09 2022 г.

РАБОЧАЯ ПРОГРАММА  
ПО ДИСЦИПЛИНЕ  
М.1.В.1 «Объектно-ориентированные языки и системы  
программирования»

<b><u>Направление:</u></b>	510200 Прикладная математика и информатика	
<b><u>Профиль:</u></b>	Математическое моделирование	
<b><u>Квалификация:</u></b>	Магистратура	
<b><u>Форма обучения:</u></b>	очная	
<b><u>Семестр</u></b>	1	
<b><u>Всего кредитов</u></b>	5 кредит	150 часов
<b><u>Лекции</u></b>	1 кредит	32 часа
<b><u>Лабораторных</u></b>	0,5 кредит	16 часов
<b><u>СРС</u></b>	3,5 кредит	102 часа

Бишкек 2022

## Лист согласования

Рабочая программа по дисциплине «Объектно-ориентированные языки и системы программирования» разработан в соответствии с требованиями ГОС ВПО по подготовке магистров и предназначен для магистрантов, обучающихся по направлению 510200 Прикладная математика и информатика профилю «Математическое моделирование».

Составитель: д.ф.-м.н., профессор Кабаева Г.Д.

Процесс рассмотрения и утверждения РПД	№ протокола	Подписи (печать)
<p>Рабочая программа дисциплины рассмотрена на заседании кафедры <u>ИМ и И</u></p> <p>(наименование учебного подразделения)</p>	<p>протокол № <u>1</u> от « <u>16</u> » <u>сентяб.</u> 202<u>2</u> г.</p>	<p>Зав. профилирующей кафедры:</p> <p><u>(подпись)</u></p> <p>Ф.И.О. <u>Тамара Кабаева Г.Д.</u></p>
<p>*Рабочая программа дисциплины рассмотрена/согласована на заседании кафедры _____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____ от « _____ » 202__ г.</p>	<p>Зав. не/профилирующей кафедры:</p> <p>Ф.И.О. _____</p> <p><u>(подпись)</u></p>
<p>Рабочая программа дисциплины одобрена руководителем ООП по направлению _____</p> <p>(наименование учебного подразделения)</p>	<p>Дата: <u>16.09.2022</u></p>	<p>Руководитель ООП:</p> <p><u>(подпись)</u></p> <p>Ф.И.О. <u>Тамара Кабаева Г.Д.</u></p>
<p>Рабочая программа дисциплины согласована на заседании Учебно-методической комиссии факультета/института _____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № <u>1</u> от « <u>19</u> » <u>сентяб.</u> 202<u>2</u> г.,</p>	<p>Председатель УМК:</p> <p><u>(подпись)</u></p> <p>Ф.И.О. <u>Золотова Е.И.</u></p>
<p>**Рабочая программа дисциплины согласована (или обсуждалась/рецензирована) _____</p> <p>(указать наименование предприятия/учреждения/организации)</p>	<p>Дата: согласования/обсуждения/рецензия</p>	<p>(должность) _____</p> <p><u>(подпись)</u></p> <p>Ф.И.О. _____</p>

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМ. И. РАЗЗАКОВА

**ВЫСШАЯ ШКОЛА МАГИСТРАТУРЫ**

**Кафедра «Прикладная математика и информатика»**

**«УТВЕРЖДАЮ»**

Директор ВШМ, к. т.н.

\_\_\_\_\_ (факультет/институт)

Омуров Ж.М

\_\_\_\_\_ (подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 202\_\_ г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС**

**ПО ДИСЦИПЛИНЕ**

**М.1.В.1 «Объектно-ориентированные языки и системы  
программирования»**

<b><u>Направление:</u></b>	510200 Прикладная математика и информатика
<b><u>Профиль:</u></b>	Математическое моделирование
<b><u>Квалификация:</u></b>	Магистратура
<b><u>Форма обучения:</u></b>	Очная

Бишкек 2022

## Лист согласования

Учебно-методический комплекс по дисциплине «Объектно-ориентированные языки и системы программирования» разработан в соответствии с требованиями ГОС ВПО по подготовке магистров и предназначен для магистрантов, обучающихся по направлению 510200 Прикладная математика и информатика профилю «Математическое моделирование».

Составитель: д.ф. –м.н., профессор Кабаева Г.Дж.

Процесс рассмотрения и утверждения УМКД	№ протокола	Подписи (печать)
Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры _____ ПМИ _____ ИИТ (наименование учебного подразделения)	протокол № _____ от « _____ » _____ 2022 г.	Зав. профилирующей кафедры: _____ (подпись) Джаманбаев М.Дж.
*Учебно-методический комплекс дисциплины рассмотрен на заседании кафедры _____ (наименование учебного подразделения)	протокол № _____ от « _____ » _____ 2022 г.	Зав. непрофилирующей кафедры: _____
Учебно-методический комплекс дисциплины одобрен руководителем ООП по направлению _____ (наименование учебного подразделения)	Дата:	Руководитель ООП: _____ (подпись) Ф.И.О. _____
Учебно-методический комплекс дисциплины согласован на заседании Учебно-методической комиссии факультета/института _____ ИИТ _____ (наименование учебного подразделения)	протокол № _____ от « _____ » _____ 2022 г.,	Председатель УМК: _____ (подпись) Ф.И.О. _____
**Учебно-методический комплекс дисциплины согласован (или обсуждался/рецензирован) _____ (указать наименование предприятия/ учреждения/организации)	Дата: согласования/ обсуждения/ рецензия	(должность) _____ _____ (подпись) Ф.И.О. _____

\*УМК дисциплины непрофилирующей кафедры обязательно согласовывается с выпускающей кафедрой, реализующей соответствующее направление/специальность

\*\*УМК должен пройти согласование или обсуждение на соответствие требованиям заинтересованных сторон (отраслевой совет, «круглый стол», совещание, заседание кафедры/методический совет с представителями производства, рецензирование (рецензия должна быть приложена) и др.)



## Содержание УМКД

Разделы		стр
1	Рабочая программа	5-7
2	Содержание дисциплины по Госстандарту. Пояснительная записка	8-9
3	Состав разделов Рабочей программы (цели и задачи изучения дисциплины, ее значение в учебном процессе)	10-11
4	Межпредметные связи. Перечень дисциплин и разделов, усвоение которых необходимо при изучении данной дисциплины	12
5	Структура дисциплины с разбивкой по видам занятий, часам и модулям.	13
6	Распределение баллов по модулям и видам учебных занятий	14
7	Перечень контрольных вопросов теоретического курса	15-18
8	Литература	19
9	Силлабус	20-23
10	Глоссарий	24-28



**ВЫСШАЯ ШКОЛА МАГИСТРАТУРЫ**

**Кафедра «Прикладная математика и информатика»**

**«УТВЕРЖДАЮ»**

Директор ВШМ, к. т.н.

\_\_\_\_\_ (факультет/институт)

Омуров Ж.М

\_\_\_\_\_ (подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 202\_\_ г.

**РАБОЧАЯ ПРОГРАММА  
ПО ДИСЦИПЛИНЕ**

**М.1.В.1 «Объектно-ориентированные языки и системы  
программирования»**

<b><u>Направление:</u></b>	510200 Прикладная математика и информатика	
<b><u>Профиль:</u></b>	Математическое моделирование	
<b><u>Квалификация:</u></b>	Магистратура	
<b><u>Форма обучения:</u></b>	очная	
<b><u>Семестр</u></b>	1	
<b><u>Всего кредитов</u></b>	5 кредит	150 часов
<b><u>Лекции</u></b>	1 кредит	32 часа
<b><u>Лабораторных</u></b>	0,5 кредит	16 часов
<b><u>СРС</u></b>	3,5 кредит	102 часа

## Лист согласования

Рабочая программа по дисциплине «Объектно-ориентированные языки и системы программирования» разработан в соответствии с требованиями ГОС ВПО по подготовке магистров и предназначен для магистрантов, обучающихся по направлению 510200 Прикладная математика и информатика профилю «Математическое моделирование».

Составитель: д.ф.-м.н., профессор Кабаева Г.Д.

Процесс рассмотрения и утверждения РПД	№ протокола	Подписи (печать)
<p>Рабочая программа дисциплины рассмотрена на заседании кафедры _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____ от «_____» _____ 202__ г.</p>	<p>Зав. профилирующей кафедры:</p> <p>_____</p> <p style="text-align: center;">(подпись)</p> <p><u>Ф.И.О.</u> _____</p>
<p>* Рабочая программа дисциплины рассмотрена/согласована на заседании кафедры _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____ от «_____» _____ 202__ г.</p>	<p>Зав. не/профилирующей кафедры:</p> <p>_____</p> <p><u>Ф.И.О.</u> _____</p>
<p>Рабочая программа дисциплины одобрена руководителем ООП по направлению _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>Дата:</p>	<p>Руководитель ООП:</p> <p>_____</p> <p style="text-align: center;">(подпись)</p> <p><u>Ф.И.О.</u> _____</p>
<p>Рабочая программа дисциплины согласована на заседании Учебно-методической комиссии факультета/института _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____ от «_____» _____ 202__ г.,</p>	<p>Председатель УМК:</p> <p>_____</p> <p style="text-align: center;">(подпись)</p> <p><u>Ф.И.О.</u> _____</p>
<p>** Рабочая программа дисциплины согласована (или обсуждалась/рецензирована) _____</p> <p>_____</p> <p>(указать наименование предприятия/учреждения/организации)</p>	<p>Дата: согласования/обсуждения/рецензия</p>	<p>(должность) _____</p> <p>_____</p> <p style="text-align: center;">(подпись)</p> <p><u>Ф.И.О.</u> _____</p>



### Пояснительная записка

Дисциплина «Объектно-ориентированное программирование» является одной из специальных дисциплин профессионального цикла магистерской программы.

### Цель курса

Учебная дисциплина «Объектно-ориентированное программирование» читается с целью усвоения магистрантами базовых знаний об основных понятиях и принципах объектно-ориентированного программирования (ООП), формирования у них систематизированных теоретических знаний и практических навыков создания объектно-ориентированного программного обеспечения с применением современных технологий.

Освоение курса должно способствовать развитию у магистрантов умения и навыков проектирования и разработки приложений с применением объектно-ориентированного подхода и технологий, профессионального применения методологии и средств технологии объектно-ориентированного программирования.

### Задачи дисциплины

Дать базовые знания об основных понятиях и принципах объектно-ориентированного программирования; формирование представлений об общей методологии и средствах технологии объектно-ориентированного программирования; научить применять принципы объектно-ориентированной разработки для написания программ на языке высокого уровня; ознакомить с методологией проектирования и разработки приложений с применением объектно-ориентированного подхода; практическое освоение технологии объектно-ориентированного программирования.

В результате освоения дисциплины «**Объектно-ориентированное программирование и системы программирования**» магистрант должен:

**знать:** основные понятия, принципы и модели ООП; пользовательские типы данных – классы, объект, объявление и реализацию класса, данные, методы; доступ, наследование свойств, виртуальные функции; алгоритмы и методы проектирования объектно-ориентированных программ; средства и подходы конструирования собственных пользовательских классов; объектно-ориентированные языки, язык C++/C#; правила организации, методы и средства программирования интерфейса; объектно-ориентированные системы;

**уметь:** применять при разработке, анализе и проверке программ принципы объектно-ориентированного программирования – такие, как инкапсуляция, полиморфизм, наследование; определять классы, создавать объекты, разрабатывать и использовать конструкторы класса; создавать программы, используя библиотеки классов; выбрать и использовать алгоритмы и методы проектирования объектно-ориентированных программ; использовать правила организации и программирования интерфейса; использовать языки, методы и средства программирования;

**владеть:** навыками объектно-ориентированного программирования на C++; основными методами и приемами, и средствами разработки объектно-ориентированных программ.

## Компетенции

Магистрант, изучивший курс «**Объектно-ориентированное программирование и системы программирования**» должен решать следующие профессиональные задачи в соответствии с видами профессиональной деятельности:

*-производственно-технологическая деятельность:*

уметь применять современные образовательные и информационные технологии, способность решать задачи производственной и технологической деятельности на профессиональном уровне с использованием объектно-ориентированной технологии в области прикладного программирования;

*-организационно -управленческая деятельность:*

применять базовые и специальные знания и объектно-ориентированный подход в профессиональной деятельности для решения прикладных задач различной степени сложности.

Магистрант, в результате освоения дисциплины «**Объектно-ориентированное программирование и системы программирования**» должен обладать следующими компетенциями:

### **а) универсальными:**

*- общенаучными (ОК):*

- способность к обобщению и анализу проблем, постановке целей и выбору путей их достижения;

- умение логически последовательно, аргументировано и ясно излагать мысли;

- стремление к саморазвитию, повышению своей квалификации и мастерства.

*- инструментальными (ИК):*

- уметь использовать средства разработки программного обеспечения и вычислительные математические пакеты;

*- социально-личностными и общекультурными (СЛК):*

- готовность приобретать с большей степенью самостоятельности новые знания, используя современные образовательные технологии;

- способность собирать и интерпретировать необходимые данные для формирования суждений по научным и этическим проблемам.

### **б) профессиональными (ПК):**

*- в производственно-технологической деятельности (ПТД):*

- иметь представление о современном состоянии и средствах разработки программного обеспечения;

- способность углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности.

*- в научно-исследовательской деятельности:*

- способностью разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач;

- способность проводить научные исследования и получать новые научные и прикладные результаты;

- способность порождать новые идеи и демонстрировать навыки самостоятельной научно-исследовательской работы и работы в научном коллективе.

**Пререквизиты:** «Информатика», «Математическая логика и теория алгоритмов», «Языки и методы программирования», «Операционные системы», «Алгоритмы. Построение и анализ».

**Постреквизиты:** все дисциплины профессионального цикла, связанные с проектированием программного обеспечения.

**Сфера применения результатов изучения дисциплины.**

Знания, полученные в ходе изучения дисциплины должны помочь магистрантам в их дальнейшей профессиональной деятельности. Магистр найдет применение своих знаний в образовательной, научно-исследовательской, проектно-производственной и организационно-управленческой деятельности.

### Структура дисциплины «Объектно-ориентированное программирование и системы программирования»

№	Темы лекций и содержание практических занятий	лк	лб	Всего
<b>1 модуль</b>				
1	Парадигмы программирования	4	4	8
2	Основные понятия и принципы ООП.	4	4	8
3	Динамические структуры данных.	5	4	9
4	Массивы объектов.	3	4	7
<b>Итого часов по 1 модулю</b>		<b>16</b>	<b>16</b>	<b>32</b>
<b>2 модуль</b>				
5	Дружественные функции и перегрузка операций.	4	2	6
6	Перегруженные функции-члены, перегрузка операторов.	2	2	4
7	Иерархия и наследование.	4	4	8
8	Классы потоков C++.	2	4	6
9	Пространства имен.	4	4	8
<b>Итого часов по 2 модулю</b>		<b>16</b>	<b>16</b>	<b>32</b>
<b>Итого часов за 1 семестр</b>		<b>32</b>	<b>32</b>	<b>64</b>

### Содержание учебной дисциплины Тематика лекционных занятий - 1 семестр

№	Темы лекций	Кол-во часов	Кол-во баллов
<b>Модуль I</b>			
1	<b>Парадигмы программирования:</b> процедурное, структурированное и объектно-ориентированное программирование, парадигмы. Объектно-ориентированные языки. Системы объектно-ориентированного программирования. Базовые средства языка C++: компоненты языка, типы данных; переменные и выражения; управляющие конструкции – операторы ветвления, операторы цикла, операторы передачи управления; Указатели и массивы.	4	2
2	<b>Основные понятия и принципы ООП.</b> Принципы объектно-ориентированного программирования: инкапсуляция, наследование и полиморфизм. Основные понятия ООП: классы и члены, определение объекта. Абстрактные типы данных и классы: объявление и реализация класса, объявление объектов, создание объектов. Объекты и передача информации. Доступ к членам класса. Ограничение доступа к членам класса. Определение методов класса. Объявление функций - членов. Конструкторы и деструкторы. Конструкторы и деструкторы, заданные по умолчанию. Классы, содержащие другие классы в качестве данных-членов.	4	2
3	<b>Динамические структуры данных.</b> Указатели. Указатель this. Статические члены: функции и данные. Указатели на члены. Структуры и объединения. Константные члены-функции и константные объекты. Динамически создаваемые объекты. Структуры. Размещение объектов в динамически распределяемой памяти, доступ к членам класса. Динамиче-	5	3

	ское размещение членов класса. Использование указателей, как средство хранения адреса. Блуждающие, дикие или зависшие указатели. Освобождение данных объекта: деструкторы. Ссылки. Передача аргументов функции как ссылок. Возвращение значений с помощью ссылок. Возвращение ссылки на объект в области динамического обмена.		
4	<b>Массивы объектов.</b> Объявление массивов в области динамического обмена. Удаление массива из области динамического обмена. Классы массивов. Линейные списки. Стеки. Очереди. Бинарные деревья.	3	1
<b>Итого за 1 модуль:</b>		16	8

<b>Модуль 2</b>			
5	<b>Дружественные функции и перегрузка операций.</b> Функции-друзья. Перегрузка бинарных и унарных операций. Перегруженные операции индексирования, вызова функций, инкремента и декремента префиксных и постфиксных, раз адресации. Перегрузка new, delete. Преобразование типов, определяемых пользователем с помощью конструкторов и операций преобразования. Неявное преобразование типов. Друзья-функции и друзья-классы.	4	2
6	<b>Перегруженные функции-члены, перегрузка операторов.</b> Перегруженные функции-члены. Перегрузка конструкторов. конструктор-копировщик. Перегрузка операторов. Основные принципы перегрузки операторов.	2	1
7	<b>Иерархия и наследование.</b> Наследование, иерархия и наследование. Типы наследования. Замещение метода базового класса в производном классе. Виртуальные методы, виртуальные функции, виртуальные деструкторы, виртуальный конструктор копировщик. Полиморфизм, одиночное наследование, множественное наследование. Конструкторы классов, полученные в результате множественного наследования. Виртуальное наследование. Указатели на функции, указатели на функции-члены, массивы указателей на функции-члены. Дополнительные возможности наследования: доступ к членам вложенного класса, передача объекта как значения, закрытое наследование, классы друга, функции друга. Различные пути передачи функциональности классу.	4	2
8	<b>Классы потоков C++.</b> Заголовочные файлы. Предопределенные объекты и потоки. Операции помещения и извлечения. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоков C++. Конструкторы файловых потоков. Открытие файлов в разных режимах. Ввод-вывод в файлы. Форматирование в памяти.	2	1
9	<b>Пространства имен.</b> Пространства имен, создание и использование, стандартное пространство имен. Анализ и проектирование объектно-ориентированных программ. Построение моделей. Проектирование программ: язык моделирования. Динамическая модель. Шаблоны, параметризованные типы, объявление шаблона, функции шаблона, шаблоны и друзья, стандартная библиотека шаблонов. Контейнеры. Последовательные контейнеры. Ассоциативные контейнеры. Классы ал-	4	2

	горитмов. Методы и средства программирования. Препроцессор и компилятор. Компилятор Visual C++. Макросы. Стиль программирования.		
	<b>Итого за 2 модуль:</b>	16	8
	<b>Всего:</b>	32	16

### Тематика лабораторных занятий

<b>Модуль 1</b>			
№	Темы занятий	Кол-во часов	Кол-во баллов
1	Основы программирования на C++ в среде Visual Studio 2010/2013. Изучение инструментов и основных возможностей среды Visual C++ 2010. Настройки среды. <i>Создание консольного проекта</i> . Структура проекта. Подключение заголовочных файлов. Базовые средства языка C++: компоненты языка, типы данных; переменные и выражения; управляющие конструкции – операторы ветвления, операторы цикла	2	2
2	Особенности языка и объектно-ориентированное программирование на языке C++. приемы программирования. <i>Создание консольного приложения</i> . Объявление переменных: локальные и глобальные переменные. Запись арифметических выражений. Построение и запуск. Отладочные механизмы.	2	2
3	<i>Создание консольного приложения</i> . Типы данных в C++: значимые и ссылочные типы. Классы. Основные понятия ООП: классы и члены, определение объекта. Абстрактные типы данных и классы: объявление и реализация класса, объявление объектов, создание объектов. Объекты и передача информации. Доступ к членам класса. Ограничение доступа к членам класса. Определение методов класса. Объявление функций - членов. Реализация арифметических вычислений. Объявление массивов. Операции с массивами.	4	2
4	<i>Создание консольного приложения</i> . Классы, объекты, методы, свойства и переменные экземпляров. Модификаторы доступа <i>public, private</i> . Системы программирования. Интегрированная среда разработки (IDE) Visual Studio 2010/2013. Методология разработки приложений в IDE . Основные команды управления разработкой, сопровождением и выполнением приложений. Разработка объектно-ориентированного приложения: отделение реализации от интерфейса. Разбивка на ряд файлов: определение класса в заголовочном файле, определение функций-элементов этого класса в файле исходных кодов с теми же базовыми именами.	4	2
5	<i>Создание консольного приложения</i> . Решение задачи поиска в массиве структур. Разработка простой базы данных сотрудников. Разработка класса <i>Сотрудник</i> . Интерфейс класса. Использование конструкторов и деструкторов. Разновидности конструкторов. Конструкторы и деструкторы, заданные по умолчанию, конструктор умолчания, конструктор преобразования, конструктор копирования, конструктор с двумя и более параметрами Классы, содержащие другие классы в качестве данных-членов.	4	3
	<b>Итого за 1 модуль:</b>	<b>16</b>	<b>14</b>



<b>Модуль 2</b>				
№	Темы занятий	Кол-во часов	Кол-во баллов	
6	<i>Создание консольного приложения.</i> Объявление и создание динамических структур данных. Указатели. Указатель this. Статические члены: функции и данные. Указатели на члены. Разработка программы с использованием указателей для передачи данных. на языке C++. Работа с динамическим массивом чисел. Сортировка массивов. Алгоритмы сортировки.	2	2	
7	<i>Создание консольного приложения.</i> Структуры и объединения. Константные члены-функции и константные объекты. Динамически создаваемые объекты. Структуры. Размещение объектов в динамически распределяемой памяти, доступ к членам класса. Динамическое размещение членов класса. Использование указателей, как средство хранения адреса. Блуждающие, дикие или зависшие указатели. Освобождение данных объекта: деструкторы. Ссылки. Передача аргументов функции как ссылок. Возвращение значений с помощью ссылок.	4	2	
8	<i>Реализация класса треугольников.</i> Использование перегрузки операторов. Перегруженные функции-члены. Перегрузка конструкторов. Применение конструктора копирования. Тестирование и отладка программы. Исправление ошибок.	2	3	
9	<i>Разработка приложения с реализацией наследования классов.</i> Взаимоотношение – отношение наследования (отношение родитель/потомок, отношение обобщение/специализация) на примере классов: <i>треугольник, четырехугольник, многоугольник.</i> Замещение функций базового класса. Конструкторы и деструкторы в базовом классе. Устранение неоднозначности при множественном наследовании. Доступ к объектам иерархии. Виртуальные методы. Разработка приложения <i>Калькулятор</i> с классом <i>Меню</i> . Составление диаграммы классов для разрабатываемой программы.	4	3	
10	Шаблоны классов. Определение шаблона класса. использование шаблона класса. Организация исходного кода. параметры шаблонов Разработка шаблонного класса для разреженных массивов. Обработка исключительных ситуаций: определение, перехват. Классы исключений.	2	2	
11	Разработка приложения с реализацией файлов и потоков в C++. Создание текстового файла с последовательным доступом. Коллекции. Стеки. Очереди	2	2	
<b>Итого за 2 модуль:</b>		<b>16</b>	<b>14</b>	
<b>Всего:</b>		<b>32</b>	<b>28</b>	
<b>Самостоятельная работа студентов</b>				
№	Темы занятий	Кол-во часов	Кол-во баллов	Литература
1	Разработать программу вычисления значений функций одной переменной на заданном отрезке, с заданным шагом.	6	1	[1], [7]
2	Создание динамического массива заданного размера, выделение из массива указанных в варианте диапазонов.	6	1	[1], [7]
3	Доработать программу калькулятор из лабораторной работы 8. Добавить функции.	6	1	[1], [7]
4	Разработать на C++ программу реализации задачи ин-	12	2	[1],[3], [5], [6],

	терполирования при заданных табличных данных. Программа должна предлагать выбор формулы интерполирования. Ввод данных осуществляется с клавиатуры.			[7]
5	Реализация численных методов решения систем линейных уравнений на C++ на основе объектно-ориентированного подхода, на основе класса массивов.	6	2	[3], [5], [6]
6	Разработка на C++ приложения для статистической обработки массива случайных данных. Генератор случайных чисел. Реализация метода наименьших квадратов для решения уравнения линейной регрессии.	4	1	[1],[3], [5], [6]
7	Разработать шаблонный класс для представления динамических одномерных массивов. Класс обеспечивает хранение данных любого типа, для него предусмотрены конструктор по умолчанию, конструктор копирования и операция присваивания.	8	2	[1],[3], [5], [6]
	<b>Всего</b>	<b>48</b>	<b>10</b>	

### Рейтинг - контроль

Успешность изучения дисциплины в системе кредитных технологий оценивается суммой набранных баллов (из 100 возможных) по всем видам работ, согласно следующему распределению:

Модуль 1	Баллы	Модуль 2	Баллы
Всего баллов	<b>30</b>	Всего баллов	<b>30</b>
Теоретический курс	8	Теоретический курс	8
Лабораторные занятия	14	Лабораторные занятия	14
СРС	5	СРС	5
Посещаемость	3	Посещаемость	3

**Модуль 1(30 баллов) + Модуль 2 (30 баллов)=60 баллов**

**Итоговый контроль (экзамен) 40 баллов.**

### Штрафные баллы:

1. Пропуски занятий по неуважительной причине (0,75 баллов за каждое занятие)
2. Несвоевременный отчет по лабораторным работам (0,5 баллов за каждое занятие)
3. Несвоевременное выполнение самостоятельной работы студента - 1 балл.

**Ожидаемый результат:** освоение магистрантом учебного материала по основным понятиям дисциплины «Объектно-ориентированное программирование»; освоение объектно-ориентированной технологии разработки программ; способность разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач, умение работать с компьютерными системами и в среде современных вычислительных пакетов.

### Карта рейтинг контроля.

№ модуля	Объем модуля в часах	Оценка в баллах		Сроки
		min	max	
<i>Текущий контроль</i>				
М.1	Лекции – 16 ч.	5	8	8 неделя
	Лаб. - 16 ч.	11	14	
	СРС - 28 ч.	3	5	
	Посещаемость	2	3	
	<b>Сумма баллов</b>	<b>21</b>	<b>30</b>	
М.2	Лекции – 16 ч.	5	8	16 неделя
	Лаб. - 16 ч.	11	14	
	СРС - 28 ч.	3	5	

	Посещаемость	2	3	
	<b>Сумма баллов</b>	<b>21</b>	<b>30</b>	
<i>Итоговый контроль</i>				
	Всего баллов	<b>19</b>	<b>40</b>	по расписанию экзаменов

На основании полученной студентом суммы баллов за семестр выставляется оценка, в соответствии с приведенной ниже таблицей.

#### **Система оценок в кредитной технологии обучения**

<b>Рейтинг (баллы)</b>	<b>Оценка по буквенной системе</b>	<b>Цифровой эквивалент оценки</b>	<b>Оценка по традиционной системе (4-х бальной)</b>
87 - 100	A	4,0	Отлично
80 - 86	B	3,33	Хорошо
74 - 79	C	3,0	
68 - 73	D	2,33	Удовлетворительно
61 - 67	E	2,0	
41 - 60	FX	0	Неудовлетворительно
0 - 40	F	0	

#### **Образовательные технологии**

При чтении лекций используются активные и пассивные формы подачи теоретического материала. Закрепление теоретического материала выполняется при проведении лабораторных работ с использованием современной вычислительной техники и интегрированной среды программирования C++. Для проведения лабораторных работ разработаны электронные методические указания и задания. Самостоятельные задания выполняются с использованием *Internet*-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы.

#### **Перечень вопросов теоретического модуля 1**

1. Абстрактные типы данных. ADT- формат;
2. Концепция типа для данных;
3. Базовые типы данных;
4. Спецификации, представление, реализация в языках C++ целочисленных типов;
5. Символьные типы, их спецификации, представление, реализация в языках C++;
6. Вещественные типы, их спецификации, представление, реализация в языках C+;
7. Типы перечисления, реализация в языках C++;
8. Тип указателей, их спецификации, представление, реализация в языках C++;
9. Массивы, их спецификации, представление, реализация в языках C++;
10. Строковые константы и переменные, записи, множества, файлы;
11. Классы: объявление и реализация класса.
12. Объявление объектов, создание объектов.
13. Конструктор, реализация конструктора; разновидности конструкторов, деструкторы
14. Динамическое программирование.
15. Основные принципы процедурного и структурированного программирования;
16. Основные принципы объектно-ориентированного программирования;
17. Объектно-ориентированные языки;
18. Основные понятия ООП - классы и члены, определение объекта.
19. Объекты и передача информации;
20. Примеры классов;

21. Ограничение доступа к членам класса. Как организовать доступ к закрытым переменным-членам?
22. Динамические структуры данных.
23. Динамически создаваемые объекты.
24. Указатели.
25. Ссылки. Передача аргументов функции как ссылок. Возвращение значений с помощью ссылок.
26. Размещение объектов в динамически распределяемой памяти, доступ к членам класса.
27. Объявление массивов в области динамического обмена. Классы массивов.
28. Сохранение массива в области динамической памяти. Указатель на массив.

### **Перечень вопросов теоретического модуля 2**


29. Перегруженные функции-члены.
30. Перегрузка конструкторов. Конструктор-копировщик.
31. Перегрузка операторов, Основные принципы перегрузки операторов;
32. Наследование, иерархия и наследование. Типы наследования;
33. Виртуальные методы, виртуальные функции.
34. Виртуальные деструкторы.
35. Виртуальный конструктор копировщик.
36. Полиморфизм.
37. Одиночное наследование, множественное наследование.
38. Конструкторы классов, полученные в результате множественного наследования.
39. Виртуальное наследование;
40. Дополнительные возможности наследования: доступ к членам вложенного класса, передача объекта как значения;
41. Закрытое наследование, классы друга, функции друга;
42. Пространства имен, создание и использование, стандартное пространство имен;
43. Анализ и проектирование объектно-ориентированных программ.
44. Построение моделей;
45. Проектирование программ: язык моделирования.
46. Динамическая модель.
47. Объявление шаблона, функции шаблона, стандартная библиотека шаблонов.
48. Контейнеры.
49. Классы алгоритмов.
50. Методы и средства программирования.
51. Препроцессор и компилятор.
52. Макросы.
53. Стиль программирования.
54. Классы потоков C++.
55. Заголовочные файлы. Предопределенные объекты и потоки.
56. Операции помещения и извлечения.
57. Форматирование. Флаги форматирования.
58. Манипуляторы. Ошибки потоков.
59. Файловый ввод-вывод с применением потоков C++.
60. Конструкторы файловых потоков.
61. Открытие файлов в разных режимах. Ввод-вывод в файлы.

## Образцы вариантов текущего и итогового контроля

### Образцы тестовых заданий

1. Согласно концепции типа для данных, каждая константа, переменная, выражение или функция бывают определенного типа. Что характеризует каждый тип?
  - a) Объем памяти, выделяемый для переменной в зависимости от ее значения;
  - b) Множество значений, к которому принадлежит константа, которые может принимать переменная или вырабатывать функция и некоторое множество операций. Тип данных определяет их размер в памяти;
  - c) Строгое разграничение между вещественными, комплексными и логическими переменными, между переменными, множествами, множествами множеств и т.д.
  - d) Множество значений, к которому принадлежит константа, которые может принимать переменная.
  
2. Какие типы данных относятся к стандартным простым типам и почему?
  - a) Стандартными простыми типами являются типы, встроенные для большинства компьютеров - это целые и вещественные числа, логические значения, множество символов печати, указатели, массивы и записи;
  - b) Целые и вещественные числа, символьные и логические значения, т.к. объекты этих типов не могут быть разделены на меньшие части, реализация этих типов обеспечивается описаниями типа языка программирования и компьютерными аппаратными средствами;
  - c) Это типы, которые являются встроенными для вычислительных машин. Они включают положительные и отрицательные целые числа, булевские значения и множество символов ASCII.
  - d) Вещественные числа, символьные и логические значения.
  
3. Какие типы данных относятся к структурированным типам?
  - a) Структурированные типы включают массивы, строки, записи, файлы, списки, стеки, очереди, деревья, графы и таблицы;
  - b) К структурированным типам относятся двумерные массивы, записи, файлы, типы перечисления и указатели;
  - c) Структурированными типами являются двумерные массивы, указатели, списки очереди, деревья графы и таблицы.
  - d) Целые и вещественные числа, символьные и логические значения, т.к. объекты этих типов не могут быть разделены на меньшие части.
  
4. Что представляют собой классы и объекты?
  - a) Классы – это определяемые пользователем типы данных с данными и функциями - членами класса. Переменная типа класс называется объектом;
  - b) Классы это структурированные типы данных, состоящие из членов, которые включают значения данных и операции по обработке этих данных. Объектом является переменная член класса;
  - c) Классы – это абстрактные типы данных, определяющие область и структуру данных вместе с набором операций. Объекты это переменные класса.
  - d) Классы и объекты – это экземпляры пользовательского типа.
  
5. Какие основные принципы поддерживаются ООП?
  - a) Принципы инкапсуляции, наследования и полиморфизма.
  - b) Принципы множественного наследования и полиморфизма.
  - c) Принципы совместимости типов, инкапсуляции и наследования.
  - d) Принципы инкапсуляции, изменчивости и наследования.

## Образцы экзаменационных билетов по дисциплине


КГТУ им. И. Раззакова		Кафедра ПМ и И
Дисциплина «Объектно-ориентированное программирование и системы программирования»		

### Экзаменационный билет № 1

1. Основные принципы процедурного программирования. Проявите реализацию принципов процедурного программирования на примере решения задачи интерполирования на C++.	10 б.
2. Основные принципы структурного программирования. Как изменится решение предыдущего примера при реализации принципов структурного программирования?	8 б.
3. Принцип инкапсуляции. Реализация принципа инкапсуляции в классах.	4 б.
4. Как объявить класс с закрытыми переменными-членами и открытыми методами доступа? Проявите на примере класса <i>Сотрудники</i> .	9 б.
5. Определите объект класса с закрытыми переменными-членами. Проявите на примере класса <i>Сотрудники</i> .	9 б.
всего	40 б.

Зав. кафедрой

д.ф.-м.н., проф. М. Дж. Джаманбаев

КГТУ им. И. Раззакова		Кафедра ПМ и И
Дисциплина «Объектно-ориентированное программирование и системы программирования»		

### Экзаменационный билет № 2

1. Основные принципы объектно-ориентированного программирования. Проявите реализацию любого одного из принципов на примере создания класса <i>Массив</i> на C++.	10 б.
2. Объявите класс <i>Student</i> с статическими элементами и статическими методами. В чем состоит отличие статических элементов класса от других ее элементов?	6 б.
3. Объявление и создание динамических структур данных. Применение указателей. Покажите на примере работы с одномерными массивами.	9 б.
4. Перегрузка операторов. Проявите на примере реализации класса <i>Треугольники</i> .	9 б.
5. Шаблоны классов. Определение шаблона класса. использование шаблона класса. Организация исходного кода. Параметры шаблонов.	6 б.
всего	40 б.

Зав. кафедрой

д.ф.-м.н., проф. М. Дж. Джаманбаев


КГТУ им. И. Раззакова		Кафедра ПМ и И
Дисциплина «Объектно-ориентированное программирование и системы программирования»		

**Экзаменационный билет № 3**

1. Основные понятия ООП - классы и члены, определение объекта. Объекты и передача информации. Приведите примеры классов и объектов.	6 б.
2. Интегрированная среда разработки (IDE) Visual Studio 2010/2013. Методология разработки приложений в IDE. Основные команды управления разработкой, сопровождением и выполнением приложений.	6 б.
3. Разработка класса <i>Point</i> . Интерфейс класса. Использование конструкторов и деструкторов. Разновидности конструкторов. Конструкторы и деструкторы, заданные по умолчанию, конструктор умолчания. Как работает конструктор копирования?	9 б.
4. Стандартные потоки. Классы стандартных потоков. Объекты и методы стандартных потоков ввода/вывода. Заголовочные файлы библиотеки ввода/вывода C++. Приведите примеры.	9 б.
5. Понятие полиморфизма функций и классов. Разновидности полиморфизма. Приведите примеры.	6 б.
всего	40 б.

Зав. кафедрой

д.ф.-м.н., проф. М. Дж. Джаманбаев

КГТУ им. И. Раззакова		Кафедра ПМ и И
Дисциплина «Объектно-ориентированное программирование и системы программирования»		

**Экзаменационный билет № 4**

1. Реализация принципа наследования классов. Приведите пример взаимоотношения – отношение наследования (отношение родитель /потомок) на примере классов: <i>треугольник, многоугольник</i> .	9 б.
2. Разработка объектно-ориентированного приложения: отделение реализации от интерфейса. Разбивка на ряд файлов: определение класса в заголовочном файле, определение функций-элементов этого класса в файле исходных кодов с теми же базовыми именами.	7 б.
3. Принцип инкапсуляции. Реализация принципа инкапсуляции в классах. Продемонстрируйте на примере класса <i>Сотрудники</i> .	5 б.
4. Создание динамического массива заданного размера, выделение из массива указанных в варианте диапазонов. Выполните сортировку массива по возрастанию.	14 б.
5. Ссылки. Передача аргументов функции как ссылок. Возвращение значений с помощью ссылок.	5 б.
всего	40 б.

Зав. кафедрой

д.ф.-м.н., проф. М. Дж. Джаманбаев

## Глоссарий

**Абстрактный класс** - это класс, содержащий хотя бы один виртуальный метод. Абстрактные классы не бывают изолированными, т.е. всегда абстрактный класс должен быть наследуемым. Поскольку у чисто виртуального метода нет тела, то создать объект абстрактного класса невозможно. Абстрактным классом можно назвать класс, специально определенный для обеспечения наследования характеристик порожденными классами.

**Абстракция** - процесс изменения уровня детализации программы. Когда мы абстрагируемся от проблемы, мы предполагаем игнорирование ряда подробностей с тем, чтобы свести задачу к более простой.

**Абстракция через параметризацию** - прием программирования, позволяющий, используя параметры, представить фактически неограниченный набор различных вычислений одной программой, которая есть абстракция этих наборов.

**Абстракция через спецификацию** - прием программирования, позволяющий абстрагироваться от процесса вычислений описанных в теле процедуры, до уровня знания того, что данная процедура делает. Это достигается путем задания спецификации, описывающей эффект ее работы, после чего смысл обращения к данной процедуре становится ясным через анализ этой спецификации, а не самого тела процедуры. Мы пользуемся абстракцией через спецификацию всякий раз, когда связываем с процедурой некий комментарий, достаточно информативный для того, чтобы иметь возможность работать без анализа тела процедуры. Абстракция через спецификацию позволяет абстрагироваться от процесса вычислений описанных в теле процедуры, до уровня знания того, что данная процедура делает. Это достигается путем задания спецификации, описывающей эффект ее работы, после чего смысл обращения к данной процедуре становится ясным через анализ этой спецификации, а не самого тела процедуры. Мы пользуемся абстракцией через спецификацию всякий раз, когда связываем с процедурой некий комментарий, достаточно информативный для того, чтобы иметь возможность работать без анализа тела процедуры.

**Аспектно-ориентированное сборочное программирование**-аспектно-ориентированное сборочное программирование - разновидность сборочного программирования, основанная на сборке полнофункциональных приложений из многоаспектных компонентов, инкапсулирующих различные варианты реализации

**Ассоциация** - отношение, означающее некоторую смысловую связь между классами.

**Данные** – представление информации в формальном виде, пригодном для передачи, интерпретации или обработки людьми или компьютером.

**Диаграмма деятельности** - методология объектно-ориентированного проектирования, предназначенная для детализации особенностей алгоритмической и логической организации системы. При этом каждое действие расчленяется на фундаментальные процессы. На диаграмме деятельности управление осуществляется:

- либо через потоки управления (явно);
- либо через определяемые потоки данных (неявно).

**Диаграмма классов** - методология объектно-ориентированного проектирования, предназначенная для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования.

**Диаграмма компонентов** - метод объектно-ориентированного проектирования, описывающий особенности физического представления системы. Диаграмма компонентов позволя-



ет определить архитектуру разрабатываемой системы, устанавливая зависимости между компонентами.

**Диаграмма состояний** - методология объектно-ориентированного проектирования, предназначенная для представления жизненного цикла объектов в реальном или абстрактном мире.

Диаграмма состояний состоит

- из множества состояний объектов;
- из множества событий, сообщающих о перемещении чего-либо в новое состояние;
- из множества правил переходов, определяющих новое состояние объекта при возникновении тех или иных событий;
- из множества действий, которые должны быть выполнены объектом, когда он переходит в новое состояние.

**Инкапсуляция** - в объектно-ориентированном программировании - сокрытие внутренней структуры данных и реализации методов объекта от остальной программы. Другим объектам доступен только интерфейс объекта, через который осуществляется все взаимодействие с ним.

**Исполняемый код** – объединенные в загружаемую программу модули объектного кода. Объединение модулей объектного кода в единую исполняемую программу является результатом работы компоновщика.

**Инструментальная среда** – это интегрированная система, которая позволяет писать, редактировать, отлаживать и запускать программы на выполнение, не выходя из самой среды.

**Интерпретируемые языки.** К этой группе относят языки программирования, исходный код которых считывается и выполняется специальной программой-интерпретатором инструкции за инструкцией. Интерпретируемыми являются большинство версий языков Basic и Forth.

**Информационная технология** - система методов, приемов, стандартизированных спецификаций и средств, обеспечивающая организацию информационных процессов.

**Исходный код** – текст компьютерной программы, написанный на выбранном языке программирования. Исходный код преобразуется транслятором (специальной программой) в объектный модуль, который хранится на диске.

**Класс** - в программировании - множество объектов, которые обладают одинаковой структурой, поведением и отношением с объектами из других классов.

**Компилируемые языки.** К этой группе относят языки программирования, исходный код которых преобразуется специальной программой-компилятором в объектные модули, которые затем собираются при помощи программы сборки (линковщика) в единый загружаемый модуль. Этот модуль представляет собой выполняемую программу. Компилируемыми являются языки C, C++, Object Pascal.

**Машинный код** – язык конкретной вычислительной машины (центрального процессора), инструкции которого выполняются непосредственно центральным процессором или его микропрограммами.

**Модель** - (лат. *modulus* — мера, образец), в широком смысле — любой образ, аналог (мысленный или условный: изображение, описание, схема, чертеж, график, план, карта и т. п.)

какого-либо объекта, процесса или явления («оригинала» данной модели), используемый в качестве его «заместителя», «представителя»

**Моделирование** - исследование каких-либо явлений, процессов или систем объектов путем построения и изучения их моделей; использование моделей для определения или уточнения характеристик и рационализации способов построения вновь конструируемых объектов. Моделирование — одна из основных категорий теории познания: на идее моделирования по существу базируется любой метод научного исследования — как теоретический (при котором используются различного рода знаковые, абстрактные модели), так и экспериментальный (использующий предметные модели).

**Наследование** - в объектно-ориентированном программировании - свойство объекта, заключающееся в том, что характеристики одного объекта (объекта-предка) могут передаваться другому объекту (объекту-потомку) без их повторного описания. Наследование упрощает описание объектов.

**Объект** - в программировании - программный модуль:

- объединяющий в себе данные (свойства) и операции над ними (методы);
- обладающий свойствами наследования, инкапсуляции и полиморфизма.

Объекты взаимодействуют между собой, посылая друг другу сообщения.

**Объектный модуль** – последовательность инструкций машинного кода, являющаяся результатом работы компилятора

**Объектно-ориентированное программирование** - Объектно-ориентированное программирование - технология программирования, при которой программа рассматривается как набор дискретных объектов, содержащих, в свою очередь, наборы структур данных и процедур, взаимодействующих с другими объектами.

**Полиморфизм** - в объектно-ориентированном программировании - способность объекта выбирать правильный метод в зависимости от типа данных, полученных в сообщении.

**Программирование** – это совокупность процессов, связанных с разработкой программ и их реализацией. В широком смысле к указанным процессам относят все технические операции, необходимые для создания программ, включая анализ требований, а также все стадии разработки и реализации в виде готового программного продукта. На рисунке показан приблизительный порядок действий при программировании. Программирование заключается в написании текста (исходного кода) программы на выбранном языке программирования. Схематичное описание процесса программирования:

- Разработка исходного кода.
- Трансляция компьютерной программы с языка программирования в программу на машинном коде.
- Для выполнения программы должен быть определенным образом подготовлен объектный модуль – определены адрес его загрузки в оперативную память и адреса связи с другими объектными модулями. Затем загружается объектный модуль в оперативную память. Объектный модуль – последовательность инструкций машинного кода, являющаяся результатом работы компилятора.
- Объединение модулей объектного кода в единую исполняемую программу является результатом работы компоновщика.

**Свойство объекта** - в объектно-ориентированном программировании - характеристика объекта. Обычно свойства изменяются с помощью методов.

**Система программирования** - программная система, предназначенная для разработки программ на конкретном языке программирования. Система программирования предоставляет пользователю специальные средства разработки программ: транслятор, (специальный) редактор текстов программ, библиотеки стандартных подпрограмм, программную докумен-

тацию, отладчик и др. Система программирования – комплекс языковых и программных средств, предназначенных для автоматизации процесса составления, отладки программы и подготовки ее к выполнению.

**Событийно-управляемое программирование** - объектно-ориентированное программирование, при котором задаются реакции программы на различные события.

**Технология программирования** - дисциплина, изучающая технологические процессы программирования и порядок их прохождения.

**Транслятор** – программа, осуществляющая трансляцию (перевод) компьютерной программы с языка программирования в программу на машинном коде с сохранением эквивалентности обеих программ. Процесс трансляции состоит из нескольких этапов.

1. Производится лексический анализ текста с разбиением его на элементы-лексемы (ключевые слова, имена-идентификаторы и т. д.).
2. Производится синтаксический разбор, то есть проверка правильности и допустимости созданных на основе лексем конструкций языка.
3. Исходный код преобразуется в исполняемый машинный код.

**Файловая система.** Общая структура, определяющая в операционной системе наименование, сохранение и размещение файлов. Различными типами файловых систем являются системы NTFS, FAT и FAT32.

**Экземпляр объекта** - в объектно-ориентированном программировании - конкретный объект из набора объектов данного класса. Все экземпляры одного класса имеют одинаковый набор операций.

**Язык программирования** предназначен для описания данных и алгоритмов их обработки на вычислительной машине. Языки программирования занимают промежуточное положение между естественными и формализованными языками. С естественными языками языки программирования роднит грамматический строй (употребление слов естественного языка, фразовая структура и т. п.), с формализованными языками – символы и понятия, а главное – строгие, точно описанные правила построения текстов.

- По факту созданию процесса языки программирования делятся на **создающие процесс и сценарные**. Языки, **создающие процесс**. После запуска программы создается отдельный процесс выполнения этой программы. Так происходит в языках C, C++, Object Pascal.
- **Сценарные языки**. Сценарий, или скрипт, – это программа, которую выполняет другая программа. В качестве примером можно привести скрипты оболочки в UNIX, программы на языках PHP, Python, Ruby.
- **Автономные языки** программирования (C, C++, Java) являются автономным инструментом для создания программ.
- **Встроенные языки** программирования являются частью какой-то системы и позволяют создавать программы, предназначенные для работы только в этой системе. Пример наиболее известного встроенного языка – VBA (Visual Basic for Applications), который используется только внутри приложений Microsoft Office для автоматизации и расширения их функциональности. Встроенными являются также язык программирования системы IC и язык JavaScript, который выполняется только внутри интернет-приложений.

## Список литературы

### ОСНОВНАЯ ЛИТЕРАТУРА

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. 3-е изд./ Пер. с англ. – М.: ООО «И.Д.Вильямс», 2008. - 720 с.
2. Т.Бадд. Объектно-ориентированное программирование. Перев. с англ. — СПб.: Питер, 1997.
3. Р. Лафоре. Объектно-ориентированное программирование в С++. 4-е изд.: Пер. с англ. СПб.: «Питер», 2004. - 928с.
4. Страуструп Бьярне. Программирование: принципы и практика с использованием С++, 2-е изд.: Пер. с англ. М. : ООО "И .Д. Вильяме", 2016. - 1328с.
5. Шилдт Г. С++: базовый курс, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2010. – 624 с.
6. Дейтел Х., Дейтел П. как программировать на С++. Третье издание: пер. с англ.- М.: Издательство БИНОМ, 2003 г. -1152 с.

### ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

7. Кнут Д.Е. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. пособие – М.: Издательский дом « Вильямс», 2000. – 720 с.
8. Павловская Т.А., Ю.А.Щупак. С++. Объектно-ориентированное программирование. Практикум –СПб. Питер.2006. -265 с.

### Интернет - ресурсы

1. <http://www.osp.ru>. -Открытые системы [Интернет-ресурсы]: интернет-издания по информационным технологиям
2. <http://ru.wikipedia.org>. – Свободная энциклопедия «Википедия» [Интернет-ресурсы]: интернет-энциклопедия.
3. <http://www.edu.ru/>
4. <http://soft.cnews.ru/>
5. Викиучебник. Объектно-ориентированное программирование [Электронный ресурс] — Режим доступа: [http://ru.wikibooks.org/wiki/ Объектно-ориентированное программирование](http://ru.wikibooks.org/wiki/Объектно-ориентированное_программирование)
6. <http://www.citforum.ru/> (сайт аналитической информации).
7. <http://dcprograms.narod.ru/> (языки и среды программирования).

## СИЛЛАБУС

Дисциплина	«Объектно-ориентированное программирование и системы программирования»
Код дисциплины	187.М1.П1.
Семестр	Магистратура, I семестр
Кредиты	4
Курс	1
Лекции	32 ч
Лабораторные занятия	32 ч
Количество рубежных контролей (РК)	2
СРС	56 ч
Экзамен	1
Общая трудоемкость	120 ч
Место проведения занятий	ауд. №2/515

### Данные о преподавателе и контактная информация

<b>Преподаватель:</b>	Кабаева Гульнара Джамалбековна
<b>Ученая степень, звание, должность:</b>	д. ф.-м.н., профессор
<b>Стаж работы:</b>	32 лет
<b>Кабинет:</b>	2/306
<b>Контактные телефоны:</b>	0555725088
<b>Время пребывания на кафедре ПМИИ:</b>	по средам с 8 <sup>00</sup> до 11 <sup>00</sup> , ауд. 2/515.
<b>Время консультаций:</b>	должно быть заранее согласовано с преподавателем

### *Цель и задачи дисциплины*

Целью дисциплины является освоение магистрантами принципов и технологии применения объектно-ориентированного программирования (ООП), формирование у них систематизированных знаний и практических навыков создания объектно-ориентированного программного обеспечения.

Задачи дисциплины: дать базовые знания о проектировании и разработке приложений с применением объектно-ориентированного подхода; формирование представлений об общей методологии и средствах технологии объектно-ориентированного программирования; практическое освоение технологии объектно-ориентированного программирования.

В результате освоения дисциплины «Объектно-ориентированное программирование и системы программирования» магистрант должен:

#### **знать**

- основные понятия, принципы и модели ООП: пользовательские типы данных – классы, объект, объявление и реализацию класса, данные, методы; доступ, наследование свойств, виртуальные функции;
- алгоритмы и методы проектирования объектно-ориентированных программ;
- средства и подходы конструирования собственных пользовательских классов; объектно-ориентированные языки, язык C++/C#;
- правила организации, методы и средства программирования интерфейса; системы программирования; объектно-ориентированные системы;

#### **уметь**

- применять при разработке, анализе и проверке программ принципы объектно-ориентированного программирования – такие, как инкапсуляция, полиморфизм, наследование;
- определять классы, создавать объекты, разрабатывать и использовать конструкторы класса; создавать программы, используя библиотеки классов;
- выбрать и использовать алгоритмы и методы проектирования объектно-ориентированных программ;
- использовать правила организации и программирования интерфейса; использовать языки, методы и средства программирования;

#### **владеть**

- навыками объектно-ориентированного программирования на C++; основными методами и приемами, и средствами разработки объектно-ориентированных программ.

**Пререквизиты** – «Информатика», «Математическая логика и теория алгоритмов», «Языки и методы программирования», «Операционные системы».

**Постреквизиты** – все дисциплины профессионального цикла, связанные с проектированием программного обеспечения, дипломное проектирование.

### Темы лекционных занятий

Мо-ду-ли	Лек-ция №	Наименование изучаемых вопросов	Кол-во часов	Лите-ратура №
<b>Модуль 1</b>	1	<b>Парадигмы программирования:</b> процедурное, структурированное и объектно-ориентированное программирование. Языки ООП. Системы ООП.	4	[1],[3]
	2	<b>Основные понятия ООП.</b> Принципы объектно-ориентированного программирования. Основные понятия ООП.	4	[1],[3],[4]
	3	<b>Динамические структуры данных.</b> Указатели. Указатель this. Статические члены: функции и данные. Указатели на члены. Структуры и объединения. Динамически создаваемые объекты. Размещение объектов в динамически распределяемой памяти, доступ к членам класса. Динамическое размещение членов класса. Использование указателей, как средство хранения адреса. Ссылки.	5	[2],[3],[4]
	4	<b>Массивы объектов.</b> Объявление массивов в области динамического обмена. Удаление массива из области динамического обмена. Классы массивов. Линейные списки. Стеки. Очереди. бинарные деревья.	3	[2],[3],[4]
		<b>Модульно-рейтинговый контроль – контрольная работа</b>		16

<b>Модуль 2</b>	5	<b>Дружественные функции и перегрузка операций.</b> Функции-друзья. Перегрузка бинарных и унарных операций. Перегруженные операции.	4	[1],[3],[4]
	6	<b>Перегруженные функции-члены, перегрузка операторов.</b> Перегруженные функции-члены. Перегрузка конструкторов.	2	[1],[3],[4]
	7	<b>Иерархия и наследование.</b> Наследование, иерархия и наследование. Типы наследования.. Виртуальные методы, функции, деструкторы, виртуальный конструктор копирования. Полиморфизм.	4	[1],[3],[4]
	8	<b>Классы потоков C++.</b> Заголовочные файлы. Предопределенные объекты и потоки. Операции помещения и извлечения. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоков C++.	2	[1],[3],[4]
	9	<b>Пространства имен.</b> Пространства имен, создание и использование, стандартное пространство имен. Анализ и проектирование объектно-ориентированных программ. Шаблоны, объявление шаблона, функции шаблона, шаблоны и друзья, стандартная библиотека шаблонов. Контейнеры. Препроцессор и компилятор. Компилятор Visual C++ .	4	[1],[3],[4]
	<b>Модульно-рейтинговый контроль – контрольная работа</b>		16	
	<b>Всего:</b>		32	

### Лабораторные занятия

Мо-ду-ли	Лаб. ра-бота № / не-деля	Темы занятий	Ко-л-во часов	Лите-ратура №	Форма кон-троля и бал-лы
<b>Модуль 1</b>	1/ 1	Основы программирования на C++ в среде Visual Studio 2010/2013. Настройки среды. <i>Создание консольного проекта.</i>	2	[3],[4]	Отчет, 2
	2/ 2	ООП на языке C++. приемы программирования. <i>Создание консольного приложения.</i> Объявление переменных. Арифметические выражения. Построение и запуск. Отладочные механизмы.	2	[3],[4],[5]	Отчет, 2
	3/ 3-4	Типы данных в C++: значимые и ссылочные типы. Классы. Основные понятия ООП, объявление и реализация класса, объявление объектов, создание объектов. Доступ к членам класса. Определение методов класса. Объявление массивов.	4	[3],[4],[5]	Отчет, 2
	4/ 5-6	Классы, объекты, методы,. Модификаторы доступа. Интегрированная среда разработки Visual Studio 2010/2013. Методология разработки приложений в IDE . Отделение реализации от интерфейса.	4	[3],[4]	Отчет, 2
	5/ 7-8	Решение задачи поиска в массиве структур. Разработка простой базы данных сотрудников и класса <i>Сотрудник</i> . Использование конструкторов и деструкторов. Разновидности конструкторов.	4	[3],[4]	Отчет, 3
	<b>Модульно-рейтинговый контроль</b>			<b>16</b>	<b>14</b>
Мо	Лаб.	Темы занятий	Ко	Лите-ра-	Форма

ду-ли	ра-бота № / не-деля		л-во ча-сов	тура №	кон-троля и бал-лы
Модуль 2	6 / 9	Объявление и создание динамических структур данных. Указатели. Указатель this. Статические члены: функции и данные.	2	[3], [4]	Отчет, 2
	7 / 10-11	Структуры и объединения. Константные члены-функции и константные объекты. Динамически создаваемые объекты. Структуры. Размещение объектов в динамически распределяемой памяти, доступ к членам класса.	4	[3], [4], [5]	Отчет, 2
	8 / 12	Реализация класса треугольников. Использование перегрузки операторов. Перегруженные функции-члены. Перегрузка конструкторов	2	[3], [4]	Отчет, 3
	9 / 13-14	Разработка приложения с реализацией наследования классов. Взаимоотношение – отношение наследования на примере классов: <i>треугольник, четырехугольник, многоугольник.</i>	4	[3], [4]	Отчет, 3
	10 / 15	Шаблоны классов. Определение шаблона класса. использование шаблона класса. Организация исходного кода. параметры шаблонов	2	[3], [4]	Отчет, 2
	11/16	Разработка приложения с реализацией файлов и потоков в C++.	2	[3],[4]	Отч, 2
<b>Модульно-рейтинговый контроль</b>			<b>16</b>		<b>14</b>
<b>Всего:</b>			<b>32</b>		<b>28</b>

### Самостоятельная работа магистров

Модули	№	Темы заданий	Ко л-во ча-сов	Форма контроля	Рекомендуемая литература	Мах баллов
Модуль 1	1	Разработать программу вычисления значений функций одной переменной на заданном отрезке, с заданным шагом.	6	Отчет с результатом работы	[1], [7]	1
	2	Создание динамического массива заданного размера, выделение из массива указанных в варианте диапазонов.	6	Отчет с результатом работы	[1], [7]	1
	3	Доработать программу калькулятор из лабораторной работы 8. Добавить функции.	6	Отчет	[1],	1
	4	Разработать на C++ программу реализации задачи интерполирования при заданных табличных данных. Программа должна предлагать выбор формулы интерполирования. Ввод данных осуществляется с клавиатуры.	12	Отчет с результатом работы	[1],[3], [5],[6], [7]	2
Модуль 2	5	Реализация численных методов решения систем линейных уравнений на C++ на основе объектно-ориентированного подхода, на основе класса массивов.	8	Отчет с результатом работы	[3], [5], [6]	2
	6	Разработка на C++ приложения для статистической обработки массива случайных данных. Генератор случайных чисел. Реализация метода наименьших квадратов для решения уравнения линейной регрессии.	8	Отчет с результатом работы	[1],[3], [5],	1
	7	Разработать шаблонный класс для представления динамических одномерных массивов. Класс обеспечивает хранение данных любого типа, для него предусмотрены конструктор по умолчанию, конструктор копирования и операция присваивания.	10	Отчет с результатом работы	[1],[3], [5], [7], [8],	2
<b>Всего</b>			<b>56</b>			<b>10</b>

### 8. Требования и оценка курса

Знания оцениваются посредством проведения двух модулей и экзамена в конце семестра. Каждый модуль оценивается максимально в 30 баллов. Всего за два модуля 60 баллов и за итоговый контроль 40 баллов. По итогам работы за семестр и экзамен определяется суммарный балл успеваемости, максимальное значение которого составляет - 100 баллов.

- Пропуски занятий по неуважительной причине (0,75 баллов за каждое занятие)
- Несвоевременный отчет по лабораторным работам (0,5 баллов за каждое занятие)
- Несвоевременное выполнение самостоятельной работы - 1 балл.

Модули	Виды активности	Оценка в баллах		Сроки текущего контроля
		min	max	
1	Теоретические знания	5	8	7 неделя

	Лабораторные занятия	11	14	1 семестр
	СРС	3	5	
	Посещаемость	2	3	
	Всего	21	30	
2	Теоретические знания	5	8	16 неделя 1 семестр
	Лабораторные занятия	11	14	
	СРС	3	5	
	Посещаемость	2	3	
	Всего	21	30	
Итоговый контроль	Итоги по текущим рейтингам	41	60	
	Экзамен	19	40	
<b>Итого:</b>		<b>61</b>	<b>100</b>	

#### Информация по оценке

Рейтинг (баллы)	Оценка по буквенной системе	Цифровой эквивалент оценки	Оценка по традиционной системе (4-х бальной)
87 - 100	A	4,0	Отлично
80 - 86	B	3,33	Хорошо
74 - 79	C	3,0	
68 - 73	D	2,33	Удовлетворительно
61 - 67	E	2,0	
41 - 60	FX	0	Неудовлетворительно
0 - 40	F	0	

#### Политика курса

В процессе совместной работы следует придерживаться следующих правил:

1. Преподаватель и магистр должны относиться друг к другу с уважением.
2. Не опаздывать, не пропускать занятия без уважительной причины.
3. Пропущенное занятие можно отработать (не позже срока, указанного преподавателем).

#### ОСНОВНАЯ ЛИТЕРАТУРА

9. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 3-е изд./ Пер. с англ. – М.: ООО «И.Д.Вильямс», 2008. - 720 с.
10. Т.Бадд. Объектно-ориентированное программирование. Перев. с англ. — СПб.: Питер, 1997.
11. Р. Лафоре. Объектно-ориентированное программирование в C++. 4-е изд.: Пер. с англ. СПб.: «Питер», 2004. - 928с.
12. Страуструп Бьярне. Программирование: принципы и практика с использованием C++, 2-е изд.: Пер. с англ. М. : ООО "И .Д. Вильяме", 2016. - 1328с.
13. Шилдт Г. C++: базовый курс, 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2010. – 624 с.
14. Дейтел Х., Дейтел П. как программировать на C++. Третье издание: пер. с англ.- М.: Издательство БИ-НОМ, 2003 г. -1152 с.

#### ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

15. Кнут Д.Е. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. пособие – М.: Издательский дом « Вильямс», 2000. – 720 с.
16. Павловская Т.А., Ю.А.Щупак. C++. Объектно-ориентированное программирование. Практикум –СПб. Питер.2006. -265 с.

#### Интернет - ресурсы

8. <http://www.osp.ru>. -Открытые системы [Интернет-ресурсы]: интернет-издания по информационным технологиям
9. <http://ru.wikipedia.org>. – Свободная энциклопедия «Википедия» [Интернет-ресурсы]: интернет-энциклопедия.
10. <http://www.edu.ru/>
11. <http://soft.cnews.ru/>
12. Викиучебник. Объектно-ориентированное программирование [Электронный ресурс] — Режим доступа: [http://ru.wikibooks.org/wiki/ Объектно-ориентированное программирование](http://ru.wikibooks.org/wiki/Объектно-ориентированное_программирование)
13. <http://www.citforum.ru/> (сайт аналитической информации).
14. <http://dcprograms.narod.ru/> (языки и среды программирования).