


МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая 2023 г.

Рабочая программа дисциплины
«Объектно-ориентированное программирование»

Направление подготовки
01.03.02 Прикладная математика и информатика

Направленность (профиль)
«Искусственный интеллект»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена на
заседании кафедры
от 21 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Объектно-ориентированное программирование» являются изучение основных этапов, методов, средств и стандартов разработки программного обеспечения, детальное изучение одного из объектно-ориентированных языков программирования (Java), обучение студентов разработке кросс-платформенных приложений. Результаты изучения дисциплины «Объектно-ориентированное программирование» востребованы при освоении дисциплин модулей: «Машинное обучение и анализ данных» и «Искусственный интеллект», преддипломной практике и выпускной квалификационной работе.

2. Место дисциплины в структуре образовательной программы бакалавриата (магистратуры, специалитета)

Дисциплина «Объектно-ориентированное программирование» согласно учебному плану входит в модуль «Алгоритмизация и программирование» и реализуется в 5-6 семестрах. Изучается на основе знаний, полученных при изучении дисциплин модулей «Аппаратное и программное обеспечение компьютера», «Современные цифровые технологии».

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы бакалавриата (магистратуры, специалитета)

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач. ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы,	ИОПК2.1 Осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач. ИОПК5.1 Обладает знаниями в области алгоритмизации и программирования.	Самостоятельно и грамотно осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач, в том числе в новой или нестандартной ситуации. Демонстрирует глубокие знания в области алгоритмизации и программирования.

пригодные для практического применения.	ИОПК5.2 Демонстрирует умение выбрать и обосновать выбор языка и среды программирования для разработки компьютерных программ.	Умеет выбирать и обосновать выбор языка и среды программирования для разработки компьютерных программ, в том числе в новой или нестандартной ситуации.
	ИОПК5.3 Владеет навыками разработки алгоритмов и компьютерных программ, пригодных для практического применения.	Полностью самостоятельно и верно разрабатывает алгоритмы и компьютерные программы, пригодные для практического применения.
Профессиональные компетенции		
ПК-1. Способен проектировать компьютерное программное обеспечение.	ИПК1.1 Разрабатывает и изменяет архитектуру компьютерного программного обеспечения.	Демонстрирует глубокое знание и понимание принципов построения и видов архитектуры компьютерного программного обеспечения, типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке компьютерного программного обеспечения, методов и средств проектирования компьютерного программного обеспечения. Полностью верно и самостоятельно использует существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения, применяет методы и средства проектирования компьютерного программного обеспечения.
	ИПК1.2 Проектирует структуру данных, баз данных и программных интерфейсов.	Глубоко знает, полностью верно и самостоятельно умеет применять методы и средства проектирования баз данных и программных интерфейсов.

	<p>ИПК1.3 Разрабатывает техническую документацию на компьютерное программное обеспечение с использованием существующих стандартов, оценивает и согласовывает сроки выполнения поставленных задач.</p>	<p>Демонстрирует свободное владение стандартами в области разработки компьютерного программного обеспечения. Полностью верно и самостоятельно разрабатывает техническую документацию, оценивает и согласовывает сроки выполнения поставленных задач.</p>
<p>ПК-9. Способен оценить качество разрабатываемого программного обеспечения путем проверки соответствия продукта заявленным требованиям, сбора и передачи информации о несоответствиях.</p>	<p>ИПК9.1 Демонстрирует умение определять и описывать тестовые случаи на основе требований, заявленных к программному обеспечению.</p>	<p>Глубоко знает и понимает классификацию видов и типов тестирования программного обеспечения, техники проектирования и комбинаторики тестов, тестовые данные, обеспечивающие проверку безопасности программного обеспечения. Полностью верно и самостоятельно применяет техники проектирования тестов, анализирует тестовые случаи на предмет полноты учета покрытия, документирует тесты, разрабатывает скрипты и/или программные модули для автоматизации тестирования программного обеспечения, в том числе для проверки информационной безопасности разрабатываемого программного обеспечения.</p>
	<p>ИПК9.2 Проводит тестирование по разработанным тестовым случаям, осуществляет сбор информации о несоответствиях заявленным требованиям.</p>	<p>Самостоятельно и полностью верно выполняет начальные настройки для проведения тестирования, необходимые виды тестирования.</p>

	<p>ИПК9.3 Анализирует результаты тестирования и дает оценку качеству разрабатываемого программного обеспечения.</p>	<p>Демонстрирует глубокое знание и понимание типов дефектов программного обеспечения, их классификации и статистики возникновения. Полностью верно и самостоятельно определяет уровень критичности дефектов программного обеспечения, составляет отчеты об анализе результатов тестирования программного обеспечения.</p>
--	---	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа					самостоятельная работа	
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1	Программирование на языке C++.	5	8	0	8	0	0	30	
2	Разработка приложений с графическим интерфейсом.		8	0	12	0	0	32	Зачет
3	Структуры данных и алгоритмы обработки данных.	6	8	0	14	0	0	40	
4	Прикладные решения на базе системы «1С:Предприятие (учебная версия)».	6	6	0	14	0	0	36	
	ИТОГО		30	0	48	0	0	138	Экзамен

Содержание разделов дисциплины:

Раздел 1. Программирование на языке C++.

Синтаксис языка. Типы данных: понятие, простые типы данных, примеры типов, совместимость и преобразование типов. Переменные, операции и выражения. Операторы. Структура программы. Организация ввода и вывода. Основные алгоритмические конструкции на примере языка C++: операторы ветвления, операторы цикла. Указатели. Динамические переменные. Массивы. Строковый тип данных. Структуры. Объединения. Функции. Файлы

Раздел 2. Разработка приложений с графическим интерфейсом.

Технологии разработки приложений с графическим интерфейсом: WPF; WindowsForms. Работа с формами: основные свойства форм; добавление форм; взаимодействие между формами; события форм. Элементы управления, их основные свойства, методы и события. Элементы управления для ввода/вывода. Переключатели, счетчик. Списки. Стандартные диалоги. Меню и панели инструментов.

Раздел 3. Структуры данных и алгоритмы обработки данных.

Задача сортировки. Базовые методы сортировки. Улучшенные методы сортировки.

Динамические переменные, ссылочные типы данных. Абстрактные типы данных. Линейные список Основные понятия теории графов. Обход графов в длину и ширину ширину. Алгоритм Дейкстры. Алгоритм Флойда. Транзитивное замыкание. Обход графов в глубину. Циклы в графах. Хроматическое число графов. Полный перебор. Перебор циклами, рекурсивный перебор, р-ичный перебор. Жадные алгоритмы. Метод динамического программирования. Прямой поиск. Бинарный поиск. Поиск Фибоначчи. Дерево поиска. AVL деревья. Хеширование. Хештаблицы.

Раздел 4. Прикладные решения на базе системы «1С:Предприятие (учебная версия)».

Назначение и состав базовых решений на основе системы «1С:Предприятие (учебная версия)». Цели, задачи, типы, подходы в разработке прикладных решений.

Проектирование прикладных решений на базе «1С:Предприятие (учебная версия)» с использованием встроенного языка программирования.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекция-беседа или «диалог с аудиторией», является наиболее распространенной и сравнительно простой формой активного вовлечения студентов в учебный процесс. Эта лекция предполагает непосредственный контакт преподавателя с аудиторией. Преимущество лекции-беседы состоит в том, что она позволяет привлекать внимание студентов к наиболее важным вопросам темы, определять содержание и темп изложения учебного материала с учетом особенностей студентов.

Мастер-класс– это особая форма учебного занятия, когда преподаватель-мастер передает свой опыт путем прямого и комментированного показа последовательности действий, методов, приемов и форм педагогической деятельности. Целью проведения мастер-класса является профессиональное, интеллектуальное и эстетическое воспитание студентов, и прежде всего, развитие в ходе мастер-класса способности студента самостоятельно и нестандартно мыслить.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

1. ОС семейства MicrosoftWindows
2. LibreOffice
3. MozillaFirefox
4. Microsoft Office

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», рекомендуемых для освоения дисциплины

а) основная литература

1. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2022. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/206699>
2. Скворцова, Л. А. Объектно-ориентированное программирование на языке C++ : учебное пособие / Л. А. Скворцова. — Москва : РТУ МИРЭА, 2020. — 246 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/163862>
3. Зайцев, М. Г. Объектно-ориентированный анализ и программирование : учебное пособие / М. Г. Зайцев. — Новосибирск : НГТУ, 2017. — 84 с. — ISBN 978-5-7782-3308-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118271>

б) дополнительная литература

1. Хорев П.Б. Объектно-ориентированное программирование : учебное пособие для вузов. - 4-е изд. - Москва : ИЦ "Академия", 2012. - 447 с.
2. Иванова Г.С. Объектно-ориентированное программирование : учебник для вузов / Иванова Г.С., Ничушкина Т.Н. ; под ред. Г.С.Ивановой. - Москва : Изд-во МГТУ им. Н.Э.Баумана, 2014. - 455 с.
3. Павловская, Т.А. C++. Объектно-ориентированное программирование : практикум : учебное пособие для вузов / Павловская Т.А., Щупак Ю.А. - СПб. : Питер, 2006. - 265 с.
4. Лафоре, Р. Объектно-ориентированное программирование в C++ / Лафоре Р. - 4-е изд. - СПб. : Питер, 2007. - 928 с.

в) ресурсы сети «Интернет»

- 1 Интерактивная доска.
- 2 <http://www.ois.org.ua/spravka/mat/index.htm> - электронная библиотека по математике.
- 3 <http://eqworld.ipmnet.ru/ru/library.htm>- учебно-образовательная физико-математическая библиотека.
- 4 <http://www.exponenta.ru/>- образовательный математический сайт.

**Приложение № 1 к рабочей программе дисциплины
«Объектно-ориентированное программирование»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

1. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости

Образцы заданий для лабораторных работ:

1. Разработайте объектную модель системы «Абитуриент», предполагая, что основными ее классами являются следующие:
 - «Абитуриент», свойствами которого являются ФИО, место жительства, номер школы, паспортные данные, выбранная специальность, результаты ЕГЭ и т.д;
 - «Результат вступительных испытаний», определяющий баллы, которые получил конкретный абитуриент при сдаче экзамена по одному предмету (ЕГЭ или вступительные экзамены);
 - «Направление подготовки»: код, название, план приема, проходной балл и т.д.
 - «Институт»: код, название института, список направлений подготовки;
 - «Приказ о зачислении», характеризующийся номером, датой и списком абитуриентов, зачисленных в студенты по различным направлениям подготовки;
 - «Студент», основная информация о котором совпадает с атрибутами класса «Абитуриент», а также имеются новые свойства, характерные для других приложений.
2. Пусть имеется описание класса комплексных чисел:

```
class comple  
x { doublere  
, im; };
```

Напишите конструкторы, позволяющие создавать объект класса `complex` а) по двум заданным параметрам;
б) по заданному значению `re` (при этом считать `im=0`);
в) без параметров.
Можно ли написать один конструктор, удовлетворяющий условиям а) – в) одновременно?
3. Приведите описание класса «обыкновенная дробь», члены-данные класса – числитель и знаменатель. При создании объектов этого класса возможно задание значений числителя и знаменателя или только числителя (знаменатель в этом случае считается равным 1). При создании дроби должно проводиться сокращение, то есть дроби $\frac{1}{2}$ и $\frac{6}{12}$ должны стать одинаковыми. Нужно ли для создания копий объектов такого класса явно определять конструктор копирования?
4. Разработайте класс для реализации методов решения квадратного уравнения $ax^2+bx+c=0$. Уравнение задается набором коэффициентов (от 1 до 3). Если при

создании указывается иное количество коэффициентов, то квадратное уравнение определить нельзя, поэтому выдается предупреждение об ошибке. В классе должны быть предусмотрены средства для решения уравнений, в которых $a=0$, $b=0$ или $c=0$. Тогда уравнение может стать линейным ($0*x^2+5x+2=0$), обратиться в тождество ($0=0$) или стать неразрешимым (например, $6=0$).

5. Опишите реализацию абстрактного типа данных. Если для него требуются вспомогательные абстрактные типы данных, приведите их описание полностью:

- Почтовый адрес. Для объектов класса должны быть предусмотрены функции изменения индекса, города, улицы, номера дома, корпуса, квартиры, вывод адреса на экран.
- Банковский счет. Необходимые данные: дата создания счета, сумма денег, которая на нем хранится, владелец счета (объект класса `person`, у которого есть фамилия и имя), информация о последних 10 операциях, проведенных со счетом. Операция, проводимая со счетом—объект, содержащий дату операции, вид операции (добавить/снять деньги) и сумму операции. У класса банковский счет должны быть методы: пополнить счет, снять деньги (если указана недопустимая сумма—должно печататься сообщение об ошибке, сумма на счете при этом не меняется), распечатать информацию о последних 10 операциях, распечатать доступную сумму денег на счете.
- Зоопарк. Необходимые данные: количество зверей, массив, содержащий информацию о животных (имя, номер клетки, название любимой еды), часы работы зоопарка адрес зоопарка, фамилия сторожа. В классе нужно определить методы, позволяющие «накормить» зверя (изменить значение соответствующего поля), поменять сторожа зоопарка (записать другую фамилию сторожа в соответствующее поле).

6. Пусть класс `Door` описан следующим образом:

```
classDoor{  
    inth  
    eight; intwidth;  
};
```

Приведите примеры операций, которые:

- a) могут быть перегружены только с помощью функции – члена класса;
- б) могут быть перегружены только с помощью функции, не являющейся членом класса;
- с) могут быть перегружены как функцией – членом класса, так и функцией, не являющейся членом класса;
- d) не могут быть перегружены.

Напишите соответствующие прототипы перегруженных операций для класса `Door`.

- 7. Опишите иерархию классов для хранения информации о сотрудниках, студентах и аспирантах университета. В качестве базового используйте класс `Person`.
- 8. Определите иерархию классов для описания сессии в университете. Базовый класс—`event`, содержит информацию о дате события, фамилию действующего лица и виртуальную функцию `print_res`, печатающую информацию о событии; классынаследники—`test` (зачет), `exam` (экзамен). В них должны быть определены дополнительные характеристики событий и метод `print_res`. В функции `main` определите сессию как массив указателей на события, проинициализируйте элементы массива и распечатайте информацию об экзаменах и зачетах.

9. Разработайте обобщенный класс Set для хранения множества элементов некоторого типа. В методе Main(string [] args) продемонстрируйте применение обобщенного класса Set для хранения целых чисел, вещественных чисел и строк.

Образцы вопросов для защиты лабораторных работ:

1. Приведите определение понятия «объект».
2. Чем характеризуется каждый объект?
3. Что такое «состояние объекта»?
4. Что такое «поведение объекта»?
5. Приведите определение понятия «класс»?
6. Что такое «экземпляр класса»?
7. Что понимают под «ассоциацией объектов»?
8. Что такое «агрегирование» объектов?
9. Что такое «композиция» объектов?
10. Перечислите известные вам принципы объектно-ориентированного программирования.
11. В чем заключается принцип абстрагирования?
12. Что предполагает принцип инкапсуляции?
13. В чем заключается принцип наследования? 14. Что предполагает принцип полиморфизма?
15. Перечислите основные возможности платформы .NET Framework.
16. Что такое общезыковая среда выполнения программ (CLR)?
17. Какие языки программирования поддерживает платформа .NET Framework?
18. Приведите синтаксис описания класса в C#.
19. Приведите синтаксис обращения к отдельному элементу класса.
20. Для каких целей используются поля класса?
21. Какой синтаксис используется для определения поля класса?
22. Какие модификаторы регулируют права доступа к полям класса?
23. Каким модификатором отмечаются статические поля?

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Вопросы к зачету:

1. Основные элементы объектной модели.
2. Отношения между объектами и классами.
3. Абстрагирование.
4. Инкапсуляция.
5. Наследование.
6. Полиморфизм.
7. Особенности платформы .NET.

8. Классы с C#, поля класса.
9. Указание области видимости: public; private; protected; internal.
10. Статические поля.
11. Константы.
12. Методы класса: синтаксис; тип возвращаемого значения; список формальных параметров.
13. Модификаторы доступа к методам: public, private, protected, internal, static.
14. Методы с переменным числом параметров.
15. Определение выходных параметров метода.
16. Вызов метода.
17. Рекурсивные методы.
18. Конструкторы.
19. Деструкторы.
20. Перегрузка методов.
21. Перегрузка операторов.
22. Свойства класса.
23. Индексаторы класса.
24. Функциональные типы в C#, делегаты.
25. Функциональные типы в C#, события.
26. Наследование.
27. Виртуальные функции.
28. Абстрактные классы.
29. Создание иерархии исключений.
30. Обобщения, основные понятия.
31. Уточнения, используемые в обобщениях.
32. Обобщенные интерфейсы.
33. Обобщенные методы.
34. Обобщенные делегаты.

Вопросы экзамену:

1. Понятие алгоритма. Свойства алгоритмов. Способы описания алгоритмов.
2. Базовые алгоритмические конструкции. Язык программирования.
3. Типы данных: понятие, простые типы данных, примеры типов, совместимость и преобразование типов. Переменные, операции и выражения.
4. Операторы. Структура программы. Организация ввода и вывода.
5. Операторы ветвления. Операторы циклов.
6. Указатели. Динамические переменные. Массивы.
7. Строковый тип данных. Структуры. Объединения.
8. Функции. Файлы.

9. Технологии разработки приложений с графическим интерфейсом. Основные свойства форм.
10. Добавление форм. Взаимодействие между формами. События форм.
11. Стандартные элементы управления, их основные свойства, основные методы, основные события.
12. Элементы управления для ввода/вывода. Переключатели.
13. Счетчик. Списки. Стандартные диалоги. Меню и панели инструментов.
14. Линейные списки. Стеки. Очереди. Деки. Основные операции, способы представления.
15. Типы данных, структуры данных, абстрактные типы данных. Деревья. Основные понятия.
16. Создание, обход, представление деревьев. Пример использования деревьев: код Хаффмана.
17. Графы. Основные понятия. Способы представления графов. Путь с наименьшим количеством дуг в графе.
18. Волновой обход графов. Путь кратчайшей длины в графе. Алгоритм Дейкстры.
19. Кратчайший путь между парами вершин графа. Алгоритм Флойда.
20. Обход графов. Метод поиска в глубину. Глубинный остовный лес графов.
21. Циклы в графах. Алгоритм нахождения циклов в графе.
22. Сильная связность. Нахождение компонент сильной связности в графе. Хроматическое число графов. Нахождение хроматического числа графов.
23. Внешняя сортировка. Сортировка слиянием. Метод полного перебора. Перебор циклами, рекурсивный перебор.
24. Метод полного перебора. Полный р-ичный перебор. Динамическое программирование. Принцип оптимальности Беллмана. "Жадные" алгоритмы. Метод ветвей и границ.
25. Задача сортировки. Метод прямого выбора. Метод прямого включения. Метод прямого обмена.
26. Улучшенная сортировка. Метод Шелла. Шейкерная сортировка.
27. Улучшенная сортировка. Пирамидальная сортировка. Быстрая сортировка. "Карманная" сортировка.
28. Задача поиска. Последовательный поиск. Поиск в упорядоченной таблице. Бинарный поиск. Поиск по бинарному дереву. Дерево поиска.
29. AVL-деревья. Хеш-таблицы. Открытое хеширование. Хеш-таблицы. Прямое хеширование. Методы разрешения коллизий.
30. Эффективность алгоритмов. NP-полные и трудно решаемые задачи.
31. 1С:Предприятие. Приложения «1С» и их назначение, типологизация. Платформа «1С:Предприятие» как средство разработки бизнес-приложений.
32. 1С:Предприятие. Конфигурация и прикладное решение. Режимы работы системы «1С:Предприятие». Создание новой информационной базы в системе «1С:Предприятие».

33. 1С:Предприятие. Возможности режима «Конфигуратор». Возможности режима «1С:Предприятие». Работа с подсистемами.
34. 1С:Предприятие. Справочники. Документы. Механизм основных форм.
35. 1С:Предприятие. Модули. Форма как программный объект. Обработчики событий в модуле формы.
36. 1С:Предприятие. Анализ кода с помощью синтакс-помощника. Работа с объектами. Сервер и клиенты.
37. 1С:Предприятие. Директивы компиляции. Исполнение кода на клиенте и на сервере. Регистры накопления.
38. 1С:Предприятие. Работа с отчетами. Макеты. Редактирование макетов и форм.
39. 1С:Предприятие. Периодический регистр сведений. Перечисления. Отчеты.
40. 1С:Предприятие. План видов характеристик. Бухгалтерский учет. План видов расчета, регистр расчета.
41. 1С:Предприятие. Организация поиска. Выполнение заданий по расписанию.
Редактирование движений в форме документа.
42. 1С:Предприятие. Список пользователей и их роли. Настройка командного интерфейса.
43. 1С:Предприятие. Обмен данными. Функциональные опции. Типовые приемы разработки.
44. 1С:Предприятие. Приемы разработки форм. Приемы редактирования форм.

Уровни оценки компетенций следующие: базовый – 55-69 баллов, повышенный – 70-100 баллов.

Преподаватель проводит систематический контроль знаний студентов, ориентируясь на перечень вопросов для проведения зачета/экзамена.

Критерии оценки лабораторных работ / практических занятий / самостоятельной работы студента (от 0 до 10 баллов):

- 9-10 баллов** выставляется студенту, если работа выполнена самостоятельно и полностью верно; представлен отчет, содержащий результаты выполнения заданий работы и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- 7-8 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- 5-6 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий лабораторной работы и ответы на контрольные вопросы; студент испытывает затруднения при проведении анализа результатов, полученных в ходе выполнения лабораторной работы, и формулировке выводов.

- **3-4 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, однако оформил отчет по результатам работы.
- **1-2 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, не оформил отчет по результатам работы.
- **0 баллов** выставляется студенту, если студент не справился с заданием, неверно ответил на представленные вопросы.

Ответ на зачете/экзамене оценивается исходя из 40 баллов (максимум). Билет содержит теоретический вопрос и практическое задание, преподаватель может задавать дополнительные вопросы. Полный ответ на основной вопрос оценивается максимум в 20 баллов, предполагает свободное изложение (не чтение) всего необходимого материала, ответы студента на уточняющие вопросы, если они есть. Правильный ответ на дополнительный вопрос оценивается максимум в 5 баллов. Правильное выполнение практического задания оценивается в 20 баллов.

Шкала оценивания компетенций:

Оценка в 100-балльной шкале	Оценка в 5-ти балльной шкале	Уровень сформированности компетенций
0-54 баллов	неудовлетворительно (не зачтено)	недостаточный
55-69 баллов	удовлетворительно (зачтено)	базовый
70-85 баллов	хорошо (зачтено)	повышенный
86-100 баллов	отлично (зачтено)	

Приложение № 2 к рабочей программе дисциплины «Объектно-ориентированное программирование»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Практикум на ЭВМ по объектно-ориентированному программированию» в первой половине курса являются лекции, так как изучение языка программирования требует знания определенного количества теоретического материала, причем в достаточно большом объеме. Во второй половине курса большинство занятий представляют из себя лекции- беседы, на которых обсуждаются технологии программирования и конкретные примеры их реализации, а так же мастер-классы, где преподаватель демонстрирует процесс разработки программного обеспечения, основные его аспекты и обсуждает возможности рассматриваемых инструментов разработки.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. В основном такими задачами являются лабораторные работы различного объема, а так же небольшие задачи связанные с исправлением ошибок и доработкой программ.

Для усвоения материала необходимо в течение всего курса выполнять задания для самостоятельной работы.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков разработки программ, в течение обучения проводятся мероприятия текущей аттестации в виде письменных контрольных работ. Также проводятся консультации (при необходимости) по разбору заданий, которые вызвали затруднения.

В конце изучения дисциплины студенты сдают зачет. Зачет выставляется по итогам работы, которая включает в себя выполнение лабораторных и контрольных работ.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Практикум на ЭВМ по объектно-ориентированному программированию» самостоятельно студенту довольно сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет по итогам изучения дисциплины студенту практически невозможно

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).
2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:
 1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт

меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/orac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «Книгообеспеченность» (http://www.lib.uniyar.ac.ru/orac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.