

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Рабочая программа дисциплины (модуля)  
**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

Направление и направленность (профиль)  
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2020

Форма обучения  
очная

Владивосток 2023

Рабочая программа дисциплины (модуля) «Объектно-ориентированное программирование» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия (утв. приказом Минобрнауки России от 19.09.2017г. №920) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

*Гриняк В.М., доктор технических наук, профессор, Кафедра информационных технологий и систем, Viktor.Grinyak@vvsu.ru*

Утверждена на заседании кафедры информационных технологий и систем от 31.05.2023 , протокол № 9

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кийкова Е.В.

<b>ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ</b>	
Сертификат	1575633692
Номер транзакции	0000000000B356DA
Владелец	Кийкова Е.В.

## 1 Цель, планируемые результаты обучения по дисциплине (модулю)

Целью изучения дисциплины «Объектно-ориентированное программирование» является теоретическая и практическая подготовка студентов в области разработки программного обеспечения с использованием объектно-ориентированной модели современных языков программирования. Знания, полученные в результате освоения дисциплины, помогут при разработке системных программных компонентов современных информационных и расчетных программ, в проектировании и реализации системных компонентов операционных систем в такой степени, чтобы студенты могли самостоятельно выбирать средства реализации, находить необходимые программные и технологические решения для практически важных системных и предметно-ориентированных задач.

Основные задачи изучения дисциплины:

- приобретение студентами знаний о сущности объектно-ориентированного подхода в программировании;
- ознакомление с технологиями создания новых типов данных в различных языках программирования;
- приобретение практических навыков по использованию средств переопределения операций, обработки исключений.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД1	Знание	информационных технологий и систем
			РД2	Умение	выполнять работы по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем
			РД3	Навык	владения информационными технологиями для внедрения и эксплуатации информационных систем
			РД4	Знание	общих технологий разработки объектов профессиональной деятельности в различных областях
			РД5	Умение	использовать общие технологии разработки объектов профессиональной деятельности в различных областях

			РД6	Навык	владения общими технологиями разработки объектов профессиональной деятельности в различных областях
--	--	--	-----	-------	---

## 2 Место дисциплины (модуля) в структуре ОПОП

Дисциплина относится к обязательной части учебного плана.

## 3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обучения	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудо-емкость (З.Е.)	Объем контактной работы (час)					СРС	Форма аттестации	
					Всего	Аудиторная			Внеаудиторная			
						лек.	прак.	лаб.	ПА			КСР
09.03.04 Программная инженерия	ОФО	Б1.Б	3	3	55	18	36	0	1	0	53	ДЗ

## 4 Структура и содержание дисциплины (модуля)

### 4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Объектно-ориентированный подход в программировании.	РД1, РД5, РД6	2	2	0	10	отчет о выполнении практической работы
2	Классы и объекты.	РД1, РД5, РД6	4	14	0	11	отчет о выполнении практической работы
3	Объектно-ориентированная методология программирования.	РД1, РД5, РД6	2	4	0	5	отчет о выполнении практической работы
4	Наследование, базовый и производный классы.	РД6	2	4	0	5	отчет о выполнении практической работы
5	Функции и классы.	РД2, РД4	2	4	0	5	отчет о выполнении практической работы
6	Шаблоны функций и классов.	РД2, РД3, РД4	2	4	0	5	отчет о выполнении практической работы
7	Потоки и файлы.	РД4	2	2	0	4	отчет о выполнении практической работы
8	Объектный подход к разработке ПО.	РД2, РД3, РД4	2	2	0	8	отчет о выполнении практической работы
<b>Итого по таблице</b>			<b>18</b>	<b>36</b>	<b>0</b>	<b>53</b>	

## 4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

### *Тема 1 Объектно-ориентированный подход в программировании.*

Содержание темы: Сущность объектно-ориентированного подхода в программировании. Цикл разработки программного обеспечения (ПО), назначение и содержание этапов. Роль анализа в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного анализа. Язык C++. Язык Java.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

### *Тема 2 Классы и объекты.*

Содержание темы: Отношения, основные типы отношений. Язык UML. Основные средства анализа и моделирования предметной области в языке UML. Статические данные. Конструктор, деструктор. Операции new и delete.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

### *Тема 3 Объектно-ориентированная методология программирования.*

Содержание темы: Технология применения объектно-ориентированных языков, их классификация и архитектура. Перегрузка операций. Преобразование типов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

### *Тема 4 Наследование, базовый и производный классы.*

Содержание темы: Простое и сложное наследование. Абстракция данных, наследование и полиморфизм.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

### *Тема 5 Функции и классы.*

Содержание темы: Виртуальные функции. Дружественные функции. Дружественные классы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

### *Тема 6 Шаблоны функций и классов.*

Содержание темы: Шаблоны функций. Шаблоны классов. Исключения. Стандартная библиотека шаблонов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по

практической работе, подготовка к промежуточной аттестации.

#### *Тема 7 Потoki и файлы.*

Содержание темы: Стандартная библиотека классов для управления потоками. Методы и средства организации и программирования интерфейса.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

#### *Тема 8 Объектный подход к разработке ПО.*

Содержание темы: Стандарты кодирования и их проекция на объектно-ориентированную модель программирования. Объектный подход к разработке ПО для распределенных систем.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по практической работе, подготовка к промежуточной аттестации.

## **5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)**

### **5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы**

В ходе изучения дисциплины «Объектно-ориентированное программирование» студенты могут посещать аудиторные занятия (лекции, практические занятия, консультации).

Особое место в овладении частью тем данной дисциплины может отводиться самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а второстепенные и более легкие вопросы, а также вопросы, специфичные для ОПОП, могут быть изучены студентами самостоятельно.

В соответствии с учебным планом процесс изучения дисциплины может предусматривать проведение лекций, практических занятий, консультаций, а также самостоятельную работу студентов. Обязательным является проведение практических занятий в специализированных компьютерных аудиториях, оснащенных подключенными к центральному серверу терминалами или персональными компьютерами.

В рамках общего объема часов, отведенных для изучения дисциплины, предусматривается выполнение следующих видов самостоятельных работ студентов: изучение теоретического материала при подготовке к защите практических работ, итоговое повторение теоретического материала.

Для закрепления материала и приобретения навыков самостоятельного написания объектно-ориентированных программ рекомендуется выполнение следующих задач:

1. Написать шаблон класса для работы с очередью FIFO. Определить функции включения и исключения элементов. Добавить механизм обработки исключений при превышении размера очереди и при попытке удалить данные из пустой очереди. Это можно сделать, добавив элемент данных – счетчик текущего числа элементов. Исключения генерируются, если счетчик превысил размер массива или если он стал меньше 0.

2. Опишите классы PointXY и PointPolar, объекты которых задают декартовы и полярные координаты точки на плоскости. Перегрузите для этих классов операции сложения, вычитания и умножения как скалярного произведения, так, чтобы в них могли участвовать

объекты как одного, так и обоих классов. Кроме того, задайте функцию преобразования одного класса в другой (для обоих классов).

3. Определить класс `stack`, который позволяет реализовать структуру данных типа стек для хранения целых чисел. Конструктор класса должен содержать параметр, определяющий размер стека. Определить для класса функции `pop()` (достать из стека), `push()` (положить в стек) и операцию определения текущего размера стека. Функции должны осуществлять проверку на выход за пределы стека. Определить класс `fifo`, реализующий структуру данных типа очередь для хранения целых чисел, породив его от класса `stack`, добавив нужные поля и переопределив функции `pop()`, `push()` и определение текущего размера очереди. Продемонстрировать работу.

4. В любой визуальной среде создать класс `Figure` с виртуальным методом `draw()`, осуществляющим прорисовку объекта на визуальном компоненте. Создать производные классы: `Rectangle` (прямоугольник), `Circle` (круг), `Triangle` (треугольник). Описать в производных классах функции `draw()` для каждой из фигур, продемонстрировать работу механизма виртуальных функций.

Для выполнения практических работ студент должен предварительно самостоятельно освоить теоретический материал соответствующих тем.

Дисциплина «Объектно-ориентированное программирование» ориентирована на изучение относительно устоявшейся к настоящему времени области программирования. По тематике дисциплины выпущено большое количество печатных изданий. В рекомендованных книгах содержатся основные принципы построения и использования языка C++, готовые полномасштабные приложения, небольшие примеры, поясняющие теорию, и множество полезных иллюстраций. Книги содержат множество упражнений, позволяющих проверить знания по всем темам.

## **5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

## **6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)**

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

## **7 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### ***0.1 Основная литература***

### ***0.2 Дополнительная литература***

### ***0.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):***

Отсутствуют

**8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения**



МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Фонд оценочных средств  
для проведения текущего контроля  
и промежуточной аттестации по дисциплине (модулю)

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

Направление и направленность (профиль)

09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2020

Форма обучения  
очная

Владивосток 2023

## 1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

## 2 Показатели оценивания планируемых результатов обучения

**Компетенция ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»**

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код результата	Тип результата	Результат	
ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД1	Знание	информационных технологий и систем	Сформировавшееся знание информационных технологий и систем
	РД2	Умение	выполнять работы по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	Сформировавшееся умение выполнять работы по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем
	РД3	Навык	владения информационными технологиями для внедрения и эксплуатации информационных систем	Сформировавшееся владение информационными технологиями для внедрения и эксплуатации информационных систем
	РД4	Знание	общих технологий разработки и объектов профессиональной деятельности в различных областях	Сформировавшееся знание общих технологий разработки объектов профессиональной деятельности в различных областях

	Р Д 5	У м е н е	использовать общие технологии разработки объектов профессиональной деятельности в различных областях	Сформировавшееся умение и использовать общие технологии и разработки объектов профессиональной деятельности в различных областях
	Р Д 6	Н а в ы к	владения общими технологиями разработки объектов профессиональной деятельности в различных областях	Сформировавшееся владение общими технологиями разработки объектов профессиональной деятельности в различных областях

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

### 3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения	Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС		
		Текущий контроль	Промежуточная аттестация	
Очная форма обучения				
РД1	Знание : информационных технологий и систем	1.1. Объектно-ориентированный подход в программировании.	Практическая работа	Тест
		1.2. Классы и объекты.	Практическая работа	Тест
		1.3. Объектно-ориентированная методология программирования.	Практическая работа	Тест
РД2	Умение : выполнять работы по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	1.5. Функции и классы.	Практическая работа	Тест
		1.6. Шаблоны функций и классов.	Практическая работа	Тест
		1.8. Объектный подход к разработке ПО.	Практическая работа	Тест
РД3	Навык : владения информационными технологиями для внедрения и эксплуатации информационных систем	1.6. Шаблоны функций и классов.	Практическая работа	Тест
		1.8. Объектный подход к разработке ПО.	Практическая работа	Тест
РД4	Знание : общих технологий разработки объектов профессиональной деятельности в различных областях	1.5. Функции и классы.	Практическая работа	Тест
		1.6. Шаблоны функций и классов.	Практическая работа	Тест
		1.7. Потoki и файлы.	Практическая работа	Тест
		1.8. Объектный подход к разработке ПО.	Практическая работа	Тест
РД5	Умение : использовать общие технологии разработки объектов профессии	1.1. Объектно-ориентированный подход в программировании.	Практическая работа	Тест

	ональной деятельности в различных областях	1.2. Классы и объекты.	Практическая работа	Тест
		1.3. Объектно-ориентированная методология программирования.	Практическая работа	Тест
РДб	Навык : владения общими технологиями разработки объектов профессиональной деятельности в различных областях	1.1. Объектно-ориентированный подход в программировании.	Практическая работа	Тест
		1.2. Классы и объекты.	Практическая работа	Тест
		1.3. Объектно-ориентированная методология программирования.	Практическая работа	Тест
		1.4. Наследование, базовый и производный классы.	Практическая работа	Тест

#### 4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Тестовые задания	Практические работы	Итого
Лекции	10		10
Практические занятия		60	60
Самостоятельная работа		20	20
Промежуточная аттестация	10		10
Итого	20	80	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями и умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.

от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.
------------	--------------------------------------	---

## 5 Примерные оценочные средства

### 5.1 Примеры тестовых заданий

#### 5.1 Пример тестовых заданий

1) Which of the following denote stream classes in C++?

- a) ios
- b) fstream
- c) ostream
- d) All the Above

2) cin in C++ program uses the operator

- a) >>
- b)
- c)
- d) >

3) Which of the following denote types of polymorphism in C++?

- a) Virtual function
- b) Function overloading
- c) Operator Overloading
- d) All the Above

4) A condition that must be true on exit from a member function if called as

- a) Precondition
- b) Post-condition
- c) Both A and B
- d) None of the Above

5) In a C++ program each statement is ended with the character

- a) .
- b) ;
- c) \*
- d) :

6) The header file that must be included while using cout function in a C++ program is

- a) conio.h
- b) math.h
- c) iostream.h
- d) None of the Above

7) The variable that contains address of another variable is called as

- a) Pointer
- b) arrays
- c) unions
- d) None of the Above

8) Which of the following language given below uses the concepts of OOPS?

- a) C++
- b) C#
- c) Java
- d) All the Above

9) The output of operation 20%3 is

- a) 6
- b) 2
- c) 1

- d) 4
- 10) In Late binding the function calls gets resolved during
- Compile Time
  - Run Time
  - Both A and B
  - None of the Above
- 11) The class that in C++ for file input is
- ifstream
  - ofstream
  - Both A and B
  - None of the Above
- 12) The operator that denotes address of a variable in C++ program is
- \*
  - %
  - \$
  - &
- 13) The notation of ternary operator is
- &
  - ?:
  -

*Краткие методические указания*

Промежуточный тест проводится в электронной форме во время последнего в учебном периоде практического занятия. Тест состоит из 20 тестовых заданий. На выполнение теста отводится 20 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем.

*Шкала оценки*

Оценка	Баллы	Описание
5	19–20	Процент правильных ответов от 95% до 100%
4	16–18	Процент правильных ответов от 80 до 94%
3	13–15	Процент правильных ответов от 65 до 79%
2	9–12	Процент правильных ответов от 45 до 64%

**5.2 Примеры заданий для выполнения практических работ**

Тема 1. Создать структуру с именем `time`. Три ее поля, имеющие тип `int`, будут называться `hours`, `minutes` и `seconds`. Написать программу, которая просит пользователя ввести время в формате часы, минуты, секунды. Программа должна хранить время в структурной переменной типа `time` и выводить количество секунд в введенном времени.

Тема 2. Создать перечисление с именем `pets` и значениями `dog`, `cat`, `rat`, `fish`, `bird`. Создать структуру с именем `animal`. Определить ее поля: `name` как массив из 20 символов типа `char`, `type` как типа `pets` и `age` типа `float`, хранящие, соответственно, имя животного, его разновидность и возраст. Написать программу, которая создает 4 переменные типа `animal` и просит пользователя ввести значения разновидности, имени и возраста всех 4 животных. Причем разновидность животного вводится по первому символу, набранному на клавиатуре (т.е. если нажата `d`, то в соответствующее поле вводится значение `dog`). В качестве результата программа должна выдать список всех имеющихся животных со всеми значениями полей

Тема 3. Создать класс с именем `Time`, содержащий три поля типа `int`, предназначенных для хранения часов, минут и секунд. Один из конструкторов класса должен инициализировать поля нулевыми значениями, а другой – заданным набором значений. Создать метод класса, который будет выводить значения полей на экран, в формате `23:59:59` и метод, складывающий значения двух объектов типа `Time`, передаваемых ему в качестве

аргументов. Продемонстрировать работу класса.

Тема 4. Создать класс с именем `fraction`, содержащий два поля типа `int` – числитель и знаменатель обыкновенной дроби. Конструктор класса должен инициализировать их заданным набором значений. Создать метод класса, который будет выводить дробь на экран в формате  $x / y$ , метод, складывающий две дроби, переданные ему в параметрах и метод, умножающий две дроби, переданные ему в параметрах.

Тема 5. Описать класс `fraction`, у которого поля  $x$  и  $y$  задают числитель и знаменатель обыкновенной дроби. Перегрузить для этого класса арифметические операции сложения, вычитания, умножения и деления так, чтобы они могли оперировать как с объектами класса, так и с числами (то есть выполнять, например, не только действие  $3/4 + 2/5$ , но и  $1/2 + 4$  или  $2 * 5/6$ ). Также перегрузить операции сравнения `==` и `!=`. Продемонстрировать работу класса.

Тема 6. Определить класс `vector2` как вектор на плоскости с данными  $x$  и  $y$ . Определить для него операции сложения, вычитания и скалярного произведения, определить функцию присваивания значения координатам вектора и функцию вывода значений на консоль. Определить класс `vector3` как вектор в пространстве, породив его от класса `vector2`. Переопределить для него функции и операции. Продемонстрировать работу класса.

Тема 7. Создать абстрактный класс `Figure` с виртуальными методами вычисления площади и периметра. Создать производные классы: `Rectangle` (прямоугольник), `Circle` (круг), `Triangle` (треугольник). Описать в производных классах функции вычисления периметра и площади, продемонстрировать работу механизма виртуальных функций.

Тема 8. Создать шаблон функции, возвращающей среднее арифметическое всех элементов массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, возвращающей значение максимального элемента массива. Аргументами функции должны быть имя и размер массива (типа `int`). Создать шаблон функции, обменивающей местами значения двух передаваемых ей по ссылке аргументов. Создать шаблон функции, осуществляющей сортировку данных массива. Аргументами функции должны быть имя и размер массива (типа `int`). Продемонстрировать работу шаблонов на данных различных типов, в том числе, на данных вновь созданного класса – вектор на плоскости, определив для него операции сравнения.

#### *Краткие методические указания*

На выполнение одной практической работы отводится не более одного двухчасового занятия (не включая затраты времени на проведение промежуточного теста на последнем в учебном периоде практическом занятии). После выполнения каждой практической работы студент должен представить отчет о ее выполнении, а также, по указаниям преподавателя, выполнить дополнительные практические задания по теме работы.

#### *Шкала оценки*

Оценка	Баллы	Описание
5	73–80	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практически все задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	61–72	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	49–60	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	33–48	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.