

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский Нижегородский государственный университет  
им. Н.И. Лобачевского»**

**Арзамасский филиал**

Факультет естественных и математических наук

УТВЕРЖДЕНО  
решением Ученого совета ННГУ  
протокол № 6 от 31.05.2023 г.

## **Рабочая программа дисциплины**

### **Объектно-ориентированное программирование**

*(наименование дисциплины)*

Уровень высшего образования

бакалавриат

*(бакалавриат / магистратура / специалитет)*

Направление подготовки / специальность

09.03.03 Прикладная информатика

*(указывается код и наименование направления подготовки / специальности)*

Направленность образовательной программы

Системное и прикладное программирование

*(указывается профиль / магистерская программа / специализация)*

Форма обучения

Очная/очно-заочная/заочная

*(очная / очно-заочная / заочная)*

Год начала подготовки 2022

Арзамас

2023 год

## 1. Место дисциплины (модуля) в структуре ООП

Дисциплина Б1.В.16 «Объектно-ориентированное программирование» относится к части, формируемой участниками образовательных отношений, образовательной программы направления подготовки 09.03.03 Прикладная информатика, направленность (профиль) Системное и прикладное программирование.

Дисциплина предназначена для освоения студентами очной/очно-заочной/заочной формы обучения в 4, 5 семестре/4, 5 семестре/4, 5 семестре.

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции* (код, содержание индикатора)	Результаты обучения по дисциплине (дескрипторы компетенции) **	
ПК-8. Способен разрабатывать лингвистическое, информационное и программное обеспечение ИС (ИИС) и сопровождающую его документацию	ПК-8.1. Демонстрирует знание современных языков и систем программирования, формализмов описания знаний на концептуальном и инфологическом уровнях, требований к технической документации на все виды обеспечения ИС (ИИС).	<i>Знать</i> современное состояние и принципиальные возможности языков программирования и использующих его систем программирования; <i>Уметь</i> устанавливать, тестировать, испытывать и использовать программные средства, ставить задачи и разрабатывать алгоритм их решения, используя программное обеспечение <i>Владеть</i> навыками разработки и отладки программ, основными шаблонами проектирования программных систем с использованием технологии программирования,	<i>Контрольные задания по теоретическим основам дисциплины</i> <i>Вопросы для устного опроса</i>
	ПК-8.2. Применяет современные языки и системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях при разработке лингвистического, информационного и программного обеспечения ИИС и сопровождающей ее документации.	<i>Знать</i> возможности программного обеспечения для проведения анализа социально-экономических задач и процессов с применением методов системного анализа и математического моделирования. <i>Уметь</i> разрабатывать основные программные документы; работать с современными системами программирования, включая объектно-ориентированные. <i>Владеть</i> приемами разработки прикладных программ на различных языках.	<i>Практические контрольные задания</i> <i>Тест</i>
	ПК-8.3. Имеет практический опыт разработки лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей ее документации.	<i>Знать</i> особенности осуществления разработки лингвистического, информационного и программного обеспечения конкретной ИС <i>Уметь</i> разрабатывать программное обеспечение ИС и сопровождающую его документацию <i>Владеть</i> способностью осуществлять разработку лингвистического,	<i>Учебно-исследовательские реферативные работы</i>

		информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей его документации.	
--	--	---------------------------------------------------------------------------------------------------	--

### 3. Структура и содержание дисциплины

#### 3.1. Структура дисциплины

Трудоемкость	очная форма обучения	очно-заочная форма обучения	заочная форма обучения
<b>Общая трудоемкость</b>	8 з.е.		
часов по учебному плану, из них	288		
	<b>Контактная работа</b> , в том числе: аудиторные занятия:		
– занятия лекционного типа	70	34	
– занятия семинарского типа	52	24	10
контроль самостоятельной работы	3	3	3
<b>Промежуточная аттестация</b> Экзамен, зачет	36	36	13
<b>Самостоятельная работа</b>	91	191	262

#### 3.2. Содержание дисциплины

(структурированное по разделам (темам) с указанием отведенного на них количества академических часов и виды учебных занятий)

4 семестр/4 семестр/4 семестр

Наименование разделов (Р) или тем (Т) дисциплины (модуля), Форма(ы) промежуточной аттестации по дисциплине	Всего (часы)			Контактная работа (работа во взаимодействии с преподавателем), часы, из них									Самостоятельная работа обучающегося, часы, в период						
				Занятия лекционного типа			Занятия семинарского типа (в т.ч. текущий контроль успеваемости)			Контроль самостоятельной работы			промежуточной аттестации (контроля)			теоретического обучения			
	семинары, практические занятия	лабораторные работы	Очная				Очно-заочная	Заочная											
	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	
1. Объектно-ориентированный подход в программировании	8	13	13	4	2					2	2						2	9	13
2. Классы и объекты. Отношения.	8	13	12	4	2					2	2						2	9	12
3. Языки ООП и их классификация	8	13	12	4	2					2	2						2	9	12
4. Наследование и полиморфизм	8	13	10	4	2					2	2						2	9	10
5. Основы программирования на языке ООП	8	11	12	4	2					2		1					2	9	11

6. Методы и классы	8	11	12	4	2					2		1						2	9	11			
7. Операции и операторы	8	11	11	4	2					2		1						2	9	10			
8. Массивы.	8	11	11	4	2					2		1						2	9	10			
9. Коллекции и словари	7	11	10	4	2					2								1	9	10			
<b>В том числе текущий контроль</b>	<b>1</b>	<b>1</b>	<b>1</b>											<b>1</b>	<b>1</b>	<b>1</b>							
Зачет			4															4					
<b>ИТОГО</b>	<b>108</b>	<b>108</b>	<b>108</b>	<b>36</b>	<b>18</b>									<b>18</b>	<b>8</b>	<b>4</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>4</b>	<b>17</b>	<b>81</b>	<b>99</b>

5 семестр/5 семестр/5 семестр.

Наименование разделов (Р) или тем (Т) дисциплины (модуля), Форма(ы) промежуточной аттестации по дисциплине	Всего (часы)			Контактная работа (работа во взаимодействии с преподавателем), часы, из них									Самостоятельная работа обучающегося, часы, в период												
				Занятия лекционного типа			Занятия семинарского типа (в т.ч. текущий контроль успеваемости)			Контроль самостоятельной работы			промежуточной аттестации (контроля)			теоретического обучения									
	семинары, практические занятия	лабораторные работы																							
	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная	Очная	Очно-заочная	Заочная							
10. Язык запросов LINQ	16	13	16	4	2		4	2												8	9	16			
11. Классы и их основные элементы	14	13	16	4	2		4	2													6	9	16		
12. События и данные в классе	14	13	16	4	2		4	2													6	9	16		
13. Отношения между классами	14	13	16	4	2		4	2													6	9	16		
14. Структуры и перечисления	14	13	12	4	2		4	2													6	9	12		
15. Разработка консольных приложений	10	13	14	2	2		2	2	1												6	9	13		
16. Формы и работа с ними	10	13	14	2	2		2	2	1												6	9	13		
17. Управление формами	10	13	14	2	2		2	2	1												6	9	13		
18. Диалоговые окна	10	9	14	2			2		1												6	9	13		
19. Формы и графика	10	9	14	2			2		2												6	9	12		
20. Файлы. Классы для работы с файлами	10	10	12	2			2														6	10	12		
21. Классы и базы данных	10	10	11	2			2														6	10	11		
<b>В том числе текущий контроль</b>	<b>2</b>	<b>2</b>	<b>2</b>											<b>2</b>	<b>2</b>	<b>2</b>									
Экзамен	36	36	9																		36	36	9		
<b>ИТОГО</b>	<b>180</b>	<b>180</b>	<b>180</b>	<b>34</b>	<b>16</b>		<b>34</b>	<b>16</b>	<b>6</b>					<b>2</b>	<b>2</b>	<b>2</b>				<b>36</b>	<b>36</b>	<b>9</b>	<b>74</b>	<b>110</b>	<b>163</b>

Текущий контроль успеваемости реализуется в рамках занятий семинарского типа, групповых или индивидуальных консультаций.

#### **4. Учебно-методическое обеспечение самостоятельной работы студентов**

Самостоятельная работа является важнейшей составной частью учебного процесса и обязанностью каждого студента.

Для обеспечения самостоятельной работы обучающихся используется электронный управляемый курс «Объектно-ориентированное программирование» <https://e-learning.unn.ru/course/view.php?id=7975>, созданный в системе электронного обучения ННГУ <https://e-learning.unn.ru/>.

Самостоятельная работа студентов по дисциплине «Объектно-ориентированное программирование» осуществляется в следующих видах: работа с основной и дополнительной литературой, выполнение заданий различных типов, составления тезисов литературных источников, подготовки рефератов, разработка проектных работ, подготовка презентаций.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.3.

#### **Методические рекомендации к самостоятельной работе**

##### **Методические рекомендации по подготовке к занятиям семинарского типа**

Подготовка к занятиям семинарского типа (практическим занятиям) – традиционная форма самостоятельной работы обучающихся, включает отработку лекционного материала, изучение рекомендованной литературы, конспектирование предложенных источников.

Подготовка к опросу, проводимому в рамках практического занятия, требует уяснения вопросов, вынесенных на конкретное занятие, подготовки выступлений, повторения основных терминов, запоминания формул и алгоритмов.

На практических занятиях рассматриваются наиболее важные, существенные, сложные вопросы, которые, как свидетельствует преподавательская практика, наиболее трудно усваиваются студентами. Готовиться к практическим занятиям необходимо заблаговременно.

Подготовка к семинарским (практическим) занятиям включает в себя:

- обязательное ознакомление с планом практического занятия, в котором содержатся основные вопросы, выносимые на обсуждение;
- изучение конспектов лекций, соответствующих разделов учебника, учебного пособия, содержания рекомендованных нормативных правовых актов;
- изучение дополнительной литературы по теме практического занятия с обязательным конспектированием материала, который понадобится при обсуждении на семинаре.

Помните, что необходимо:

- выписать основные термины и запомнить их дефиниции;
- записывать возникшие во время самостоятельной работы с учебниками и научной литературы вопросы, чтобы затем на семинаре получить на них ответы;
- иметь продуманные и аргументировано обоснованные формулировки собственной позиции по каждому вопросу плана практического занятия;
- обращаться за консультацией к преподавателю при возникновении затруднений в освоении материала практической работы.

Выступление на практических занятиях должно удовлетворять следующим требованиям: в выступлении излагаются теоретические подходы к рассматриваемому вопросу, дается анализ принципов, законов, понятий и категорий; теоретические положения подкрепляются фактами, примерами, выступление должно быть аргументированным. Для более углубленного изучения вопросов рекомендуется конспектирование основной и дополнительной литературы.

Большую помощь при подготовке к занятиям может оказать изучение публикаций в научных журналах, а также специальные Интернет-ресурсы по тематике дисциплины, указанные п. 6 настоящей рабочей программы дисциплины

## **Рекомендации для работы с основной и дополнительной литературой**

Работа с литературой должна сопровождаться записями в форме конспекта, плана, тезисов. При этом важно не только привлечь более широкий круг литературы, но и суметь на ее основе разобраться в степени изученности темы. Стоит выявить дискуссионные вопросы, нерешенные проблемы, попытаться высказать свое отношение к ним. Привести и аргументировать свою точку зрения или отметить, какой из имеющихся в литературе точек зрения по данной проблематике придерживаетесь и почему.

По завершении изучения рекомендуемой литературы полезно проверить уровень своих знаний с помощью контрольных вопросов для самопроверки. Необходимо вести систематическую работу над литературными источниками. Необходимо изучать не только литературу, рекомендуемую в данных учебно-методических материалах, но и новые, важные издания по курсу, вышедшие в свет после публикации. При этом следует выделять неясные, сложные для восприятия вопросы. В целях прояснения последних нужно обращаться к преподавателю.

### **Рекомендации для написания учебно-исследовательской реферативной работы**

Учебно-исследовательская реферативная работа – изложение в письменном виде содержания научного труда (трудов), литературы по теме. Цель написания учебно-исследовательской реферативной работы – овладение навыками анализа и краткого изложения изученных материалов в соответствии с требованиями, предъявляемыми к таким работам. Это самостоятельная работа студента, где раскрывается суть исследуемой проблемы, приводятся различные точки зрения, собственные взгляды на нее. Содержание работы должно быть логическим, изложение материала носит проблемно-тематический характер.

#### ***Примерный алгоритм действий при написании реферата:***

1. Подберите и изучите основные источники по теме (как правило, при разработке реферата или доклада используется не менее 8-15 различных источников).
2. Составьте библиографию.
3. Разработайте план реферата или доклада исходя из имеющейся информации.
4. Обработайте и систематизируйте подобранную информацию по теме.
5. Отредактируйте текст реферата или доклад с использованием компьютерных технологий.
6. Подготовьте публичное выступление по материалам реферата или доклада, желательно подготовить презентацию, иллюстрирующую основные положения работы.

Критерии результатов работы для самопроверки:

- актуальность темы исследования;
- соответствие содержания теме;
- глубина проработки материала;
- правильность и полнота использования источников;
- соответствие оформления реферата или доклада предъявляемым требованиям.

### **Самостоятельное изучение отдельных тем (вопросов) в соответствии со структурой дисциплины по учебной и специальной литературе**

Активизация учебной деятельности и индивидуализация обучения предполагает вынесение для самостоятельного изучения отдельных тем или вопросов. Выбор тем (вопросов) для самостоятельного изучения – одна из ключевых проблем педагога в организации эффективной работы обучающихся по овладению учебным материалом.

Особую роль самостоятельное изучение отдельных тем (вопросов) дисциплины играет для студентов заочной формы обучения.

При этом, как правило, основанием выбора является наилучшая обеспеченность литературой и учебно-методическими материалами по данной теме, ее обобщающий характер, сформированный на аудиторных занятиях алгоритм изучения. Обязательным условием результативности самостоятельного освоения темы (вопроса) является контроль выполнения задания.

Вопросы для самостоятельного изучения тем (вопросов) указаны в рабочей программе дисциплины (модуля)».

Результаты самостоятельного изучения вопросов, будут проверены преподавателем в форме: опросов, конспектов, рефератов, ответов на экзаменах.

### **Самостоятельное выполнение расчетных заданий**

1. Внимательно прочитайте теоретический материал – конспект, составленный на лекционном занятии, материал учебника, пособия. Выпишите формулы из конспекта по изучаемой теме.

2. Обратите внимание, как использовались данные формулы при решении задач на занятии.

3. Решите предложенную задачу, используя выписанные формулы.

4. В случае необходимости воспользуйтесь справочными данными.

5. Проанализируйте полученный результат (проверьте размерности величин, правильность подстановки в формулы численных значений, правильность расчетов, правильность вывода неизвестной величины из формулы).

6. Решение задач должно сопровождаться необходимыми пояснениями. Расчётные формулы приводите на отдельной строке, выделяя из текста, с указанием размерности величин. Формулы записывайте сначала в общем виде (буквенное выражение), затем подставляйте числовые значения без указания размерностей, после чего приведите конечный результат расчётной величины.

Показатели результатов работы для самопроверки:

- грамотная запись условия задачи и ее решения;
- грамотное использование формул;
- грамотное использование справочной литературы;
- точность и правильность расчетов;
- обоснование решения задачи.

### **Подготовка к промежуточной аттестации: подготовка к экзамену**

#### **Методические рекомендации по подготовке к экзамену**

Экзамен проводится в традиционной форме (ответ на вопросы экзаменационного билета, контрольная работа, тестирование) и/или в иных формах (с учетом оценок за коллоквиум, кейс, деловая или ролевая игра, презентация проекта и др.)

Подготовка к зачету, экзамену начинается с первого занятия по дисциплине. При этом важно с самого начала планомерно осваивать материал, руководствуясь требованиями, конспектировать важные для решения учебных задач источники, обращаться к преподавателю за консультацией по неусвоенным вопросам.

Для подготовки к сдаче зачета, экзамена необходимо первоначально прочитать лекционный материал, а также соответствующие разделы рекомендуемых изданий. Лучшим вариантом является тот, при котором при подготовке используется несколько источников информации. Это способствует разностороннему восприятию каждой конкретной темы дисциплины.

В обобщённом варианте подготовка к сдаче зачета, экзамена включает в себя:

- просмотр программы учебной дисциплины, перечня вопросов к зачету, экзамену;
- подбор рекомендованных преподавателем источников (учебников, нормативных правовых актов, дополнительной литературы и т.д.),
- использование конспектов лекций, материалов занятий и их изучение;

- консультирование у преподавателя.

### Учебно-методические документы, регламентирующие самостоятельную работу

адреса доступа к документам:

<https://arz.unn.ru/sveden/document/>

[https://arz.unn.ru/pdf/Metod\\_all\\_all.pdf](https://arz.unn.ru/pdf/Metod_all_all.pdf)

## 5. Фонд оценочных средств для промежуточной аттестации по дисциплине

### 5.1. Описание шкал оценивания результатов обучения по дисциплине

В ходе промежуточной аттестации по дисциплине осуществляется оценка сформированности компонентов компетенций (полнота знаний/ наличие умений/ навыков), т.е. результатов обучения, указанных в таблице п.2 настоящей рабочей программы, на основе оценки усвоения содержания дисциплины.

Обобщенная оценка сформированности компонентного состава компетенции в ходе промежуточной аттестации по дисциплине проводится на основе учета текущей успеваемости в ходе освоения дисциплины и учета результата сдачи промежуточной аттестации.

Выявленные признаки несформированности компонентов (индикаторов) хотя бы одной компетенции не позволяют выставить интегрированную положительную оценку сформированности компетенций и освоения дисциплины на данном этапе обучения.

Обобщенная оценка сформированности компонентного состава компетенций на промежуточной аттестации, которая вносится в зачетно-экзаменационную ведомость по дисциплине и зачетную книжку студента, осуществляется по следующей оценочной шкале.

### Шкала оценки сформированности компонентного состава компетенций на промежуточной аттестации

Оценка		Уровень подготовки
Зачтено	Отлично	сформированность компонентного состава (индикаторов) компетенций соответствует требованиям компетентностной модели будущего выпускника на данном этапе обучения, основанным на требованиях ОС ННГУ по направлению подготовки, студент готов самостоятельно решать стандартные и нестандартные профессиональные задачи в предметной области дисциплины в соответствии с типами задач профессиональной деятельности осваиваемой образовательной программы
	Хорошо	сформированность компонентного состава (индикаторов) компетенций соответствует требованиям компетентностной модели будущего выпускника на данном этапе обучения, основанным на требованиях ОС ННГУ по направлению подготовки, но студент готов самостоятельно решать только различные стандартные профессиональные задачи в предметной области дисциплины в соответствии с типами задач профессиональной деятельности осваиваемой образовательной программы
	Удовлетворительно	сформированность компонентного состава (индикаторов) компетенций соответствует в целом требованиям компетентностной модели будущего выпускника на данном этапе обучения, основанным на требованиях ОС ННГУ по направлению подготовки, но студент способен решать лишь минимум стандартных профессиональных задач в предметной области дисциплины в соответствии с типами задач профессиональной деятельности осваиваемой образовательной программы
Не зачтено	Неудовлетворительно	сформированность компонентного состава (индикаторов) компетенций не соответствует требованиям компетентностной модели будущего выпускника на данном этапе обучения, основанным на требованиях ОС ННГУ по направлению подготовки, студент не готов решать профессиональные задачи в предметной области дисциплины в соответствии с типами задач профессиональной деятельности осваиваемой образовательной программы

## Шкала оценивания сформированности компетенции

Уровень сформированности компетенции (индикатора достижения компетенции)	неудовлетворительно	удовлетворительно	хорошо	отлично
	не зачтено	зачтено		
<b><u>Знания</u></b>	Уровень знаний ниже минимальных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибок.	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок.	Уровень знаний в объеме, соответствующем требованиям программы подготовки, без ошибок.
<b><u>Умения</u></b>	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	Продемонстрированы основные умения, решены типовые задачи с негрубыми ошибками, выполнены все задания, но не в полном объеме.	Продемонстрированы все основные умения, решены все основные задачи с негрубыми ошибками, выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения, решены все основные задачи с отдельными незначительными недочетами, выполнены все задания в полном объеме.
<b><u>Навыки</u></b>	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами.	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.

### 5.2 Критерии и процедуры оценивания результатов обучения по дисциплине

#### Критерии оценки устного опроса на занятии/ экзамене

*Оценка «отлично»* - Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности, грамотный научный язык; ответ самостоятельный.

*Оценка «хорошо»* - Ответ полный и правильный, на основании изученной теории; материал изложен в определенной логической последовательности при этом допущены две-три незначительные ошибки, исправленные по требованию преподавателя.

*Оценка «удовлетворительно»* - Ответ полный, но при этом допущена существенная ошибка или неполный, несвязный ответ.

*Оценка «неудовлетворительно»* - Ответ обнаруживает непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые не могут быть исправлены при наводящих вопросах преподавателя.

#### Критерии оценивания письменных контрольных работ

*оценка «отлично»* выставляется студенту, если представленная контрольная работа выполнена полностью без ошибок и недочетов;

*оценка «хорошо»* выставляется студенту, если представленная контрольная работа выполнена полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов;

*оценка «удовлетворительно»* выставляется студенту, если представленная им контрольная работа выполнена правильно не менее чем на 2/3 всей работы или в работе допущены не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и трех недочетов, при наличии четырех-пяти недочетов;

*оценка «неудовлетворительно»* выставляется студенту, если число ошибок и недочетов в работе превысило норму для оценки 3 или правильно выполнено менее 2/3 всей работы.

#### Критерии оценки тестирования

*Оценка "отлично"* - 85-100% правильных ответов;

Оценка "хорошо" 66-84 % правильных ответов;  
Оценка "удовлетворительно" – 50-65 % правильных ответов;  
Оценка "неудовлетворительно" - меньше 50 %.

### 5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения и для контроля формирования компетенции

4 семестр/4 семестр/4 семестр

#### Типовые вопросы для устного опроса для оценки сформированности компетенций ПК-8

1. Понятие типа. Константы и переменные. Операции и формы их записи.
2. Сортировка одномерного массива методом обмена.
3. Логический тип данных. Диапазон значений логического типа данных. Логические операции.
4. Сортировка одномерного массива методом вставки.
5. Целочисленные типы данных. Прямой, обратный и дополнительный код целочисленных типов данных. Диапазоны значений целочисленных типов данных.
6. Сортировка одномерного массива методом выбора.
7. Целочисленные типы данных. Операции над целочисленными типами данных.
8. Задача о минимаксе.
9. Вещественные типы данных. Операции над вещественными типами данных.
10. Открытые массивы в языке .
11. Вещественные типы данных. Представление вещественных типов данных на примере модельного типа.
12. Многомерные статические массивы в языке .
13. Вещественные типы данных. Представление вещественных типов данных в памяти компьютера. Диапазоны значений вещественных типов данных.
14. Примеры задач на обработку массивов.
15. Символьный тип данных. Литералы символьного типа.

#### Типовые тестовые задания для оценки сформированности компетенций ПК-8

1. **Какую роль выполняет наследование?**
  - а). расширить поведение базового (или родительского) класса, наследуя основную функциональность в производном подклассе;
  - б). расширить поведение производного класса (класса-потомка), наследуя основную функциональность в исходных файлах;
  - в). обеспечивать взаимодействие класса-родителя и класса-потомка.
2. **Объектно-ориентированное программирование - это:**
  - а) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.
  - б) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является отдельно существующей структурой (функцией, переменной, процедурой).
  - в) методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых может быть приведен к общему виду (структуре) программы.
3. **Классы — это:**
  - а) аналоги переменных и структур в процедурном программировании.

- б) это абстракция описывающая методы, свойства, ещё не существующих объектов.
- в) это абстракция описывающая методы, свойства существующих объектов.

**4. Три основных принципа построения классов – это:**

- а) инкапсуляция, полиморфизм, наследование.
- б) наследование, распределенность, структурируемость.
- в) полиморфизм, инкапсуляция, стандартизация.

**5. Инкапсуляция – это:**

- а) механизм объектно-ориентированного подхода, позволяющий расширить и/или изменить структуру уже существующего (родительского) класса, путём написания нового класса (потомка), который полностью наследует все свойства и методы и добавляет свои.
- б) это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.
- в) это свойство объектов скрывать некоторые свои данные и способы их обработки (методы) от окружающей его цифровой среды.

**6. Три основных уровня инкапсуляции:**

- а) open, closed и average.
- б) public, protected и private.
- в) public, not public и average.

**7. Конструктор предназначен для ...**

- а) инициализации данных класса при создании нового экземпляра.
- б) удаления объекта заданного класса.
- в) определения структуры класса.

**8. Какое из описаний конструктора будет верным?**

- а) ~Student (char \*);
- б) Student::~~Student ()
- в) Student (char \*);

**9. Какое из описаний деструктора будет верным?**

- а) ~Student (char \*);
- б) Student::~~Student ()
- в) Student (char \*);

**10. В заданной строке определяется имя класса и имя метода (функции) класса void Student::show (void), а именно:**

- а) имя класса – Student, метода - show.
- б) имя класса – show, метода - void.
- в) имя класса – show, метода - Student.

**11. В заданном классе определите методы и объекты класса**

```
class Students {  
public:  
void set_name(std::string);  
std::string get_name();  
void set_last_name(std::string);  
std::string get_last_name();  
private:  
std::string name;  
std::string last_name;
```

};

а) методы класса – name, last\_name; объекты класса – set\_name, get\_name, set\_last\_name, get\_last\_name.

б) методы класса – set\_name, get\_name, set\_last\_name, get\_last\_name; объекты класса – name, last\_name.

в) методы класса – void, string; объекты класса – private, public.

**12. Можно ли в производном классе объявлять новые поля и методы?**

а) Нет;

б) Да.

**13. Обязательно ли делать поля класса приватными?**

а) Нет;

б) Да.

**14. Что такое виртуальная функция?**

а) пользовательская функция;

б) функция, которая вызывает сама себя;

в) функция базового класса, переопределенная в одном или в нескольких производных классах;

г) функция дочернего класса, переопределенная в одном или в нескольких производных классах;

**15. Полиморфизм – это...**

а). способность функции обрабатывать данные разных типов;

б). способность функции вызывать методы раздела private любой другой функции программы;

в). свойство базового класса;

г). способ описания методов класса.

**16. Метод это:**

а). Функция класса, получающая в качестве обязательного параметра указатель на объект

б). Структура, хранящая указатели this, parent, supper

в). Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии

г). Определенный программистом абстрактный тип данных

**17. Объект это:**

а). Функция класса, получающая в качестве обязательного параметра указатель на объект

б). Структура, хранящая указатели this, parent, supper

в). Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии

г). Определенный программистом абстрактный тип данных

**18. Для создания контекстного меню используется компонент класса:**

а). TButton

б). TEdit

в). TCheckBox

г). TPopupMenu

д). TBitBtn

**19. Для создания многострочного редактируемого текстового поля используется компонент класса:**

- а). TLabel
- б). TMainMenu
- в). TMemo
- г). TEdit
- д). TForm

**20. Для создания главного меню используется компонент класса:**

- а). TLabel
- б). TMenu
- в). TEdit
- г). TMainMenu
- д). TPopupMenu

**21. Для создания однострочного редактируемого текстового поля используется компонент класса:**

- а). TEdit
- б). TMemo
- в). TLabel
- г). TPopupMenu
- д). TPanel

**22. Свойство, отвечающее за высоту формы:**

- а). Top
- б). Size
- в). Style
- г). Width
- д). Height

**23. Типы данных бывают:**

- а). Integer, real, char, for
- б). Float, integer, if, string
- в). Integer, float, char, string, boolean
- г). Integer, real, printf, string
- д). For, scanf, integer, float

**24. Для создания строки состояния используется компонент класса:**

- а). TProgressBar
- б). TCoolBar
- в). TToolBar
- г). TStatusBar
- д). TTrackBar

**25. Какое свойство компонента класса TEdit отвечает за текст находящийся в нём:**

- а). String
- б). Text
- в). Lines
- г). Caption
- д). Edit

**26. Вкладка Events в Object Inspector предназначена для...**

- а). Изменения свойств компонентов
- б). Редактирования кода программы
- в). Поиска компонентов
- г). Просмотра структуры проекта
- д). Определения событий

**27. Для выравнивания текста по правому или левому краю у компонента класса TEdit имеется свойство...**

- а). Align
- б). AutoSize
- в). Font
- г). Alignment
- д). Left

**28. Для того чтобы сделать любой компонент невидимым предназначено свойство...**

- а). Visible
- б). Enable
- в). Style
- г). Transparent
- д). Glyph

**29. Ограничение на количество вводимых символов для компонента Edit задётся в свойстве...**

- а). Text
- б). Left
- в). Font
- г). AutoSize
- д). MaxLength

**30. Когда нажата кнопка мыши, генерируется событие:**

- а). OnMouseMove
- б). OnMouseUp
- в). OnMouseDown
- г). OnMouseWheel
- д). OnClick

**31. Событие OnCreate генерируется, когда...**

- а). Окно закрывается
- б). Окно создается
- в). Окно изменяет размеры
- г). Пользователь щелкнул по форме
- д). Окно меняет цвет

**32. Вкладка Properties в Object Inspector предназначена для...**

- а). Редактирования кода программы
- б). Поиска компонентов
- в). Просмотра структуры проекта
- г). Определения событий
- д). Изменения свойств компонентов

**33. Функция, выполняющая преобразование числа в строку имеет вид:**

- а). StrToInt(s)

- б). FloatToStr(n)
- в). Chr(n)
- г). IntToStr(a,s)
- д). StrToFloat(s)

**34. Свойство формы, отвечающее за заголовок окна.**

- а). Caption
- б). Text
- в). Name
- г). Font
- д). String

**35. Что будет выведено на экран при выполнении фрагмента кода...**

```
int c;
int a=5, b=2;
c = a/b;
cin>>a<<"/"<<b<<"="<<c<<endl;
```

- а). 5 / 2 = 2
- б). 5 / 2 = 2,5
- в). 5 / 2 = 3
- г). 5, 2, 5 / 2
- д). 5, 2, 2,5

**36. Что будет выведено на экран при выполнении фрагмента кода...**

```
double c;
double a=5, b=2;
c = a/b;
cin>>a<<"/"<<b<<"="<<c<<endl;
```

- а). 5 / 2 = 2
- б). 5 / 2 = 2,5
- в). 5 / 2 = 3
- г). 5, 2, 5 / 2
- д). 5, 2, 2,5

**37. Какое имя получит обработчик события соответствующий щелчку мыши на кнопке TButton?**

- а). void TForm1::Button1Enter(TObject \*Sender)
- б). void TForm1::Button1MouseDown(TObject \*Sender, TMouseButton Button, TShiftState Shift, int X, int Y)
- в). void TForm1::Button1KeyPress(TObject \*Sender, wchar\_t &Key)
- г). void TForm1::Button1Click(TObject \*Sender)
- д). void TForm1::Button1MouseEnter(TObject \*Sender)

**38. Если имеется код int x; cin >> x; и вводится 1.2, то что будет в переменной x?**

- а). 1,2
- б). 2
- в). 2,0
- г). 1,2
- д). 1

**39. В программе на языке C++ обязательно имеется функция...**

- а). head

- б). start
- в). prime
- г). main
- д). finish

**40. Какое свойство компонента TCheckBox определяет состояние флажка?**

- а). Enabled
- б). Visible
- в). Checked
- г). Hint
- д). State

### Типовая контрольная работы для оценки сформированности компетенций ПК-8

#### Задание 1.

Составить на любом языке программирования консольное приложение, которое содержит описание класса **Time** (время), который должен содержать:

Класс должен включать:

1. Закрытые свойства для хранения часов и минут
2. Методы доступа к закрытым свойствам
3. Конструктор или несколько конструкторов, для создания экземпляров класса
4. Метод отображения на экране времени в формате (чч:мм)

Программа должна делать следующее:

1. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
2. Задать им следующие значения (2ч 30м, 5ч 15м, 3ч 45м)
3. Вывести на экран время, хранящееся во всех объектах.
4. Рассчитать разницу в днях между 1 и 2 объектами и вывести ее на экран.

#### Задание 2.

Составить на любом языке программирования консольную программу, которая содержит описание класса

Date - дата (год, месяц, день)

Класс должен включать:

1. Закрытые свойства для хранения год, месяц, день.
  2. Методы доступа к закрытым свойствам.
  3. Конструктор или несколько конструкторов, для создания объектов класса.
  4. Метод - показать на экране время в формате (дд/мм/гг)
  5. Метод - рассчитать количество дней с начала года до даты  
public int Days( )
1. Программа должна делать следующее:
2. В функции main() нужно объявить и создать массив из 3 объектов описанного класса
  3. Задать им следующие значения (1.5.2001 5.2.2002 13.7.2001)
  4. Вывести на экран даты, хранящиеся во всех объектах.
  5. Рассчитать разницу в днях между 1 и 3 объектами и вывести ее на экран

### Вопросы к зачету

Вопрос	Код компетенции

1. Статические одномерные массивы.	ПК-8
2. Основные методы строкового типа данных.	ПК-8
3. Алгоритм бинарного поиска.	ПК-8
4. Строковый тип данных, сравнение строк	ПК-8
5. Алгоритм линейного поиска	ПК-8
6. Понятие типа. Константы и переменные. Операции и формы их записи	ПК-8
7. Сортировка одномерного массива методом обмена	ПК-8
8. Логический тип данных. Диапазон значений логического типа данных. Логические операции	ПК-8
9. Сортировка одномерного массива методом вставки.	ПК-8
10. Целочисленные типы данных. Прямой, обратный и дополнительный код целочисленных типов данных. Диапазоны значений целочисленных типов данных	ПК-8
11. Сортировка одномерного массива методом выбора	ПК-8
12. Целочисленные типы данных. Операции над целочисленными типами данных	ПК-8
13. Задача о минимаксе	ПК-8
14. Вещественные типы данных. Операции над вещественными типами данных	ПК-8
15. Открытые массивы	ПК-8
16. Вещественные типы данных. Представление вещественных типов данных на примере модельного типа	ПК-8
17. Многомерные статические массивы в языке Pascal	ПК-8
18. Вещественные типы данных. Представление вещественных типов данных в памяти компьютера. Диапазоны значений вещественных типов данных	ПК-8
19. Примеры задач на обработку массивов	ПК-8
20. Символьный тип данных. Литералы символьного типа	ПК-8
21. Динамические одномерные массивы	ПК-8
22. Процедуры. Процедуры-функции	ПК-8
23. Метод пошаговой детализации	ПК-8
24. Потоки ввода/вывода	ПК-8
25. Пример синтаксического анализа	ПК-8
26. Классификация программного обеспечения	ПК-8
27. Задача о расстановке скобок	ПК-8
28. Связный список и его реализация	ПК-8
29. Трансляция. Виды трансляций	ПК-8
30. Стек и его реализация в языке	ПК-8

### Вопросы к экзамену

5 семестр/5 семестр/5 семестр

#### Типовые вопросы для устного опроса для оценки сформированности компетенций ПК-8

1. Динамические одномерные массивы в языке .
2. Процедуры. Процедуры-функции.
3. Статические одномерные массивы в языке.
4. Основные методы строкового типа данных.
5. Алгоритм бинарного поиска.
6. Строковый тип данных, сравнение строк.

7. Алгоритм линейного поиска.
8. Метод пошаговой детализации.
9. Потоки ввода/вывода.
10. Пример синтаксического анализа.
11. Классификация программного обеспечения.
12. Задача о расстановке скобок.
13. Основные элементы структуры платформы .Net.
14. Связный список и его реализация в языке .
15. Трансляция. Виды трансляций.

**Типовые тестовые задания для оценки  
сформированности компетенций ПК-8**

1. ПРОГРАММА, ПЕРЕВОДЯЩАЯ ВХОДНУЮ ПРОГРАММУ НА ИСХОДНОМ ЯЗЫКЕ В ЭКВИВАЛЕНТНУЮ ЕЙ ВЫХОДНУЮ ПРОГРАММУ НА МАШИННОМ ЯЗЫКЕ, НАЗЫВАЕТСЯ:

1. Ассемблер
2. Линковщик
3. Интерпретатор
4. Компилятор

2. КАКОЕ ИЗ ПРИВЕДЕННЫХ НИЖЕ ИМЕН ПЕРЕМЕННЫХ РАЗРЕШЕНО ПРАВИЛАМИ C++?

1. 666666xyz
2. The first
3. \_\_\_counter\_\_\_
4. File-name

3. ОПЕРАТОРЫ МОГУТ БЫТЬ ОБЪЕДИНЕНЫ В БЛОК:

1. Заключением в угловые скобки
2. Заключением в фигурные скобки
3. Заключением в квадратные скобки
4. Заключением в кавычки

4. КАКОЙ ИЗ ПЕРЕЧИСЛЕННЫХ ТИПОВ ДАННЫХ НЕ ЯВЛЯЕТСЯ ТИПОМ ДАННЫХ В C++?

1. Int
2. Double
3. Float
4. Real

5. КАКИЕ ЗНАЧЕНИЯ БУДУТ ИМЕТЬ ПЕРЕМЕННЫЕ a, b, c ПОСЛЕ ВЫПОЛНЕНИЯ СЛЕДУЮЩЕГО ФРАГМЕНТА ПРОГРАММЫ:

```
Int a=4, b=3, c=2;  
If (a--!=2) {c*=b};  
Else {b/=c};
```

- |        |        |        |
|--------|--------|--------|
| 1. a=4 | 2. a=3 | 3. a=3 |
| b=6    | b=3    | b=3    |
| c=2    | c=2    | c=6    |

6. КАКОЕ ЗНАЧЕНИЕ ПОЛУЧИТ ОТНОШЕНИЕ

$$(2*3) \leq (23 \% 8)$$

ПОСЛЕ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ?

1. 6
2. True
3. Integer
4. False

7. ПУСТЬ a1 И a2 - ОПЕРАНДЫ ВЕЩЕСТВЕННОГО ТИПА.  
КАКАЯ ИЗ ПРИВЕДЕННЫХ ЗАПИСЕЙ ЯВЛЯЕТСЯ ЛОГИЧЕСКИ  
ОТНОШЕНИЕМ:

1. A1=a2
2. (a1-a2)<e
3. (a1-3)/(2\*a2)
4. A1><a2

8. КАКОМУ ЗАРЕЗЕРВИРОВАННОМУ СЛОВУ ПРОГРАММА ПЕРЕДАЁТ  
УПРАВЛЕНИЕ В СЛУЧАЕ, ЕСЛИ ЗНАЧЕНИЕ ПЕРЕМЕННОЙ ИЛИ  
ВЫРАЖЕНИЯ ОПЕРАТОРА switch НЕ СОВПАДАЕТ  
НИ С ОДНИМ КОНСТАНТНЫМ ВЫРАЖЕНИЕМ?

1. Default
2. Break
3. Else
3. Case

9. КАКОЕ ЗНАЧЕНИЕ БУДЕТ ИМЕТЬ ПЕРЕМЕННАЯ b  
ПОСЛЕ ВЫПОЛНЕНИЯ СЛЕДУЮЩЕГО ФРАГМЕНТА ПРОГРАММЫ:

```
Int a=6, b=1;  
For (int i=2; i>-5; i--)  
    A=a+i;  
    B+=a;
```

1. B=37
2. B=-5
3. B=1
4. B=0

10. КАКОЕ ЗНАЧЕНИЕ БУДЕТ ИМЕТЬ ПЕРЕМЕННАЯ b ПОСЛЕ ВЫПОЛНЕНИЯ СЛЕДУЮЩЕГО ФРАГМЕНТА ПРОГРАММЫ:

```
Int a=6, b=1, i=2;
While (i>0)
{ a+=i; b+=a; i--; }
```

1. B=18
2. B=-5
3. B=1
4. B=0

11. КАКОЕ ИЗ ПРИВЕДЕННЫХ ВЫРАЖЕНИЙ ЯВЛЯЕТСЯ ПРАВИЛЬНОЙ ЗАПИСЬЮ ОПЕРАТОРА ЦИКЛА:

1. While(x>=0.5) x\*=0.1 do;
2. Do while (x>=0.5) x\*=0.1;
3. While x\*=0.1 do(x>=0.5);
4. Do x\*=0.1 while(x>=0.5);

12. ПУСТЬ i=0. КАКОЕ КОЛИЧЕСТВО РАЗ ВЫПОЛНИТСЯ ЦИКЛ:

```
While(i<5)
{ i++; }
```

1. 1
2. 0
3. 5
4. 4

13. ПУСТЬ j=0. КАКОЕ КОЛИЧЕСТВО РАЗ ВЫПОЛНИТСЯ ЦИКЛ:

```
For(int i=j;i<=4;i++)
{ s++; }
```

1. 4
2. 5
3. 0
4. 1

14. УКАЖИТЕ ПРАВИЛЬНОЕ ОБЪЯВЛЕНИЕ МАССИВА:

1. Int anarray[10];
2. Array anarray[10];
3. Int anarray;
4. Anarray{ 10};

15. ПУСТЬ ИМЕЕМ ТАКИЕ ФРАГМЕНТЫ ПРОГРАММЫ

```
Int s1=0;
For(int i=1;i<=10;i++)
S1+=a[i];
```

```
int s2=1;
for(int i=1;i<=10;i++)
s2*=a[i];
```

```
int s3=0;
for(int i=1;i<=10;i++)
if (a[i]>0) s3++;
```

КАКОЕ ИМЯ ПЕРЕМЕННОЙ, ЗНАЧЕНИЕМ КОТОРОЙ ЯВЛЯЕТСЯ СУММА ЭЛЕМЕНТОВ МАССИВА

1. s1
2. s2
3. s3
4. a

16. УКАЖИТЕ ТИП ВОЗВРАЩАЕМОГО ЗНАЧЕНИЯ СЛЕДУЮЩЕЙ ФУНКЦИИ

```
Int func(char x, float v, double t);
```

1. Char
2. Float
3. Int
4. Double

17. КАК НАЗЫВАЮТСЯ ПЕРЕМЕННЫЕ, ОПИСАННЫЕ В ФУНКЦИИ, (ПО ОТНОШЕНИЮ К ПЕРЕМЕННЫМ ВЫЗЫВАЮЩЕЙ ПРОГРАММЫ) ?

1. Глобальными
2. Локальными
3. Формальными
4. Фактическими

18. КАКОЕ ЗНАЧЕНИЕ ПОЛУЧИТ ПЕРЕМЕННАЯ c В СЛЕДУЮЩЕМ ФРАГМЕНТЕ ПРОГРАММЫ:

```
.....  
Function t(x)  
{ return 5*x+2.7; }  
.....  
c=t(2)-2;
```

1. 10.7
2. 12.7
3. 5.7
4. 0

19. КАКОЕ ЗНАЧЕНИЕ ПОЛУЧИТ ПЕРЕМЕННАЯ v  
В СЛЕДУЮЩЕМ ФРАГМЕНТЕ ПРОГРАММЫ

```
.....  
Function cf(a,b)  
{ return a/b+b; }  
.....  
V=cf(4.4,2);
```

1. 6.2
2. 6.4
3. 4.2
4. 2.2

20. КАКОЙ ТИП ДАННЫХ УКАЗЫВАЕТСЯ, ЕСЛИ ФУНКЦИЯ  
НЕ ВОЗВРАЩАЕТ НИКАКИХ ЗНАЧЕНИЙ?

1. Int
2. Char
3. Void
4. Ничего не указывается

21. ПУСТЬ x="0123456789". КАКОЕ ЗНАЧЕНИЕ ВЕРНЕТ МЕТОД:

```
X->indexof("12");
```

1. 0
2. 1
3. False
4. 2

22. КАКОЙ ОПЕРАТОР ЗАПИШЕТ В String-ПЕРЕМЕННУЮ s2  
ПОДСТРОКУ String-ПЕРЕМЕННОЙ s1 ОТ ЕЕ 3-ГО ДО 7-ГО СИМВОЛА  
ВКЛЮЧИТЕЛЬНО.

1. S2= s1->Substring(3,5)
2. S2=Substring(s1,2,5)
3. S1=s2->Substring(3,7)
4. S2= s1->Substring(2,5)

23. УКАЖИТЕ ПРАВИЛЬНОЕ ОБЪЯВЛЕНИЕ УКАЗАТЕЛЯ В C++

1. Int x;
2. Int \*x;
3. Ptr x;
4. Int &x;

24. В КАКОЙ ИЗ СЛЕДУЮЩИХ ЗАПИСЕЙ ИСПОЛЬЗУЕТСЯ ОПЕРАЦИЯ РАЗЫМЕНОВАНИЯ?

1. Address(a)
2. &a
3. \*a
4. A\*

25. В ПРОГРАММЕ ИСПОЛЬЗОВАНЫ ОБЪЯВЛЕНИЯ:

```
Double S;  
Double *P;
```

КАКОЙ ИЗ ПРИВЕДЕННЫХ НИЖЕ ОПЕРАТОРОВ ДОПУСТИМ ?

1. New S;
2. S=P;
3. P\*=S;
4. S\*=P;
5. Все неверны

26. УКАЖИТЕ ЗАРЕЗЕРВИРОВАННОЕ КЛЮЧЕВОЕ СЛОВО ДЛЯ ДИНАМИЧЕСКОГО ВЫДЕЛЕНИЯ ПАМЯТИ В C++

1. New
2. Malloc
3. Value
4. Create

27. КАК ПРАВИЛЬНО ВЫСВОБОДИТЬ ПАМЯТЬ, ПОСЛЕ ВЫПОЛНЕНИЯ ЭТОГО КОДА?

```
Char *a;  
A = new char[20];
```

1. Delete a;
2. Delete [] a;
3. Delete a[];
4. A[].delete;

28. УКАЖИТЕ ПОРЯДКОВЫЙ НОМЕР ИЗ СПИСКА ТИПОВ, КОТОРЫЙ ИМЕЕТ ПЕРЕМЕННАЯ NUM, ОПИСАННАЯ ТАК:

```
Enum SM {ONE,TWO};  
SM NUM;
```

1. Интервальный;
2. Множество;
3. Перечисление;
4. Целый;

29. ПРАВИЛЬНОЕ ОПРЕДЕЛЕНИЕ СТРУКТУРЫ В C++:

1. Struct a\_struct int a;
2. Struct a\_struct {int a;};
3. Struct a\_struct {int a;}
4. Struct {int a;}

30. КАКОВО ЗНАЧЕНИЕ ИНДЕКСА У ПОСЛЕДНЕГО ЭЛЕМЕНТА В СОЗДАННОМ НИЖЕ МАССИВЕ

```
Int users[] = {1, 2, 3};
```

1. 4
2. 3
3. 0
4. 2

31. КАКАЯ ФУНКЦИЯ ПРЕДНАЗНАЧЕНА ДЛЯ ЗАПИСИ СТРОКИ В ФАЙЛ?

1. Readline()
2. Writeline()
3. Appendtext()
4. Opentext()

32. КАКОЕ ДЕЙСТВИЕ БУДЕТ ВЫПОЛНЕНО ПОСЛЕ ВЫПОЛНЕНИЯ КОМАНДЫ

```
File::opentext(filename);
```

1. Создание текстового файла
2. Открытие текстового файла в режиме записи
3. Открытие текстового файла в режиме чтения
4. Запись текстового файла на жесткий диск

33. КАК СВЯЗАНЫ ПОНЯТИЯ ОБЪЕКТА И КЛАССА?

1. Объект — это экземпляр класса
2. Класс — это экземпляр объекта
3. Это тождественные понятия
4. Эти понятия не связаны

34. КАКОГО СПЕЦИФИКАТОРА ДОСТУПА В КЛАССАХ НЕТ?

1. Public
2. Hidden
3. Private
4. Protected

35. КАКИЕ ЭЛЕМЕНТЫ КЛАССА-НАСЛЕДНИКА НЕ МОГУТ БЫТЬ ПЕРЕОПРЕДЕЛЕНЫ ПО ОТНОШЕНИЮ К РОДИТЕЛЬСКОМУ КЛАССУ?

1. Поля
2. Методы
3. Поля и методы
4. Все элементы могут переопределяться

36. КЛЮЧЕВОЕ СЛОВО `this` — ЭТО:

1. Объявленный неявно в классе указатель на себя
2. Оператор разыменования
3. Указатель на базовый класс
4. Указатель на производный класс

37. ЕСЛИ КЛАСС А ИМЕЕТ ЗАКРЫТЫЕ, ЗАЩИЩЕННЫЕ, ОТКРЫТЫЕ ФУНКЦИИ-ЧЛЕНЫ, ТО ПРОИЗВОДНЫЙ `public`-КЛАСС ОТ А МОЖЕТ НЕПОСРЕДСТВЕННО ОБРАЩАТЬСЯ К:

1. Закрытым, защищенным, открытым функциям-членам базового класса
2. Закрытым, открытым функциям-членам базового класса
3. Защищенным, открытым функциям-членам базового класса
4. Только открытым функциям-членам базового класса

38. ЧТО ЭТО ЗА ОПЕРАТОР `::` ?

1. Наследования объектов
2. Разыменования
3. Описания класса
4. Расширения области видимости

39. ЧЕМ ОТЛИЧАЮТСЯ ВЕРСИИ ПЕРЕГРУЖЕННОЙ ФУНКЦИИ ДРУГ ОТ ДРУГА?

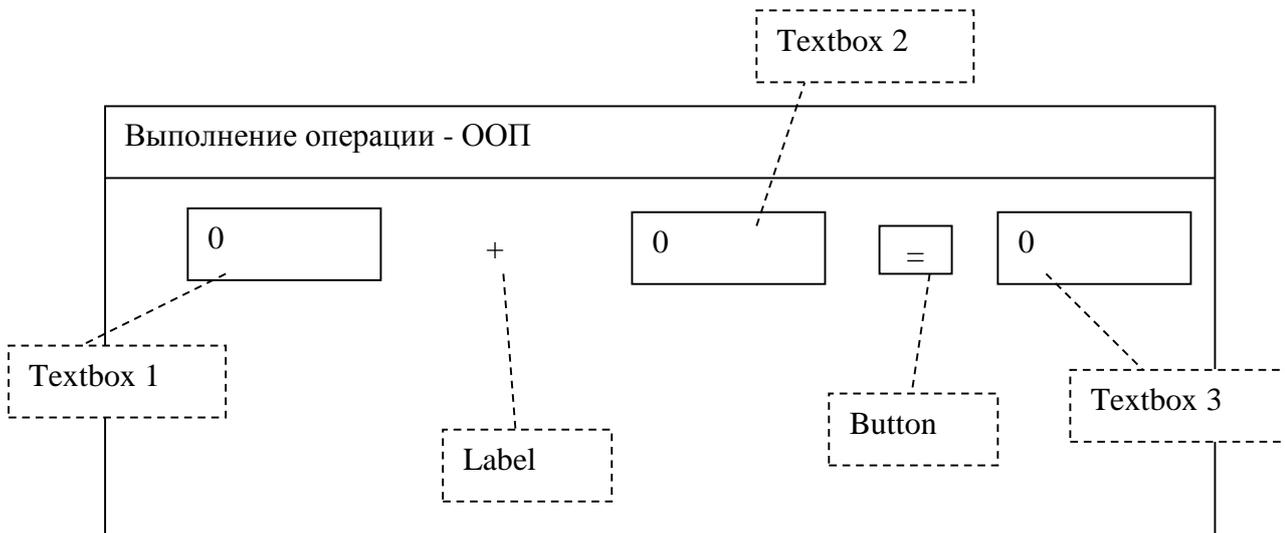
1. Только типом возврата
2. Только типом параметров
3. Только количеством параметров
4. Только типом и/или количеством параметров

40. УКАЖИТЕ БЛОК, В КОТОРЫЙ ЗАКЛЮЧАЕТСЯ КОД, КОТОРЫЙ МОЖЕТ ГЕНЕРИРОВАТЬ ОШИБКУ, СОДЕРЖАЩУЮ ИСКЛЮЧЕНИЕ:

1. `Catch`
2. `Throw`
3. `Try`
4. `Finally`

## Типовая контрольная работы для оценки сформированности компетенций ПК-8

1. Написать программу, которая, используя объект класса производного от класса Form, выводит следующее окно:



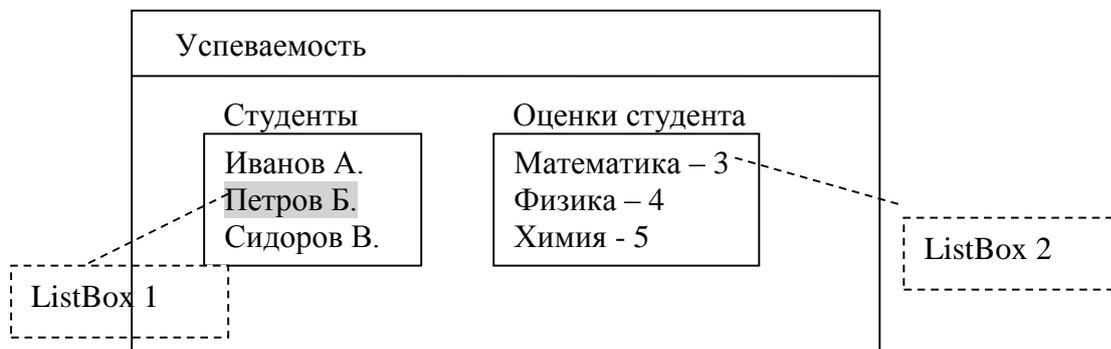
При нажатии кнопки Button, текстовое поле Textbox3 должно показать результат выбранной в ListBox операции над числами, введенными в Textbox1 и Textbox2

При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

2. Написать программу, которая, используя объект класса производного от класса Form, позволяет просматривать информацию об успеваемости студентов по годам:

ФИО студента	Математика	Физика	Химия
Иванов А.	4	5	4
Петров Б.	3	4	5
Сидоров В.	5	5	5

Форма окна программы:



При выборе студента в ListBox1 в ListBox2 должны показываться его оценки.

## Вопросы к экзамену

Вопрос	Код компетенции
1. Архитектура компьютера	ПК-8
2. Понятие предметной области. Объекты и классы	ПК-8
3. Основные идеи и компоненты .Net технологии	ПК-8
4. Последовательность создания и выполнения программ на платформе .Net.	ПК-8
5. Сборка (assembly) и ее структура в .Net технологии. Основные части и их назначение	ПК-8
6. Структура программы на языке ООП.	ПК-8
7. Описания классов на языке ООП.	ПК-8
8. Встроенные типы языка ООП. Неявное и явное приведение (преобразование) встроенных типов	ПК-8
9. Описание и использование методов класса. Перегрузка методов класса	ПК-8
10. Базовый класс System.Object. Назначение, методы	ПК-8
11. Операции языка ООП. Перегрузка операций в создаваемых классах	ПК-8
12. Программирование явного и неявного преобразования пользовательских типов (собственных классов).	ПК-8
13. Динамические одномерные массивы.	ПК-8
14. Процедуры. Процедуры-функции.	ПК-8
15. Метод пошаговой детализации.	ПК-8
16. Потоки ввода/вывода.	ПК-8
17. Пример синтаксического анализа.	ПК-8
18. Классификация программного обеспечения.	ПК-8
19. Задача о расстановке скобок.	ПК-8
20. Связный список и его реализация.	ПК-8
21. Трансляция. Виды трансляций.	ПК-8
22. Стек и его реализация в языке.	ПК-8
23. Архитектура компьютера.	ПК-8
24. Понятие предметной области. Объекты и классы.	ПК-8
25. Основные идеи и компоненты .Net технологии.	ПК-8
26. Последовательность создания и выполнения программ на платформе .Net.	ПК-8
27. Сборка (assembly) и ее структура в .Net технологии. Основные части и их назначение.	ПК-8
28. Структура программы на языке ООП.	ПК-8
29. Описания классов на языке ООП.	ПК-8
30. Встроенные типы языка ООП. Неявное и явное приведение (преобразование) встроенных типов.	ПК-8
31. Описание и использование методов класса. Перегрузка методов класса.	ПК-8
32. Базовый класс System.Object. Назначение, методы.	ПК-8
33. Операции языка ООП. Перегрузка операций в создаваемых классах.	ПК-8
34. Программирование явного и неявного преобразования пользовательских типов (собственных классов).	ПК-8

## 6. Учебно-методическое и информационное обеспечение дисциплины

1. **Зыков С.В.** Программирование. Объектно-ориентированный подход: учебник и практикум для академического бакалавриата / С. В. Зыков. — М.:Издательство Юрайт, 2017. — 155 с. — ЭБС Юрайт: [Электронный ресурс]. — Адрес доступа: <https://urait.ru/book/programmirovanie-obektno-orientirovannyy-podhod-414203>

2. **Казанский А.А.** Программирование на Visual C# 2013: учебное пособие для прикладного бакалавриата / А. А. Казанский. — М.: Издательство Юрайт, 2017. — 191 с. — ЭБС Юрайт: [Электронный ресурс]. — Адрес доступа: <https://urait.ru/book/obektno-orientirovannyy-analiz-i-programmirovanie-na-visual-basic-2013-470205>

3. **Тузовский А.Ф.** Объектно-ориентированное программирование: учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — М.: Издательство Юрайт, 2017. — 206 с. — ЭБС Юрайт: [Электронный ресурс]. — Адрес доступа: <https://urait.ru/book/obektno-orientirovannoe-programmirovanie-414163>

4. **Хорев П.Б.** Объектно-ориентированное программирование с примерами на ООП: Учебное пособие / Хорев П.Б. — М.: Форум, НИЦ ИНФРА-М, 2016. — 200 с. — ЭБС Znanium.com: [Электронный ресурс]. — Адрес доступа: <http://znanium.com/catalog.php?bookinfo=529350>

### б) дополнительная литература:

1. **Васюткина И.А.** Технология разработки объектно-ориентированных программ на JAVA / Васюткина И.А. — Новосибир. НГТУ, 2012. — 152 с. — ЭБС Znanium.com: [Электронный ресурс]. — Адрес доступа: <http://znanium.com/catalog.php?bookinfo=557111>

2. **Шакин В.Н.** Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.Net / В.Н.Шакин, А.В.Загвоздкина, Г.К.Сосновиков — М.: Форум, ИНФРА-М, 2015. — 400 с. — ЭБС Znanium.com: [Электронный ресурс]. — Адрес доступа: <http://znanium.com/catalog.php?bookinfo=501448>

### в) программное обеспечение и Интернет-ресурсы:

Лицензионное программное обеспечение: Операционная система Windows.

Лицензионное программное обеспечение: Microsoft Office.

### *Профессиональные базы данных и информационные справочные системы*

Российский индекс научного цитирования (РИНЦ), платформа Elibrary: национальная информационно-аналитическая система. Адрес доступа: [http://elibrary.ru/project\\_risc.asp](http://elibrary.ru/project_risc.asp)

ГАРАНТ. Информационно-правовой портал [Электронный ресурс].— Адрес доступа: <http://www.garant.ru>

MathSciNet: информационно-библиографическая и реферативная база данных по математике, в т.ч. прикладной математике и статистике. Электронная версия Mathematical Reviews. Адрес доступа: <http://www.ams.org/mathscinet>

Math-Net.Ru: Общероссийский математический портал. Адрес доступа: <http://www.mathnet.ru/>

### *Свободно распространяемое программное обеспечение:*

программное обеспечение LibreOffice;

программное обеспечение Yandex Browser;

программное обеспечение Paint.NET;

программное обеспечение 1С:

- \* "Бухгалтерия предприятия", редакция 3.0, см. <http://v8.1c.ru/buhv8/> ,
- \* "Управление торговлей", редакция 11.1, см. <http://v8.1c.ru/trade/> ,
- \* "Зарплата и управление персоналом", редакция 3.0, см. <http://v8.1c.ru/hrm/> ,
- \* "Управление небольшой фирмой", редакция 1.5, см. <http://v8.1c.ru/small.biz/> ,
- \* "ERP Управление предприятием 2.0", см. <http://v8.1c.ru/erp/> .
- \* "Бухгалтерия государственного учреждения", редакция 1.0, см. <http://v8.1c.ru/stateacc/> ,
- \* "Зарплата и кадры государственного учреждения", редакция 1.0, <http://v8.1c.ru/statehrm/> .

программное обеспечение PascalABC.NET

#### ***Электронные библиотечные системы и библиотеки:***

Электронная библиотечная система "Лань" <https://e.lanbook.com/>

Электронная библиотечная система "Консультант студента" <http://www.studentlibrary.ru/>

Электронная библиотечная система "Юрайт" <http://www.urait.ru/ebs>

Электронная библиотечная система "Znanium" <http://znanium.com/>

Электронно-библиотечная система Университетская библиотека ONLINE <http://biblioclub.ru/>

Фундаментальная библиотека ННГУ [www.lib.unn.ru/](http://www.lib.unn.ru/)

Сайт библиотеки Арзамасского филиала ННГУ. – Адрес доступа: [lib.arz.unn.ru](http://lib.arz.unn.ru)

Ресурс «Массовые открытые онлайн-курсы Нижегородского университета им. Н.И. Лобачевского» <https://mooc.unn.ru/>

Портал «Современная цифровая образовательная среда Российской Федерации» <https://online.edu.ru/public/promo>

#### **7. Материально-техническое обеспечение дисциплины (модуля)**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: ноутбук, проектор, экран.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет и обеспечены доступом в электронную информационно-образовательную среду ННГУ.

Программа дисциплины **Объектно-ориентированное программирование** составлена в соответствии с образовательным стандартом высшего образования (ОС ННГУ) по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата) (приказ ННГУ от 17.05.2023 года № 06.49-04-0214/23)

Автор(ы):

Старший преподаватель

Сазанов А.А.

Рецензент (ы):

д.т.н., профессор

Ямпурин Н.П.

Кафедра математики, физики и информатики

д.п.н., доцент

Фролов И.В.

Программа одобрена на заседании методической комиссии от 24.05.2023 года, протокол № 5

Председатель МК

к.п.н., доцент

факультета естественных и математических наук

Володин А.М.

П.6. а) СОГЛАСОВАНО:

Заведующий библиотекой

Федосеева Т.А.