

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Новоуральский технологический институт –
филиал федерального государственного автономного образовательного учреждения высшего об-
разования «Национальный исследовательский ядерный университет «МИФИ»

УТВЕРЖДЕНА
Методическим советом НТИ НИЯУ МИФИ
Протокол № 4 от 30.08.2021 г.

Рабочая программа учебной дисциплины
"Объектно-ориентированное программирование"

Направление подготовки (специальность)	<u>09.03.01 – Информатика и вычис- лительная техника</u>
Профиль подготовки (специализация)	<u>Автоматизированные системы обработки информации и управ- ления</u>
Квалификация (степень) выпускника	<i>Бакалавр</i>
Форма обучения	очная

г. Новоуральск, 2021

Семестр	5
Трудоемкость, ЗЕТ	2
Трудоемкость, ч.	72
Аудиторные занятия, в т.ч.:	54
- лекции	18
- практические занятия	18
- лабораторные занятия	18
Самостоятельная работа	18
Форма промежуточного контроля	зачет

Семестр	6
Трудоемкость, ЗЕТ	3
Трудоемкость, ч.	108
Аудиторные занятия, в т.ч.:	48
- лекции	16
- лабораторные занятия	16
Контроль самостоятельной работы	16
Самостоятельная работа	60
Форма промежуточного контроля	экзамен

Учебную программу составил старший преподаватель кафедры автоматизации управления Кузин-Куликов Сергей Николаевич

Содержание

1. Цели освоения учебной дисциплины	4
2. Место учебной дисциплины в структуре образовательной программы	4
3. Формируемые компетенции и планируемые результаты обучения	4
4. Воспитательный потенциал дисциплины	5
5. Структура и содержание учебной дисциплины.....	6
6. Образовательные технологии.....	10
7. Фонд оценочных средств.....	11
8 Учебно-методическое и информационное обеспечение учебной дисциплины	23
9 Материально-техническое обеспечение учебной дисциплины	23
10 Учебно-методические рекомендации для студентов и преподавателей.....	24

Рабочая программа составлена в соответствии с Образовательным стандартом высшего образования Национального исследовательского ядерного университета «МИФИ» по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (квалификация (степень) «бакалавр»), и рабочим учебным планом (РУП) по направлению подготовки 09.03.01 «Информатика и вычислительная техника» (профиль – «Автоматизированные системы обработки информации и управления»).

1. Цели освоения учебной дисциплины

Глобальная цель курса – подготовка студентов к проектно-технологической деятельности в области создания компонентов программных комплексов и баз данных, автоматизации технологических процессов с использованием современных инструментальных средств и технологий программирования.

2. Место учебной дисциплины в структуре образовательной программы

В соответствии с кредитно-модульной системой подготовки бакалавров по направлению 09.03.01 «Информатика и вычислительная техника» данная учебная дисциплина входит в профессиональную часть.

Дисциплина «Объектно-ориентированное программирование» входит в число дисциплин формирования профессиональных компетенций выпускника и служит опорой для подготовки к его итоговой государственной аттестации.

Дисциплина знакомит с принципами разработки и сопровождения программного обеспечения. Предшествующий уровень образования обучаемого – среднее (полное) общее образование.

3. Формируемые компетенции и планируемые результаты обучения

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-10.1 Способен разрабатывать и тестировать прототип информационной системы в соответствии с требованиями технического задания	<p>З-ПК-10.1 Знать: языки программирования и работы с базами данных, основы современных операционных систем, основы современных систем управления базами данных, современные объектно-ориентированные языки программирования, устройство и функционирование современных ИС.</p> <p>У-ПК-10.1 Уметь: кодировать на языках программирования, тестировать результаты собственной работы.</p> <p>В-ПК-10.1 Владеть: методами разработки кода прототипа ИС и баз данных прототипа в соответствии с трудовым заданием, проведения тестирования.</p>
ПК-10.3 Способен создавать программный код в соответствии с техническим заданием (готовыми спецификациями)	<p>З-ПК-10.3 Знать: основы программирования, современные объектно-ориентированные языки программирования, современные структурные языки программирования.</p> <p>У-ПК-10.3 Уметь: кодировать на языках программирования, тестировать результаты кодирования.</p> <p>В-ПК-10.3 Владеть: методами разработки кода ИС и баз данных ИС, верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС, устранения обнаруженных несоответствий.</p>

4. Воспитательный потенциал дисциплины

Цели и задачи воспитания, воспитательный потенциал дисциплин:

Направления/цели воспитания	Задачи воспитания (код)	Воспитательный потенциал дисциплин
Интеллектуальное воспитание	- формирование культуры умственного труда (B11)	Использование воспитательного потенциала дисциплины для формирования культуры умственного труда посредством вовлечения студентов в учебные исследовательские задания, курсовые работы и др.
Профессиональное воспитание	- формирование чувства личной ответственности за научно-технологическое развитие России, за результаты исследований и их последствия (B17)	Использование воспитательного потенциала дисциплины для формирования чувства личной ответственности за достижение лидерства России в ведущих научно-технических секторах и фундаментальных исследованиях, обеспечивающих ее экономическое развитие и технологического лидерства России.
	- формирование ответственности за профессиональный выбор, ответственности за профессиональное развитие и профессиональные решения (B18)	Использование воспитательного потенциала дисциплины для формирования чувства личной ответственности за свое профессиональное развитие посредством выбора студентами индивидуальных образовательных траекторий, организации системы общения между всеми участниками образовательного процесса, в том числе с использованием новых информационных технологий.
	- формирование навыков коммуникации, командной работы и лидерства (B20)	Использование воспитательного потенциала дисциплины для развития навыков коммуникации, командной работы и лидерства.
	- формирование способности и стремления следовать в профессии нормам поведения, обеспечивающим нравственный характер трудовой деятельности и неслужебного поведения (B21)	Использование воспитательного потенциала дисциплины для развития стремления следовать в профессиональной деятельности нормам поведения, обеспечивающим нравственный характер трудовой деятельности и неслужебного поведения.
	- формирование творческого инженерного мышления, навыков организации коллективной проектной деятельности (B22)	Использование воспитательного потенциала дисциплины для развития - формирования производственного коллективизма в ходе совместного решения как модельных, так и практических задач, а также путем под-

		крепление рационально- технологических навыков взаимодействия в проектной деятельности эмоциональным эффектом успешного взаимодействия.
	- формирование навыков цифровой гигиены (В30)	Использование воспитательного потенциала дисциплины для формирования культуры написания и оформления программ, формирование навыков цифровой гигиены, формирование ответственности за обеспечение кибербезопасности, формирование профессионально значимых установок: не работать на заказчика, чьи цели противоречат гуманистическим ценностям.
	- формирование ответственности за обеспечение кибербезопасности (В31)	
	- формирование профессионально значимых установок (В32)	

5. Структура и содержание учебной дисциплины

5.1 Структура учебной дисциплины

Семестр – 5

№	Название темы/раздела учебной дисциплины	Виды учебных занятий, и их трудоемкость (в часах)					Форма контроля
		Лекции	Практические занятия	Лабораторные занятия	Курсовые работы/проекты	Самостоятельная работа	
1	2	4	5	6	7	8	9
1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	4	4	4	-	4	ЗЛР1
2.	<i>Основные понятия языка программирования C#.</i>	6	6	6	-	6	ЗЛР2
3.	<i>Операции и операторы.</i>	4	4	4	-	4	ЗЛР3
4.	<i>Делегаты и события классов.</i>	4	4	4	-	4	ЗЛР4
Итого:		18	18	18	-	18	
5.	Зачет						ТЗ

Семестр – 6

№ п/п	Название темы/раздела учебной дисциплины	Виды учебных занятий, и их трудоем- кость (в часах)					Форма контроля
		Лекции	Практические заня- тия	Лабораторные заня- тия	Курсовые рабо- ты/проекты	Самостоятельная работа	
1	2	4	5	6	7	8	10
1.	<i>Основные понятия языка программирования C++.</i>	4	-	4	-	15	ЗЛР4
2.	<i>Операции и операторы.</i>	4	-	4	-	15	
3.	<i>События классов.</i>	4	-	4	-	15	
4.	<i>Создание форм и работа с ними.</i>	4	-	4	-	15	
Итого:		16	-	16	16	60	
5.	Экзамен						ТЗ

Сокращение наименований форм текущего контроля и аттестации разделов:
ЗЛР – защита лабораторной работы, ТЗ – тестовое задание.

5.2 Содержание учебной дисциплины

5.2.1 Лекции

/*	Тема/раздел учебной дисциплины	Содержание
1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	Лекция 1. Качество ПО. Лекция 2. Критерии объектной ориентации.
2.	<i>Основные понятия языка программирования C#.</i>	Лекция 3. Модульность. Лекция 4. Подходы к повторному использованию. Лекция 5. Статические структуры: классы. Классы, а не объекты - предмет обсуж-

		<i>дения.</i>
3.	<i>Операции и операторы.</i>	Лекция 6. <i>Динамические структуры: объекты.</i> Лекция 7. <i>Управление памятью. Что происходит с объектами.</i>
4.	<i>Делегаты и события классов.</i>	Лекция 8. <i>Делегат - ссылки на методы с определенным списком параметров и возвращаемым типом.</i> Лекция 9. <i>Обработка и вызов событий.</i>
5.	<i>Основные понятия языка программирования C#.</i>	Лекция 10. <i>Начальные сведения о языке.</i> Лекция 11. <i>Имена, переменные и константы.</i>
6.	<i>Операции и операторы.</i>	Лекция 12. <i>Операции и выражения.</i> Лекция 13. <i>Операторы.</i> Лекция 14. <i>Функции.</i>
7.	<i>События классов.</i>	Лекция 15. <i>Классы и объекты.</i> Лекция 16. <i>Производные классы, наследование.</i>
8.	<i>Создание форм и работа с ними.</i>	Лекция 17. <i>Создание форм и работа с ними.</i> Лекция 18. <i>Создание форм и работа с ними.</i>

5.2.2 Лабораторные работы

№	Тема/раздел учебной дисциплины	Содержание
1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	Лабораторная работа 1.
2.	<i>Основные понятия языка программирования C#.</i>	Лабораторная работа 2.
3.	<i>Операции и операторы.</i>	Лабораторная работа 3.
4.	<i>Делегаты и события классов.</i>	Лабораторная работа 4.
5.	<i>Основные понятия языка программирования C++.</i>	Лабораторная работа 5. <i>Переменные и базовые типы данных языка C++.</i>
6.	<i>Операции и операторы.</i>	Лабораторная работа 6. <i>Организация циклов в языке C++. Принятие решений. Условные операторы в языке C++.</i>
7.	<i>События классов.</i>	Лабораторная работа 7. <i>Обработка событий классов</i>

8.	<i>Создание форм и работа с ними.</i>	Лабораторная работа 8. Программы на языке C++ при использовании статически подключаемой библиотеки
----	---------------------------------------	--

5.2.3 Практические занятия

№	Тема/раздел учебной дисциплины	Содержание
1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	Практическое занятие 1. Компьютерная 2D-графика в Microsoft XNA Game Studio 3.0.
2.	<i>Основные понятия языка программирования C#.</i>	Практическое занятие 2. Компьютерная графика 3D в XNA.
3.	<i>Операции и операторы.</i>	Практическое занятие 3. Windows Forms и XNA 3.0.
4.	<i>Делегаты и события классов.</i>	Практическое занятие 4. Создание приложений WPF.

5.2.4 Курсовая работа

Примерные темы курсовых работ:

1. Операции с комплексными числами
2. Отображение динамических структур данных в графическом режиме
3. Параметризованные коллекции
4. Планирование семейного бюджета
5. Погодный информер
6. Построение минимального остового дерева по алгоритму Краскала Visual C#
7. Проверка графа на двудольность
8. Программа для построения диаграмм
9. Разработка приложений в среде Visual Studio на языке C#
10. Разработка редактора интерьеров
11. Реализация графических часов на языке C#
12. Создание текстового редактора на языке C#

Студент может разработать собственную тему курсовой работы по согласованию с преподавателем.

5.2.5 Самостоятельная работа

Самостоятельная работа студента по учебной дисциплине регламентируется «Положением об организации самостоятельной работы студентов в НТИ НИЯУ МИФИ».

№	Тема/раздел учебной дисциплины	Вид самостоятельной работы и ее содержание ¹
1	2	3

¹ В соответствии с «Положением об организации самостоятельной работы студентов в НТИ НИЯУ МИФИ»

1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	Подготовка к выполнению лабораторной работы
2.	<i>Основные понятия языка программирования C#.</i>	Подготовка к выполнению лабораторной работы
3.	<i>Операции и операторы.</i>	Подготовка к выполнению лабораторной работы.
4.	<i>Делегаты и события классов.</i>	Подготовка к выполнению лабораторной работы.
5.	<i>Основные понятия языка программирования C++.</i>	Подготовка к выполнению лабораторной работы. Выполнение курсовой работы на выбранную тему.
6.	<i>Операции и операторы.</i>	Подготовка к выполнению лабораторной работы. Выполнение курсовой работы на выбранную тему.
7.	<i>События классов.</i>	Подготовка к выполнению лабораторной работы. Выполнение курсовой работы на выбранную тему.
8.	<i>Создание форм и работа с ними.</i>	Подготовка к выполнению лабораторной работы. Выполнение курсовой работы на выбранную тему.

6. Образовательные технологии

Рекомендации для преподавателя по использованию информационно-образовательных технологий содержатся в «Положении об организационных формах и технологиях образовательного процесса в НТИ НИЯУ МИФИ».

№	Тема/раздел учебной дисциплины	Форма занятия ²	Используемые технологии, включая перечень программного обеспечения и информационные справочные системы (при наличии)
1.	<i>Теоретические основы объектно-ориентированного программирования.</i>	Лекции, Лабораторные работы, Практические занятия.	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, СУБД MySQL, Devart dbForge Studio for MySQL, C#.
2.	<i>Основные понятия языка программирования C#.</i>	Лекции, Лабораторные работы, Практические занятия.	Компьютерный класс, банк лекций-презентаций, СУБД MySQL, Devart dbForge Studio for MySQL, C#.

² В соответствии с «Положением об организационных формах и технологиях образовательного процесса в НТИ НИЯУ МИФИ»

3.	<i>Операции и операторы.</i>	Лекции, Лабораторные работы, Практические занятия.	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, СУБД MySQL, Devart dbForge Studio for MySQL, C#.
4.	<i>Делегаты и события классов.</i>	Лекции, Лабораторные работы, Практические занятия.	Компьютерный класс, банк лекций-презентаций, СУБД MySQL, Devart dbForge Studio for MySQL, C#.
5.	<i>Основные понятия языка программирования C++.</i>	Лекции, Лабораторные работы	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, C++.
6.	<i>Операции и операторы.</i>	Лекции, Лабораторные работы	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, C++..
7.	<i>События классов.</i>	Лекции, Лабораторные работы	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, C++..
8.	<i>Создание форм и работа с ними.</i>	Лекции, Лабораторные работы	Мультимедиа лекции, проблемная технология, банк лекций-презентаций, C++..

7. Фонд оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий рубежного и промежуточного контроля по дисциплине.

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении контрольных мероприятий. Полученные баллы переводятся в 5-балльную систему по следующей шкале:

Оценка по 5 балльной шкале	Зачет	Сумма баллов по дисциплине	Оценка (ECTS)	Градация
5 (отлично)	Зачтено	90-100	A	Отлично
4 (хорошо)		85-89	B	Очень хорошо
		75-84	C	Хорошо
		70-74	D	Удовлетворительно
3 (удовлетворительно)		65-69	E	Посредственно
	60-64			
2 (неудовлетворительно)	Не зачтено	Ниже 60	F	Неудовлетворительно

7.1 Материалы, необходимые для оценки результатов обучения

Зачетное тестовое задание

1. Сколько элементов содержит массив *cube*?

cube : array [1..2,1..2,1..2,1..2] of Boolean ;

Ваш ответ: _____

2. Что выведет следующая программа?

var

i: Integer;

Mas: array of Integer;

begin

SetLength(*Mas*, 10);

for *i*:= 0 to Length(*Mas*)-5 do

SetLength(*Mas*, Length(*Mas*)-1);

Writeln(Length(*Mas*));

SetLength(*Mas*, 10);

i:=0;

while *i*<Length(*Mas*)-5 do

begin

SetLength(*Mas*, Length(*Mas*)-1);

inc(*i*);

end;

Writeln(Length(*Mas*));

end.

a) 4

4

b) 4

7

c) 5

5

d) 7

4

3. Что выведет следующая программа?

var

arr1 : array [-5..5] of Byte;

begin

Writeln(Length(*arr1*));

end.

a) Программа не скомпилируется.

b) 5

c) 10

d) 11

4. Какая процедура возвращает в кучу кусок памяти, зарезервированный за типизированным указателем? Напишите любую. (маленькими строчными буквами, без скобок и аргументов)

Ваш ответ: _____

5. Что хранится в *tempStr[0]*?

var

tempStr : String [20];

begin

```
tempStr := 'I love delphi.';
// <----
end.
```

- a) 14
- b) 'I'
- c) 0
- d) Не определено.

6. Чему будет равна переменная B?

```
var
  B: Boolean;
  P: PChar;
  S: string;
begin
  S := 'Stroka';
  P := 'Stroka';
  B := (P = S)
end;
```

- a) True
- b) False
- c) Ошибка компиляции

7. Что выведет следующий код?

```
{SZ1}
type
  TFoo = (foo=255, bar);
begin
  WriteLn(SizeOf(TFoo));
end.
```

- a) 1
- b) 2
- c) 4
- d) Ошибка компиляции
- e) Ошибка времени выполнения

8. Что выведет следующая программа?

```
var
  a : integer;
begin
  a := 1000;
  Writeln(a)
end.
```

- a) 0
- b) 1000
- c) Ошибка компиляции в строчке 'Writeln(a)' из-за отсутствия ';'

9. Что произойдет при выполнении следующей программы?

```
var
  a : Integer;
begin
  a := 8;
  repeat
```

```
a := a+2
until (a=8);
Writeln(a);
end.
```

- a) Программа выведет 10.*
- b) Программа не скомпилируется.*
- c) Программа никогда не закончит работу.*
- d) Возникнет переполнение, но программа завершится.*

10. Что выведет следующая программа?

```
type
A = class
  procedure w;
end;
B = class
  procedure w;
end;
AB = class(A,B);
procedure A.w;
begin
  Writeln('w from A');
end;
```

```
procedure B.w;
begin
  Writeln('w from B');
end;
```

```
var
  abba : AB;
begin
  abba.w;
end.
```

- a) w from A*
 - b) w from B*
 - c) w from AB*
 - d) w from A*
- w from B*
- e) Программа не скомпилируется.*

11. Что необходимо написать вместо ??????, чтобы программа вывела "100"?

```
var
  ar1 : array of Integer;
begin
  ??????
  Writeln(Length(ar1));
end.
```

- a) ar1[0] := 100;*
- b) SetLength(ar1,100);*
- c) ar1 := 100;*
- d) ar1(100);*

12. Что выведет следующий код?

```
var
  i                               :Byte;
  b                               :Boolean;
begin
  i                               :=          42;
  b                               :=          Boolean(i);
  WriteLn(b = true, ' ', b = false);
end.
```

- a) TRUE, TRUE
- b) FALSE, TRUE
- c) FALSE, FALSE
- d) TRUE, FALSE

13. Что выведет следующая программа?

```
var
  a,b : Integer;
begin
  a := 8;
  b := 4;
  if (a mod b = 0) xor (b mod 2 = 0) then
    writeln('!')
  else
    Writeln('?');
end.
```

- a) !
- b) ?
- c) ?
- !
- d) Программа не скомпилируется.

14. Что выведет следующая программа?

```
уре
  A = class
    procedure w;
  end;
  B = class(A);
  procedure A.w;
  begin
    Writeln('w from A');
  end;

var
  abba : B;
begin
  abba.w;
end.
```

- a) Программа ничего не выведет.
- b) Программа не скомпилируется.
- c) w from A
- d) w from B

```

15. Имеется:
unit TestUnit;

interface

type

  TLeftClass = class (TObject)
    private
      class procedure SayIt;
    end;

  TRightClass = class (TObject)
    procedure DoIt;
    end;

implementation

class procedure TLeftClass.SayIt;
begin
  Writeln('It works !');
end;

procedure TRightClass.DoIt;
begin
  TLeftClass.SayIt;
end;

end.

```

Каким будет результат выполнения программы:

```

program Project1;

{$APPTYPE CONSOLE}

uses
  Classes,
  TestUnit in 'TestUnit.pas';

var
  TestClass : TRightClass;
begin
  TestClass := TRightClass.Create;
  try
    TestClass.DoIt;
  finally
    TestClass.Free;
  end;
end.

a) It works!

```


- b) Возникнет исключение *invalid pointer operation*.
- c) Программа не скомпилируется.

16. Что выведет следующая программа?

```
{$J+}
const
  a : Integer = Ord('5') - Ord('5');
var
  i : Integer;
begin
  for i := 1 to 2 do
    a := a * 2 + 1;
  writeln(a);
end.
```

- a) 5
- b) 0
- c) 3
- d) Программа не скомпилируется

17. Что выведет следующая программа?

```
label
  m1,m2;
var
  a : Integer;
begin
  a := 10;
  for a := 1 to 11 do
    if (a=5) then goto m1;
  m2:
  if (a=6) then
    Writeln('!')
  else
    Writeln('?');
  m1:
  a := a+1;
  if (a < 10) then goto m2;
end.
```

- a) 11
- b) !
- c) ?
- d) !
- ?
- ?
- ?
- e) !
- !
- !
- !

18. *High Cohesion* (сильное сцепление) - это ОО принцип, наиболее ассоциирующийся с ...

- a) сокрытием реализации
- b) тем, что класс спланирован с единственным и конкретным назначением

- c) разрешением одному объекту быть видимым как разные типы
- d) тем, как много классы знают про другие только через их API

19. Какой принцип ООП необходимо использовать, чтобы заменить конструкции *if-then-else* в данном фрагменте кода:

```

if      (animal.IsCat())      {      /*      код      */      }
else    if      (animal.IsDog())  {      /*      код      */      }
else    if      (animal.IsKoala()) {      /*      код      */      }
.
else if (animal.isMouse()) { /* код */ }

```

- a) Полиморфизм
- b) Агрегация
- c) Композиция
- d) Инкапсуляция
- e) Персистентность

20. Выберите наиболее подходящее определение Класса.

- a) Тип, содержащий набор функций
- b) Тип, который отображает состояние некоторого объекта
- c) Тип, описывающий поведение некоторой сущности
- d) Тип, описывающий характеристики и поведение объекта

21. Наследование и композиция взаимоисключающие понятия. То есть при создании иерархии объектов используется либо наследование, либо композиция.

- a) Да
- b) Нет

22. Почему в некоторых языках программирования отказываются от поддержки множественного наследования (имеется в виду наследование реализации)?

- a) Поддержка множественное наследования ведет к большим потерям производительности, так как для каждого класса необходимо держать сильно-ветвящуюся иерархию его предков
- b) Множественное наследование практически никогда не используется, в отличие от обычного наследования от одного класса
- c) Множественное наследование невозможно реализовать с помощью таблицы виртуальных функций, поэтому требуются другие намного более сложные алгоритмы
- d) Из-за неоднозначности выбора поведения, в случае если суперклассы некоторого класса содержат методы с одинаковыми сигнатурами

23. К проявлению какого элемента ООП можно отнести следующую фразу: "Переминая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

- a) Наследование
- b) Инкапсуляция
- c) Полиморфизм

24. Как называется способность объекта скрывать свои данные и реализацию от других объектов системы?

- a) Полиморфизм

- b) *Инкапсуляция*
- c) *Абстракция*
- d) *Наследование*

25. Драконы умеют летать (как, например, птицы) и ползать (как, например, ящерицы). С точки зрения ООП, примером чего является данная ситуация (выберите наиболее точный вариант)?

- a) *Инкапсуляция*
- b) *Композиция*
- c) *Наследование*
- d) *Множественное наследование*
- e) *Полиморфизм*

26. Какая разница между идентичностью (*identity*) и равенством (*equality*) объектов в ООП?

- a) *Идентичность означает, что у объектов есть общий неабстрактный предок, а равенство - любой общий предок*
- b) *Идентичность означает, что у объектов одинаковые поля, а равенство - что они содержат одинаковые данные*
- c) *Идентичность означает, что объекты являются экземплярами одного и того же класса, а равенство - что они содержат одинаковые данные*
- d) *Идентичность означает, что две ссылки указывают на один и тот же объект, а равенство - что они содержат одинаковые данные*

27. Какой принцип ООП нарушает следующий фрагмент кода:

```
class Counter {  
    public int count;  
    public void increment() { count++; }  
    public int get() { return count; }  
}  
public class Main {  
    public static void main(String[] args) {  
        Counter counter = new Counter();  
        counter.count = 5;  
    }  
}
```

- a) *Инкапсуляция*
- b) *Равенство*
- c) *Композиция*
- d) *Полиморфизм*
- e) *Агрегация*

28. Какому принципу ООП удовлетворяет обоняние собаки если: "Собака чует кошку - она лает, собака чует пищу - она бежит к миске"?

- a) *Наследование*
- b) *Инкапсуляция*
- c) *Полиморфизм*

29. "Черный ящик" как способ повторного использования - это...

- a) *Стиль повторного использования, основанный на наследовании классов*
- b) *Стиль повторного использования, основанный на композиции объектов.*
- c) *Название структурного паттерна проектирования*

d) *Стиль повторного использования, основанный на сильной связанности объектов*

30. *Как в терминах ООП называется объект, который не может быть изменён после создания?*

- a) *Internal object*
- b) *Abstract object*
- c) *Immutable object*
- d) *Sealed object*
- e) *Все варианты*
- f) *Среди вариантов нет правильного ответа*

Ключ

<i>Вопрос</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>	<i>13</i>	<i>14</i>	<i>15</i>
<i>Ответ</i>	<i>16</i>	<i>b</i>	<i>d</i>	<i>dispose, freemem</i>	<i>a</i>	<i>a</i>	<i>b</i>	<i>b</i>	<i>d</i>	<i>e</i>	<i>b</i>	<i>c</i>	<i>b</i>	<i>c</i>	<i>a</i>

<i>16</i>	<i>17</i>	<i>18</i>	<i>19</i>	<i>20</i>	<i>21</i>	<i>22</i>	<i>23</i>	<i>24</i>	<i>25</i>	<i>26</i>	<i>27</i>	<i>28</i>	<i>29</i>	<i>30</i>
<i>c</i>	<i>d</i>	<i>b</i>	<i>a</i>	<i>d</i>	<i>b</i>	<i>d</i>	<i>b</i>	<i>b</i>	<i>d</i>	<i>d</i>	<i>a</i>	<i>c</i>	<i>b</i>	<i>c</i>

Экзаменационное задание

Используя среду программирования C# решить поставленную задачу. Исправить ошибки. Результат работы предъявить преподавателю.

1. Каков результат компиляции и выполнения приведенного ниже кода?

```

01.static void Main(string[] args)
02.{
03.Console.WriteLine(GetSomeResult(10000));
04.}
05.
06.static long GetSomeResult(long someValue)
07.{
08.long value1 = 10 * 1000 * 10000 * someValue;
09.long value2 = 10 * 1000 * 10000 * 100000;
10.return value2 / value1;
11.}

```

2. Какое значение присвоено x, если приведенный ниже код выводит False?

```

1.float x;
2.Console.Write(x == x)

```

3. Почему следующий код выводит False?

```

1.public static void Main()
2.{
3.Test t = new Test();
4.Console.WriteLine(t.Equals(t));
5.}

```

4. Что будет выведено на экран при выполнении приведенного ниже кода?

```

1.static void Main(string[] args)
2.{
3.char a = 'a';
4.int b = 0;
5.Console.WriteLine(true ? a : b);

```

6.}

5. Что будет выведено на экран при выполнении приведенного ниже кода?

1. `NameValueCollection col = new NameValueCollection();`

2. `Console.WriteLine("Элемент test " + col["test"] != null ? "Существует!" : "Не существует!");`

6. Что следует ожидать на экране?

1. `Console.WriteLine("A" + "B" + "C");`

2. `Console.WriteLine('A' + 'B' + 'C');`

7. Циклическая инициализация полей. Что в результате будет выведено на консоли?

1. `public class A { public static int x = B.y + 1; }`

2. `public class B { public static int y = A.x + 1; }`

3.

4. `static void Main(string[] args)`

5. {

6. `Console.WriteLine("A.x = " + A.x);`

7. `Console.WriteLine("B.y = " + B.y);`

8. }

8. Что в результате будет выведено на консоли?

1. `int j = 0;`

2.

3. `for (int i = 0; i < 10; i++)`

4. `j = j++;`

5.

6. `Console.WriteLine(j);`

9. Что будет выведено в результате такого цикла?

1. `int end = int.MaxValue;`

2. `int begin = end - 100;`

3. `int counter = 0;`

4.

5. `for (int i = begin; i <= end; i++)`

6. `counter++;`

7.

8. `Console.WriteLine(counter);`

10. Что будет выведено в результате такого цикла?

1. `float begin = 1000000000;`

2. `int counter = 0;`

3.

4. `for (float i = begin; i < (begin + 10); i++)`

5. `counter++;`

6.

7. `Console.WriteLine(counter);`

11. Какой же метод выберет компилятор?

1. `class A { public void Test(int n) { Console.WriteLine("A"); } }`

2. `class B : A { public void Test(double n) { Console.WriteLine("B"); } }`

3.

4. `static void Main(string[] args)`

```
5.{
6.B b = new B();
7.b.Test(5);
8.}
```

12. Какой же метод выберет компилятор?

```
1.public class Test
2.{
3.public Test(object obj) { Console.WriteLine("object"); }
4.public Test(int[] obj) { Console.WriteLine("int[]"); }
5.}
6.
7.public static void Main() { Test t = new Test(null); }
```

13. Что будет выведено на экран в результате выполнения кода приведенного ниже?

```
1.List<int> list = new List<int>() { 1, 2, 3, 4, 5 };
2.
3.List<int> all = list.FindAll(
4.i => { Console.WriteLine(i); return i < 3; }
5.);
```

14. Какой же метод выберет компилятор?

```
1.List<int> list = new List<int>() { 1, 2, 3 };
2.var x = list.GroupBy(i => { Console.WriteLine(i); return i; });
3.var y = list.ToLookup(i => { Console.WriteLine(i); return i; });
```

15. Что будет выведено на экран в каждом из трех случаев, приведенных ниже:

A)

```
1.try {
2.Console.WriteLine("Hello ");
3.return;
4.}
5.finally { Console.WriteLine("Goodbye "); }
6.Console.WriteLine("world!");
```

Б)

```
1.try {
2.Console.WriteLine("Hello ");
3.Thread.CurrentThread.Abort();
4.}
5.finally { Console.WriteLine("Goodbye "); }
6.Console.WriteLine("world!");
```

В)

```
1.try {
2.Console.WriteLine("Hello ");
3.System.Environment.Exit(0);
4.}
5.finally { Console.WriteLine("Goodbye "); }
6.Console.WriteLine("world!");
```

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

Компетенции по дисциплине «Объектно-ориентированное программирование» формируются последовательно в ходе проведения лекционных и практических занятий, а также в процессе подготовки и выполнения лабораторных работ.

С целью определения уровня овладения компетенциями, закрепленными за дисциплиной, в заданные преподавателем сроки проводится текущий и промежуточный контроль знаний, умений и навыков каждого обучающегося. Все виды текущего контроля осуществляются на практических занятиях.

Фонд оценочных средств сформирован на основе ключевых принципов оценивания:

- валидность - объекты оценки соответствуют поставленным целям обучения;
- надежность - используются единообразные стандарты и критерии для оценивания достижений;
- справедливость - студенты имеют равные возможности добиться успеха;
- эффективность - соответствие результатов деятельности поставленным задачам.

Процедура оценивания компетенций обучающихся основана на принципах единства используемой технологии для всех обучающихся, выполнения условий сопоставимости результатов оценивания.

8 Учебно-методическое и информационное обеспечение учебной дисциплины

8.1 Перечень литературы, необходимой для освоения дисциплины

1. **Финогенов К.Г.** Основы объектно-ориентированного программирования : [лабораторный практикум]. Электронный ресурс. Москва : МИФИ. 2008 – точка доступа – ЭБС НИЯУ МИФИ – mephi.ru
2. **Шереметьев, А.И** Информатика. Ч.1. Элементы программирования на языке Си : [учебное пособие]. Электронный ресурс. Москва : НИЯУ МИФИ. 2011 – точка доступа – ЭБС НИЯУ МИФИ – mephi.ru
3. **Финогенов К.Г.** Программирование в системе Windows с помощью объектно-ориентированный библиотек [лабораторный практикум]. Электронный ресурс. Москва : МИФИ. 2008 – точка доступа – ЭБС НИЯУ МИФИ – mephi.ru

8.2 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

Наименование ресурса	Электронный адрес ресурса
Национальный Открытый Университет «ИНТУИТ»	https://intuit.ru/
Образовательная платформа Юрайт	https://urait.ru/bcode/468952
Сайт Российской национальной библиотеки:	http://www.nlr.ru/
Книжная поисковая система:	http://www.ebdb.ru/
Образовательный портал НИЯУ МИФИ	https://online.mephi.ru/
Научная библиотека НИЯУ МИФИ	http://library.mephi.ru/

9 Материально-техническое обеспечение учебной дисциплины

Учебная дисциплина обеспечена учебно-методической документацией и материалами. Её содержание представлено в локальной сети учебного заведения и находится в режиме свободного доступа для студентов. Доступ студентов для самостоятельной подго-

товки осуществляется через компьютеры библиотеки и компьютерных классов НТИ НИЯУ МИФИ.

Лекционные занятия:

1. банк электронных презентаций/слайдов,
2. аудитория, оснащенная презентационной техникой (проектор, экран, ноутбук).

Лабораторные работы:

1. компьютерный класс, оснащенный персональными компьютерами.
2. специализированное ПО: СУБД My SQL, Devart dbForge Studio for MySQL, C#, СУБД Caché.

10 Учебно-методические рекомендации для студентов и преподавателей

Вид работ	Методические рекомендации
лекции	Слушая лекции, необходимо уяснить цель, которую лектор ставит перед вами. Отмечать наиболее существенную информацию. Сравнивать то, что услышано на лекции, с прочитанным и усвоенным ранее, укладывать новую информацию в собственную, уже имеющуюся, систему знаний. Если на лекции вы не пучили ответа на возникшие вопросы, разрешается в конце лекции задать их лектору.
лабораторные работы	При подготовке к лабораторной работе необходимо прочитать записанную лекцию, обращая внимание на наиболее важные моменты, прочитать рекомендованный материал из учебно-методической литературы. Поскольку некоторые темы лабораторной работы могут быть составной частью курсовой работы, предполагается активная позиция студента в роли администратора базы данных.
самостоятельная работа	Результатом самостоятельной работы должна быть систематизация и структурирование учебного материала по изучаемой теме, включение его в уже имеющуюся у вас систему знаний.
зачет	Следует понимать, что зачёт - это предварительный отчётный этап работы в семестре по данной дисциплине. Как обычно, определённую роль играют не только посещение занятий, но также и то, насколько внимательны и активны вы были на лекциях, при выполнении и защите лабораторные работ.
экзамен	Следует понимать, что экзамен - это заключительный этап работы в семестре по данной дисциплине. Но решающую роль успешная защита самостоятельно выполненной курсовой работы. Всё это проявляется при ответе на вопросы, предоставленные вам для подготовки к экзамену.
курсовая работа	Тема курсовой работы, как правило, выбирается самим студентом при согласовании с преподавателем. Предметная область исследования выбирается реальной.