

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Новосибирский национальный исследовательский государственный университет»  
(Новосибирский государственный университет, НГУ)

**Физический факультет  
Кафедра физико-технической информатики**



д.ф.-м.н. \_\_\_\_\_

УТВЕРЖДАЮ  
Декан ФФ  
В. Е. Блинов  
« 29 » \_\_\_\_\_ 2024 г.

**Рабочая программа дисциплины**

**ВВЕДЕНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

направление подготовки: **03.03.02 Физика**  
направленность (профиль): **Физическая информатика**

Форма обучения  
**Очная**

Семестр	Общий объем	Виды учебных занятий (в часах)				Промежуточная аттестация (в часах)				
		Контактная работа обучающихся с преподавателем			Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем			
		Лекции	Практические занятия	Лабораторные занятия			Консультации	Зачет	Дифференцированный зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11
4	108	32	32		20	18	4			2
Всего 108 часов / 3 зачетные единицы, из них: - контактная работа 70 часов										
Компетенции ОПК-3										

Руководитель программы  
д.ф.-м.н., проф.

С. В. Цыбуля

Новосибирск, 2024

Содержание:

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы .....	3
2. Место дисциплины в структуре образовательной программы.....	3
3. Трудоёмкость дисциплины в зачётных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу .....	3
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведённого на них количества академических часов и видов учебных занятий. ....	4
5. Перечень учебной литературы .....	6
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся .....	7
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.....	7
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.....	7
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине. ....	7
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине .....	8

## 1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

Цель дисциплины – ознакомить студентов с основными концепциями и принципами объектно-ориентированного программирования, а также с языком программирования C++ и его особенностями. Студенты также получают знания о наследовании, полиморфизме, исключениях и использовании стандартной библиотеки шаблонов (STL). Кроме того, курс познакомит студентов с принципами SOLID и стандартами C++, чтобы они могли писать современный и эффективный код.

Дисциплина нацелена на формирование у выпускника следующей компетенции:

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
<b>ОПК-3.</b> Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	<b>ОПК - 3.1.</b> Применяет различные источники информации для решения задач профессиональной сферы деятельности. <b>ОПК – 3.2.</b> Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач. <b>ОПК – 3.3.</b> Применяет методологию поиска научной и технической информации в сети Интернет и специализированных базах данных.	<b>Знать</b> основные понятия и принципы объектно-ориентированного программирования. <b>Уметь</b> применять на практике современные методы и возможности языка C++ в разработке программного обеспечения; применять на практике принципы SOLID. <b>Использовать</b> основные современные методы расчета объекта научного исследования; навыки разработки кода на C++ для решения определенных физических и математических задач.

## 2. Место дисциплины в структуре образовательной программы

Дисциплина «Введение в объектно-ориентированное программирование» читается для студентов в весеннем семестре 2 курса, обучающихся по направлению подготовки **03.03.02 Физика профиль (направленность) «Физическая информатика»**.

Изучение данной дисциплины базируется на дисциплинах «Основы программного конструирования» и «Структурное программирование». После освоения данной дисциплины изучаются курсы «Практическое объектно-ориентированное программирование» и «Основы компьютерных сетей».

## 3. Трудоемкость дисциплины в зачётных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу

Дисциплина рассчитана на **один семестр (4-й)**. Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, практические занятия, самостоятельная работа студента, консультации, экзамен.

Семестр	Общий объем	Виды учебных занятий (в часах)				Промежуточная аттестация (в часах)				
		Контактная работа обучающихся с преподавателем			Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем			
		Лекции	Практические занятия	Лабораторные занятия			Консультации	Зачет	Дифференцированный зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11
4	108	32	32		20	18	4			2
Всего 108 часа / 3 зачетные единицы, из них: - контактная работа 70 часов										
Компетенции ОПК-3										

Реализация дисциплины предусматривает практическую подготовку при проведении следующих видов занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью: лекции, практические занятия, консультации, самостоятельная работа студента, экзамен.

Программой дисциплины предусмотрены следующие виды контроля:

- Текущий контроль успеваемости: опрос студентов в начале каждого занятия, решение задач;
- Промежуточная аттестация: экзамен.

Общая трудоемкость рабочей программы дисциплины составляет **108** академических часа / **3** зачетные единицы:

- занятия лекционного типа – 32 часа;
- практические занятия – 32 часа;
- самостоятельная работа обучающегося в течение семестра, не включая период сессии – 20 часов;
- самостоятельная подготовка к промежуточной аттестации – 18 часов;
- консультации – 4 часа;
- промежуточная аттестация (экзамен) – 2 часа.

Объем контактной работы обучающегося с преподавателем (занятия лекционного типа, практические занятия, экзамен) составляет 70 часов.

#### **4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведённого на них количества академических часов и видов учебных занятий.**

Дисциплина представляет собой семестровый курс, читаемый на 2-ом курсе физического факультета НГУ в четвертом семестре. Общая трудоемкость дисциплины составляет 3 зачетные единицы, 108 академических часа.

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в часах)				Промежуточная аттестация (в часах)
			Всего	Аудиторные часы		Сам. работа в течение семестра (не включая период сессии)	
				Лекции	Практические занятия		
1	2	3	4	5	6	7	8
1	Введение в ООП	1	5	2	2	1	
2	Введение в С++	2	5	2	2	1	
3	Классы С++	3-5	15	6	6	3	
4	Наследование	6-8	15	6	6	3	
5	Исключения	9	5	2	2	1	
6	Стандартная библиотека шаблонов	10-12	15	6	6	3	
7	Принципы SOLID	13-15	15	6	6	3	
8	Стандарты С++	16	3	2		1	
9	Подготовка к защите итогового проекта.		4			4	
10	Защита итогового проекта.	16	2		2		
11	Групповая консультация		4				4
12	Самостоятельная подготовка обучающегося к промежуточной аттестации		18				18
13	Экзамен		2				2
	<b>Всего за 5 семестр:</b>		<b>108</b>	<b>32</b>	<b>32</b>	<b>20</b>	<b>24</b>

### Программа и основное содержание лекций (32 часа)

1	<b>Введение в ООП.</b> Вводная лекция: обзор курса, постановка проблемы сложности разработки ПО. Обзор парадигм программирования.
2	<b>Введение в C++.</b> Операторы new/delete, ссылки, перегрузка функций, пространства имён.
3-5	<b>Классы C++.</b> Обзор классов C++. Указатель this. Конструкторы/деструкторы. Использование friend. Перегрузка операторов.
6-8	<b>Наследование.</b> Обзор наследования: public, private, protected. Полиморфизм. Виртуальные функции. Статическое и динамическое связывание. Множественное наследование.
9	<b>Исключения.</b> Использование исключений, стандартная иерархия исключений. Безопасность исключений.
10-12	<b>Стандартная библиотека шаблонов.</b> Обзор STL. Последовательные и ассоциативные контейнеры, алгоритмы STL. Итераторы
13-15	<b>Принципы SOLID.</b> Обзор принципов SOLID и их использования. Принцип единой ответственности. Принцип открытости-закрытости. Принцип подстановки Барбары Лисков. Принцип разделения интерфейса. Принцип инверсии зависимостей.
16	<b>Стандарты C++.</b> Обзор стандартов C++ от C++98 до C++23. Использование современного C++.

### Программа практических занятий (32 часа)

Содержание занятия	Объем, час
Блок 1. Реализация программ на C++ с использованием классов.	6
Блок 2. Применение наследования в программировании.	8
Блок 3. Работа с STL (стандартной библиотекой шаблонов)	8
Блок 4. Применение принципов SOLID при разработке ПО.	8
Блок 5. Защита проектов.	2
<b>Итого:</b>	<b>32</b>

### Самостоятельная работа студентов (38 часов)

Перечень занятий на СРС	Объем, час
Подготовка к практическим занятиям.	16
Подготовка к защите итогового проекта.	4
Подготовка к экзамену.	18

### 5. Перечень учебной литературы

1. Мейерс, Скотт. Наиболее эффективное использование C++. 35 новых рекомендаций по улучшению ваших программ и проектов: [пер. с англ.] / Скотт Мейерс. Москва : ДМК Пресс, 2014. 296 с. : ил. ; 20 см (Для программистов) ISBN 978-5-94074-990-5 (1 экз.)
2. Страуструп Бьерн. Язык программирования C++ / Бьерн Страуструп ; пер. с англ. под ред. Н.Н. Мартынова. — Спец. изд. — Москва : Бином, 2012. — 1135 с. : ил., ISBN 978-5-7989-0425-9 (3 экз.)
3. Керниган, Брайан В. Язык программирования C / Брайан Керниган, Деннис Ритчи ; [пер. с англ. и ред. В.Л. Бродового] 2-е изд., перераб. и доп. Москва [и др.] : Вильямс, 2009. 289 с. ; 21 см. ISBN 978-5-8459-0891-9 (1 экз.)
4. Керниган Брайан В. Практика программирования / пер. с англ. под ред. И.В. Романовского. – Москва; Санкт-Петербург : Бином : Невский Диалект, 2001. — 380 с. : ил., ISBN 5-7940-0058-9 ISBN 0-201-61586-X (4 экз.)

## **6. Перечень учебно-методических материалов по самостоятельной работе обучающихся**

1. М.И. Болски. Язык программирования Си: [Справочник] / Пер. с англ. С.В. Денисенко. — М. : Радио и связь, 1988. — 96 с.
2. Мейерс Скотт. Эффективное использование C++: 50 рекомендаций по улучшению ваших программ и проектов [пер. с англ.] – М.; СПб.: ДМК Пресс : ПИТЕР, 2006. — 235 с. : ил.

## **7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.**

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет.

### **7.1 Современные профессиональные базы данных**

Не используются.

### **7.2. Информационные справочные системы**

Не используются.

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.**

### **8.1 Перечень программного обеспечения**

Для обеспечения реализации дисциплины используется стандартный комплект регулярно обновляемого открытого и свободного программного обеспечения, включающий ОС семейства Linux, среды разработки (например, Microsoft Visual Studio Code), компилятор GNU C++, инструмент сборки CMake и распределённую систему управления версиями git.

Использование специализированного программного обеспечения для изучения дисциплины не требуется.

### **8.2 Информационные справочные системы**

Не используются.

## **9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.**

Для реализации дисциплины используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, практических занятий и промежуточной аттестации.

2. Помещения для самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

## **10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине**

Перечень результатов обучения по дисциплине «Введение в объектно-ориентированное программирование» и индикаторов их достижения представлен в виде знаний, умений и владений в разделе 1.

### **10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине**

#### **Текущий контроль**

Текущий контроль успеваемости обучающегося происходит в течение всего семестра. В процессе приема задач, заполнения ведомостей контрольных недель преподаватель выставляет оценки, отражающие уровень освоения пройденного материала обучающимся.

Примеры задач приведены в п. 10.3.

Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования C++ и демонстрации ее работы в соответствии с требованиями задания, прохождение автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

Для сдачи итогового проекта по курсу необходимо придумать тему проекта и согласовать её с преподавателем. Во время самостоятельной работы написать код на языке программирования C++, соответствующий требованиям задания. По завершении работы подразумевается защита кода проекта на занятии и ответ на вопросы по нему, чтобы подтвердить авторство и продемонстрировать понимание принципов объектно-ориентированного программирования на языке C++.

#### **Промежуточная аттестация**

Освоение компетенций оценивается согласно шкале оценки уровня сформированности компетенции. Положительная оценка по дисциплине выставляется в том случае, если заявленная компетенция ОПК-3 сформирована не ниже порогового уровня в части, относящейся к формированию способности использовать специализированные знания в области использования языков описания аппаратуры в профессиональной деятельности.

Окончательная оценка работы студента в течение семестра происходит на экзамене. Экзамен проводится в конце семестра в сессию в устной форме. Вопросы подбираются таким образом, чтобы проверить уровень сформированности компетенции ОПК-3.

Вывод об уровне сформированности компетенций принимается преподавателем. Каждый вопрос билета оценивается от 0 до 5 баллов. Положительная оценка ставится, когда все компетенции освоены не ниже порогового уровня. Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Итоги промежуточной аттестации (экзамен) оцениваются по пятибалльной шкале:

«отлично» - сдано не менее 4 задач в течение семестра, защищён итоговый проект, дан развернутый ответ на вопросы билета, даны правильные ответы на дополнительные вопросы, имеющие принципиальное значение для данной дисциплины;

«хорошо»- сдано не менее 4 задач в течение семестра, защищён итоговый проект, дан полный ответ на вопросы билета (допускается несколько несущественных ошибок), даны правильные ответы на дополнительные вопросы, имеющие принципиальное значение для данной дисциплины;

«удовлетворительно» - сдано не менее 4 задач в течение семестра, защищён итоговый проект, дан ответ на один вопрос билета, даны ответы на дополнительные вопросы, имеющие принципиальное значение для данной дисциплины;

«неудовлетворительно»- сдано менее 4 задач в течение семестра или не защищён итоговый проект.

### Соответствие индикаторов и результатов освоения дисциплины

Таблица 10.1

Индикатор	Результат обучения по дисциплине	Оценочные средства
<b>ОПК - 3.1.</b> Применяет различные источники информации для решения задач профессиональной сферы деятельности.	<b>Знать</b> основные понятия и принципы объектно-ориентированного программирования.	Опрос по каждой лекции, экзамен.
<b>ОПК – 3.2.</b> Применяет основные приемы, возможности и правила работы со стандартными и специализированными программными продуктами при решении профессиональных задач. <b>ОПК – 3.3.</b> Применяет методологию поиска научной и технической информации в сети Интернет и специализированных базах данных.	<b>Уметь</b> применять на практике современные методы и возможности языка C++ в разработке программного обеспечения; применять на практике принципы SOLID. <b>Использовать</b> основные современные методы расчета объекта научного исследования; навыки разработки кода на C++ для решения определенных физических и математических задач.	Опрос по каждой лекции, экзамен.

### 10.2 Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Введение в объектно-ориентированное программирование».

Критерии оценивания результатов обучения	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Уровень освоения компетенции			
		Не сформирован (0 баллов)	Пороговый уровень (3 балла)	Базовый уровень (4 балла)	Продвинутый уровень (5 баллов)
1	2	3	4	5	6

Полнота знаний	ОПК-3.1	Уровень знаний ниже минимальных требований. Имеют место грубые ошибки.	Минимально допустимый уровень знаний. Допускается значительное количество негрубых ошибок.	Уровень знаний соответствует программе подготовки по темам/разделам дисциплины. Допускается несколько негрубых/несущественных ошибок. Не отвечает на дополнительные вопросы.	Уровень знаний соответствует программе подготовки по темам/разделам дисциплины. Свободно и аргументированно отвечает на дополнительные вопросы.
Наличие умений	ОПК-3.2	Отсутствие минимальных умений. Не умеет решать стандартные задачи. Имеют место грубые ошибки.	Продемонстрированы частично основные умения. Решены типовые задачи. Допущены негрубые ошибки.	Продемонстрированы все основные умения. Решены все основные задания с негрубыми ошибками или с недочетами.	Продемонстрированы все основные умения. Решены все основные задания в полном объеме без недочетов и ошибок.
Наличие навыков (владение опытом)	ОПК-3.3	Отсутствие владения материалом по темам/разделам дисциплины. Нет навыков в решении стандартных задач. Наличие грубых ошибок.	Имеется минимальный набор навыков при решении стандартных задач с некоторыми недочетами.	Имеется базовый набор навыков при решении стандартных задач с некоторыми недочетами.	Имеется базовый набор навыков при решении стандартных задач без ошибок и недочетов. Продемонстрированы знания по решению нестандартных задач.

### 10.3 Типовые контрольные задания и материалы, необходимые для оценки результатов обучения

#### Примеры задач для решения.

- Реализация программ на C++ с использованием классов.
- Применение наследования в программировании.
- Работа с STL (стандартной библиотекой шаблонов)
- Применение принципов SOLID при разработке ПО.

#### Тематика вопросов к экзамену:

- Классы в C++. Указатель this. Использование friend. Перегрузка операторов.
- Классы в C++. Конструкторы/деструкторы.
- Наследование в C++. Полиморфизм. Public, private, protected наследование.
- Наследование в C++. Виртуальные функции. Абстрактные классы.
- Наследование в C++. Множественное наследование.
- Использование исключений, стандартная иерархия исключений. Безопасность исключений.
- STL. Последовательные контейнеры. Алгоритмы STL.
- STL. Ассоциативные контейнеры. Алгоритмы STL.
- STL. Итераторы. Алгоритмы STL.
- Принципы SOLID. Принцип единой ответственности.
- Принципы SOLID. Принцип открытости-закрытости.
- Принципы SOLID. Принцип подстановки Барбары Лисков.

- Принципы SOLID. Принцип разделения интерфейса.
- Принципы SOLID. Принцип инверсии зависимостей.
- Стандарты C++. C++98, C++11, C++14

Оценочные материалы по промежуточной аттестации, предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям СУОС, хранятся на кафедре-разработчике РПД в печатном и электронном виде.

**Лист актуализации рабочей программы  
по дисциплине «Введение в объектно-ориентированное программирование»  
по направлению подготовки 03.03.02 Физика  
Профиль «Физическая информатика»**

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Учёного совета ФФ НГУ	Подпись ответственного