

**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**

академик РАН



УТВЕРЖДАЮ  
Декан ФФ  
А. Е. Бондарь  
2020 г.

**Рабочая программа дисциплины**

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ 1**

направление подготовки: **03.03.02 Физика, Курс 2, семестр 3**

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
3	180	32		64	82				2	
Всего 180 часов / 5 зачётных единиц, из них: - контактная работа 98 часов - в интерактивных формах 64 часа										
Компетенции ПК-1, ПК-2										

Разработчик:  
доцент

*Рылов*

В. Ю. Рылов

Заведующий кафедрой ФТИ ФФ НГУ  
к.ф.-м.н.

*Кроковный*

П. П. Кроковный

Ответственный за образовательную программу  
д.ф.-м.н., проф.

*Цыбуля*

С. В. Цыбуля

Новосибирск, 2020

<b>Содержание</b>	
<b>Аннотация</b> .....	3
1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы. ....	5
2. Место дисциплины в структуре образовательной программы. ....	5
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. ....	6
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведённого на них количества академических часов и видов учебных занятий. ....	7
5. Перечень учебной литературы. ....	14
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся. ....	15
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины. ....	15
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине. ....	15
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине. ....	15
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине. ....	16

**Аннотация**  
**к рабочей программе дисциплины**  
**«Объектно-ориентированное программирование 1»**  
Направление: **03.03.02 Физика**  
**Направленность (профиль): Физическая информатика**

Программа курса «Объектно-ориентированное программирование 1» составлена в соответствии с требованиями СУОС по направлению подготовки **03.03.02 Физика, направленность «Физическая информатика»**, а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ. Дисциплина реализуется на физическом факультете Федерального государственного автономного образовательного учреждения высшего профессионального образования Новосибирский национальный исследовательский государственный университет (НГУ) кафедрой физико-технической информатики. Дисциплина изучается студентами второго курса физического факультета в осеннем семестре в качестве одной из дисциплин по выбору вариативной части образовательной программы.

Дисциплина «Объектно-ориентированное программирование» имеет своей целью изучение основ классической теории объектно-ориентированного программирования, в том числе:

- Пути эволюции технологий программирования от алгоритмического к ООП
- Основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Иерархия, Модульность, Типизация, Параллелизм, Сохраняемость)
- Понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG
- Изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL
- Выработку практических навыков проектирования и реализации объектно-ориентированных программ на языке программирования C++.

Дисциплина нацелена на формирование у выпускника профессиональных компетенций:

**ПК-1** – способность использовать специализированные знания в области физики для освоения профильных физических дисциплин.

**ПК-2** – способность проводить научные исследования в избранной области экспериментальных и (или) теоретических физических исследований с помощью современной приборной базы (в том числе сложного физического оборудования) и информационных технологий с учетом отечественного и зарубежного опыта.

В результате освоения дисциплины обучающийся должен:

- **Знать:** знать особенности построения объектно-ориентированных программных систем на C++.
- **Уметь:** использовать инструментальные средства языка C++ и стандартной библиотеки.
- **Владеть** библиотекой STL.

Курс рассчитан на один семестр. Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные занятия, самостоятельная работа студента и её контроль преподавателями с помощью заданий, дифференцированный зачёт.

Программой дисциплины предусмотрены следующие виды контроля:

Текущий контроль: задания для самостоятельного решения.

Промежуточная аттестация: дифференцированный зачёт.

Общая трудоемкость рабочей программы дисциплины составляет **180** академических часов / **5** зачетных единиц.

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

Дисциплина «Объектно-ориентированное программирование» имеет своей целью изучение основ классической теории объектно-ориентированного программирования, в том числе:

- Пути эволюции технологий программирования от алгоритмического к ООП
- Основных принципов объектно-ориентированного построения программных систем (Абстракция, Инкапсуляция, Иерархия, Модульность, Типизация, Параллелизм, Сохраняемость)
- Понятий классов, объектов, взаимоотношений между ними, а также многоуровневой модели OMG
- Изучение средств объектно-ориентированного и обобщенного программирования языка C++, средств стандартной библиотеки STL
- Выработку практических навыков проектирования и реализации объектно-ориентированных программ на языке программирования C++.

Дисциплина нацелена на формирование у выпускника профессиональных компетенций:

**ПК-1** – способность использовать специализированные знания в области физики для освоения профильных физических дисциплин.

**ПК-2** – способность проводить научные исследования в избранной области экспериментальных и (или) теоретических физических исследований с помощью современной приборной базы (в том числе сложного физического оборудования) и информационных технологий с учетом отечественного и зарубежного опыта.

В результате освоения дисциплины обучающийся должен:

- **Знать:** знать особенности построения объектно-ориентированных программных систем на C++.
- **Уметь:** использовать инструментальные средства языка C++ и стандартной библиотеки.
- **Владеть** библиотекой STL.

## 2. Место дисциплины в структуре образовательной программы.

Учебный курс «Объектно-ориентированное программирование 1» относится к вариативной части программы дисциплин бакалавриата.

Предварительными требованиями к студентам являются:

- Знание одного из классических процедурно-ориентированных языков, предпочтительно языка C
- Знания в области алгоритмической декомпозиции, основных структур данных и технологий работы с ним
- Знание основ теории множеств

Содержание дисциплины охватывает круг вопросов, связанных с теорией объектно-ориентированного программирования и особенностями ее поддержки и реализации основных принципов в языке программирования C++ и его стандартной библиотеке

По окончании курса студенты получают следующие знания и навыки:

- Знание основ технологии объектно-ориентированной декомпозиции программных систем, базовых шаблонов проектирования (Наблюдатель, Итератор, Одиночка, Фабрика,

Заместитель), отношений между классами и основ UML (диаграммы классов и последовательностей).

- Знание особенностей построения объектно-ориентированных программных систем на C++.
- Основные инструментальные средства языка C++ и стандартной библиотеки
- Знания и навыки использования системы и библиотеки автоматизированного тестирования Google Test Framework для C++
- Навыки использования среды Microsoft Visual Studio C++

**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.**

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
3	180	32		64	82				2	
Всего 180 часов / 5 зачётных единиц, из них: - контактная работа 98 часов - в интерактивных формах 64 часа										
Компетенции ПК-1, ПК-2										

Реализация дисциплины предусматривает практическую подготовку при проведении следующих видов занятий, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью: лекции, лабораторные занятия, самостоятельная работа студента и её контроль преподавателями с помощью заданий, дифференцированный зачёт.

Программой дисциплины предусмотрены следующие виды контроля:

- текущий контроль успеваемости: задания для самостоятельного решения;
- промежуточная аттестация: дифференцированный зачёт.

Общая трудоемкость рабочей программы дисциплины составляет 5 зачетных единиц.

- занятия лекционного типа – 32 часа;
- лабораторные занятия – 64 часа;
- самостоятельная работа обучающегося в течение семестра, не включая период сессии – 82 часа;
- промежуточная аттестация (дифференцированный зачёт) – 2 часа.

Объём контактной работы обучающегося с преподавателем (занятия лекционного типа, лабораторные занятия, дифференцированный зачёт) составляет 98 часов.

Работа с обучающимися в интерактивных формах составляет 64 часа (лабораторные занятия).

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

Общая трудоёмкость дисциплины составляет 5 зачётных единиц, 180 академических часов.

№ п/п	Раздел дисциплины	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоёмкость (в часах)					Консультации перед экзаменом (часов)	Промежуточная аттестация (в часах)
			Всего	Аудиторные часы		Сам. работа во время занятий (не включая период сессии)	Сам. работа во время промежуточной аттестации		
				Лекции	Лабораторные занятия				
1	2	3	4	5	6	7	8	9	10
1.	Основные принципы объектно-ориентированного программирования	1-2	22	4	8	10			
2.	Объектно-ориентированная модель	3-4	22	4	8	10			
3.	Классы	5-6	22	4	8	10			
4.	Основные алгоритмические отличия С++ от С	7-8	22	4	8	10			
5.	Средства объектного программирования языка С++	9-10	22	4	8	10			
6.	Средства объектно-ориентированного программирования С++	11-12	22	4	8	10			
7.	Обобщенное программирование	13-14	22	4	8	10			
8.	Стандартная библиотека С++	15-16	24	4	8	12			
9.	Дифференцированный зачёт		2						2
<b>Всего</b>			<b>180</b>	<b>32</b>	<b>64</b>	<b>82</b>			<b>2</b>

## Программа и основное содержание лекций (32 часа)

1. Основные принципы объектно-ориентированного программирования (8 часов)
  - 1.1. Эволюция методологий программирования
    - 1.1.1. Начало начал, или первое поколение языков программирования.
    - 1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования.
    - 1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования.
    - 1.1.4. Зарождение объектной модели, четвертое поколение языков программирования.
    - 1.1.5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование.
    - 1.1.6. Парадигмы программирования.
  - 1.2. Составные части объектного подхода
    - 1.2.1. Абстрагирование
    - 1.2.2. Инкапсуляция
    - 1.2.3. Модульность
    - 1.2.4. Иерархия
    - 1.2.5. Типизация
    - 1.2.6. Параллелизм
    - 1.2.7. Сохраняемость
2. Объектно-ориентированная модель (4 часа)
  - 2.1. Понятие объекта
  - 2.2. Свойства, присущие объектам
    - 2.2.1. Состояние
    - 2.2.2. Поведение
    - 2.2.3. Идентичность
  - 2.3. Отношения между объектами
    - 2.3.1. Типы отношений
    - 2.3.2. Связь (ассоциация)
    - 2.3.3. Агрегация
3. Классы (4 часа)
  - 3.1. Природа классов.
  - 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF
  - 3.3. Отношения между классами.
    - 3.3.1. Типы отношений
    - 3.3.2. Ассоциация
    - 3.3.3. Агрегация
    - 3.3.4. Использование
    - 3.3.5. Наследование
    - 3.3.6. Инстанцирование
  - 3.4. Отношения между классами и объектами



4. Основные алгоритмические отличия C++ от C (4 часа)
  - 4.1. Использование ссылок. Передача аргументов функции по ссылке.
  - 4.2. Использование констант.
  - 4.3. Логические тип и перечисления.
  - 4.4. Операторы управления динамической памятью, инициализация массивов.
  - 4.5. Структура программы, отдельная компиляция и особенности использования статической памяти.
  - 4.6. Пространства имен и исключения (краткий обзор)
  - 4.7. Библиотека ввода вывода (краткий обзор iostream)
  - 4.8. Функциональный полиморфизм.
  - 4.9. Лямбды, замыкания и указатели на функции
  
5. Средства объектного программирования языка C++ (4 часа)
  - 5.1. Представление объектов и классов.
    - 5.1.1. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс.
    - 5.1.2. Структура объявления класса.
    - 5.1.3. Доступ к членам класса.
    - 5.1.4. Поля данных класса как механизм реализации состояния объекта.
    - 5.1.5. Функции члены класса как механизм реализации поведения объекта.
    - 5.1.6. Спецификаторы доступа для обеспечения инкапсуляции.
    - 5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы.
    - 5.1.8. Конструирование и уничтожение объектов и массивов объектов.
    - 5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания.
    - 5.1.10. Описание селекторов и модификаторов.
    - 5.1.11. Перегрузка операторов C++ как реализация поведения с predetermined смыслом.
    - 5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности.
    - 5.1.13. Статические поля и методы классов. Инициализация статических полей.
  - 5.2. Реализация отношений между объектами и классами
    - 5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни.
    - 5.2.2. Использование и зависимость от интерфейсов.
    - 5.2.3. Объекты при передаче параметров и возврате из методов.
    - 5.2.4. Варианты реализации отношения клиент-сервер.
    - 5.2.5. Внутренние классы.
  - 5.3. Средства модульности
    - 5.3.1. Разбиение программы на модули, типы связывания
    - 5.3.2. Объектная структура модуля, утилиты для анализа модулей, архивные библиотеки
    - 5.3.3. Компоновка программ и динамических библиотек, .ELF и .COFF, особенности Windows и Linux
    - 5.3.4. Поддержка многомодульных проектов в средах разработки
    - 5.4.5. Системы автоматической сборки, Make, Autoconf, CMake

## 6. Средства объектно-ориентированного программирования C++ (4 часа)

- 6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковской.
- 6.2. Одиночное наследование.
  - 6.2.1. Понятие производного класса.
  - 6.2.2. Управление доступом в производных классах.
  - 6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии.
  - 6.2.4. Абстрактные классы и виртуальные функции.
  - 6.2.5. Виртуальный полиморфизм.
  - 6.2.6. Информация о типе на этапе выполнения. RTTI.
- 6.3. Множественное наследование
  - 6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий.
  - 6.3.2. Порядок вызовов конструкторов и деструкторов при множественном наследовании.
  - 6.3.3. Чистые виртуальные классы, понятие интерфейса.
  - 6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP.
- 6.4. Пространства имен.
  - 6.4.1. Пространства имен как средство реализации модульности.
  - 6.4.2. Поиск имен и разрешение конфликтов.
  - 6.4.3. Объединение пространств имен.
  - 6.4.4. Принципы дизайна пакетов.
- 6.5. Обработка исключений.
  - 6.5.1. Обработка ошибок.
  - 6.5.2. Группировка и перехват исключений.
  - 6.5.3. Управление ресурсами.
  - 6.5.4. Исключения и эффективность.
  - 6.5.5. Альтернативные методы обработки ошибок.
  - 6.5.6. Стандартные исключения.

## 7. Обобщенное программирование. (4 часа)

- 7.1. Шаблоны классов.
  - 7.1.1. Определение шаблона.
  - 7.1.2. Инстанцирование.
  - 7.1.3. Параметры шаблонов и проверка типов.
- 7.2. Шаблоны функций.
- 7.3. Специализация.
- 7.4. Наследование и шаблоны.

## 8. Стандартная библиотека C++ (4 часа)

- 8.1. Библиотека стандартных шаблонов
  - 8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов-функций)

- 8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры)
- 8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов)
- 8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)
- 8.2. Библиотека ввода-вывода
  - 8.2.1. Потоки вывода. Вывод типов определяемых пользователем.
  - 8.2.2. Потоки ввода. Ввод типов определяемых пользователем.
  - 8.2.3. Форматирование в потоках ввода-вывода.
  - 8.2.4. Буферизация.

### **Программа лабораторных занятий (64 часа)**

- 1. Основные принципы объектно-ориентированного программирования (8 часов)
  - 1.1. Эволюция методологий программирования
    - 1.1.1. Начало начал, или первое поколение языков программирования.
    - 1.1.2. Развитие алгоритмических абстракций, или второе поколение языков программирования.
    - 1.1.3. Модуль как единица построения программных систем, третье поколение языков программирования.
    - 1.1.4. Зарождение объектной модели, четвертое поколение языков программирования.
    - 1.1.5. Объектные языки программирования, объектно-ориентированные языки программирования, объектно-ориентированный анализ, дизайн и проектирование.
    - 1.1.6. Парадигмы программирования.
  - 1.2. Составные части объектного подхода
    - 1.2.1. Абстрагирование
    - 1.2.2. Инкапсуляция
    - 1.2.3. Модульность
    - 1.2.4. Иерархия
    - 1.2.5. Типизация
    - 1.2.6. Параллелизм
    - 1.2.7. Сохраняемость
- 2. Объектно-ориентированная модель (8 часов)
  - 2.1. Понятие объекта
  - 2.2. Свойства, присущие объектам
    - 2.2.1. Состояние
    - 2.2.2. Поведение
    - 2.2.3. Идентичность
  - 2.3. Отношения между объектами
    - 2.3.1. Типы отношений
    - 2.3.2. Связь (ассоциация)
    - 2.3.3. Агрегация

### 3. Классы (8 часов)

- 3.1. Природа классов.
- 3.2. UML – унифицированный язык моделирования. Четырехуровневая метамодель MOF
- 3.3. Отношения между классами.
  - 3.3.1. Типы отношений
  - 3.3.2. Ассоциация
  - 3.3.3. Агрегация
  - 3.3.4. Использование
  - 3.3.5. Наследование
  - 3.3.6. Инстанцирование
- 3.4. Отношения между классами и объектами

### 4. Основные алгоритмические отличия C++ от C (8 часов)

- 4.1. Использование ссылок. Передача аргументов функции по ссылке.
- 4.2. Использование констант.
- 4.3. Логические тип и перечисления.
- 4.4. Операторы управления динамической памятью, инициализация массивов.
- 4.5. Структура программы, отдельная компиляция и особенности использования статической памяти.
- 4.6. Пространства имен и исключения (краткий обзор)
- 4.7. Библиотека ввода вывода (краткий обзор iostream)
- 4.8. Функциональный полиморфизм.
- 4.9. Лямбды, замыкания и указатели на функции

### 5. Средства объектного программирования языка C++ (8 часов)

- 5.1. Представление объектов и классов.
  - 5.1.1. Реализация поведения объектов на примере добавления функций—членов в структуры. Структура как вырожденный класс.
  - 5.1.2. Структура объявления класса.
  - 5.1.3. Доступ к членам класса.
  - 5.1.4. Поля данных класса как механизм реализации состояния объекта.
  - 5.1.5. Функции члены класса как механизм реализации поведения объекта.
  - 5.1.6. Спецификаторы доступа для обеспечения инкапсуляции.
  - 5.1.7. Средства управления жизнью объекта. Конструкторы и деструкторы.
  - 5.1.8. Конструирование и уничтожение объектов и массивов объектов.
  - 5.1.9. Особенности использования конструктора копии, конструктора по умолчанию, оператора присваивания.
  - 5.1.10. Описание селекторов и модификаторов.
  - 5.1.11. Перегрузка операторов C++ как реализация поведения с предопределенным смыслом.
  - 5.1.12. Дружественность как механизм нарушения инкапсуляции. Достоинства и недостатки механизма дружественности.
  - 5.1.13. Статические поля и методы классов. Инициализация статических полей.

- 5.2. Реализация отношений между объектами и классами
  - 5.2.1. Ассоциация и агрегация объектов и классов. Зависимость по времени жизни.
  - 5.2.2. Использование и зависимость от интерфейсов.
  - 5.2.3. Объекты при передаче параметров и возврате из методов.
  - 5.2.4. Варианты реализации отношения клиент-сервер.
  - 5.2.5. Внутренние классы.
- 5.3. Средства модульности
  - 5.3.1. Разбиение программы на модули, типы связывания
  - 5.3.2. Объектная структура модуля, утилиты для анализа модулей, архивные библиотеки
  - 5.3.3. Компоновка программ и динамических библиотек, .ELF и .COFF, особенности Windows и Linux
  - 5.3.4. Поддержка многомодульных проектов в средах разработки
  - 5.4.5. Системы автоматической сборки, Make, Autoconf, CMake
- 6. Средства объектно-ориентированного программирования C++ (8 часов)
  - 6.1. Наследование как средство организации иерархий классов. Принцип замещения Лисковской.
  - 6.2. Одиночное наследование.
    - 6.2.1. Понятие производного класса.
    - 6.2.2. Управление доступом в производных классах.
    - 6.2.3. Конструкторы и деструкторы, совмещение имен методов при наследовании, иерархии.
    - 6.2.4. Абстрактные классы и виртуальные функции.
    - 6.2.5. Виртуальный полиморфизм.
    - 6.2.6. Информация о типе на этапе выполнения. RTTI.
  - 6.3. Множественное наследование
    - 6.3.1. Проблема множественного наследования. Виртуальное наследование как средство разрешения коллизий.
    - 6.3.2. Порядок вызовов конструкторов и деструкторов при множественном наследовании.
    - 6.3.3. Чистые виртуальные классы, понятие интерфейса.
    - 6.3.4. Принципы дизайна иерархий классов. OCP, DIP, ISP.
  - 6.4. Пространства имен.
    - 6.4.1. Пространства имен как средство реализации модульности.
    - 6.4.2. Поиск имен и разрешение конфликтов.
    - 6.4.3. Объединение пространств имен.
    - 6.4.4. Принципы дизайна пакетов.
  - 6.5. Обработка исключений.
    - 6.5.1. Обработка ошибок.
    - 6.5.2. Группировка и перехват исключений.
    - 6.5.3. Управление ресурсами.
    - 6.5.4. Исключения и эффективность.
    - 6.5.5. Альтернативные методы обработки ошибок.
    - 6.5.6. Стандартные исключения.

## 7. Обобщенное программирование. (8 часов)

### 7.1. Шаблоны классов.

#### 7.1.1. Определение шаблона.

#### 7.1.2. Инстанцирование.

#### 7.1.3. Параметры шаблонов и проверка типов.

### 7.2. Шаблоны функций.

### 7.3. Специализация.

### 7.4. Наследование и шаблоны.

## 8. Стандартная библиотека C++ (8 часов)

### 8.1. Библиотека стандартных шаблонов

#### 8.1.1. Общие сведения (понятия контейнеров, итераторов и объектов-функций)

#### 8.1.2. Контейнеры (виды контейнеров, последовательные и ассоциативные контейнеры, адаптеры)

#### 8.1.3. Итераторы (итератор как обобщение указателя, классы итераторов)

#### 8.1.4. Алгоритмы (примеры алгоритмов с использованием итераторов: алгоритмы сортировки, алгоритмы, не изменяющие содержание контейнера, алгоритмы, изменяющие содержание контейнера)

### 8.2. Библиотека ввода-вывода

#### 8.2.1. Потоки вывода. Вывод типов определяемых пользователем.

#### 8.2.2. Потоки ввода. Ввод типов определяемых пользователем.

#### 8.2.3. Форматирование в потоках ввода-вывода.

#### 8.2.4. Буферизация.

### Самостоятельная работа студентов (82 часа)

Перечень занятий на СРС	Объем, час
Изучение, повторение теоретического материала лекций в течении семестра	82 часа

## 5. Перечень учебной литературы.

### 5.1. Основная литература

1. Страуструп, Бьерн. Язык программирования C++ / Бьерн Страуструп ; пер. с англ. С. Анисимова и М. Кононова ; под ред. Ф. Андреева и А. Ушакова. Спец. изд. Москва : Бином-Пресс, 2008. 1098 с. : ил. ; 24 см.

### 5.2. Дополнительная литература

2. Соломон, Мартин К. Oracle. Программирование на языке Java : [пер. с англ.] / Мартин К. Соломон, Нирва Мориссо-Леруа, Джули Басу. М. : ЛОРИ, 2007. XXVIII, 484 с. : ил. ; 26 см.

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

Самостоятельная работа студентов поддерживается следующими учебными пособиями:

3. Замулин, Александр Васильевич. Введение в объектно-ориентированное программирование : Учебно- метод. пособие / А.В. Замулин ; Новосиб. гос. ун-т. Новосибирск : НГУ, 2000. 34 с. ; 20 см. (Учебник НГУ) .

## **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.

Интернет-ресурсы:

1. Справочный сайт по языку С++ и стандартной библиотеке (на английском языке), свободный доступ: <http://www.cplusplus.com/>
2. Справочный сайт по языку С++ и стандартной библиотеке (многоязычная версия), свободный доступ: <http://www.cppreference.com/>
3. Справочный сайт Microsoft по Visual Studio и С++ (многоязычная версия), свободный доступ: <https://docs.microsoft.com/ru-ru/cpp/?view=vs-2017>
4. Справочный сайт по системе сборки CMake (на английском языке), свободный доступ: <https://gitlab.kitware.com/cmake/community/wikis/home>
5. Справочные материалы по системе автоматического модульного тестирования Google test (на английском языке), свободный доступ: <https://github.com/google/googletest/tree/master/googletest/docs>

### **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.**

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Также используется среда программирования Atmel Studio.

## **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, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой 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.**

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

#### ***Текущий контроль***

Текущий контроль успеваемости по дисциплине «Объектно-ориентированное программирование 1» осуществляется во время проведения лабораторных занятий и коллоквиума. Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования C++ и демонстрации ее работы в соответствии с требованиями лабораторного задания, прохождения автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

При защите задания обучающийся должен:

- Изложить необходимый для решения теоретический материал
- Указать методику решения задания, предоставить диаграмму классов в нотации UML
- Предоставить исходный код программы или программ, реализующий требования задания
- Продемонстрировать корректную работу программы и прохождения автоматических модульных тестов.

За решенные задания начисляются баллы в соответствии с условиями задач. В случае сдачи задания с недочетами или позже установленного срока начисляются баллы меньше балловой стоимости задачи, но не менее 50% при условии успешного решения.

Коллоквиум состоит из 18-20 вопросов проводится в виде письменного теста, за каждый правильный ответ на вопрос коллоквиума начисляется от 1 до 2 баллов в зависимости от сложности вопроса.

#### ***Промежуточная аттестация***

Освоение компетенций оценивается согласно шкале оценки уровня сформированности компетенции. Положительная оценка по дисциплине выставляется в том случае, если заявленные компетенции ПК-1 и ПК-2 сформированы не ниже порогового уровня в части, относящейся к формированию способности использовать специализированные знания в области объектно-ориентированного программирования в профессиональной деятельности.



Окончательная оценка работы студента в течение семестра происходит на дифференцированном зачёте в конце семестра с учётом сданных лабораторных работ и результатов коллоквиума. Дифференцированный зачёт проходит в устной форме по вопросам билета. В процессе сдачи дифференцированного зачёта студенту могут задаваться дополнительные задания по теме вопросов билета в форме написания фрагмента кода, демонстрирующего определённый механизм языка программирования или технику объектно-ориентированного программирования на C++. Вопросы подбираются таким образом, чтобы проверить уровень сформированности компетенций ПК-1 и ПК-2.

Итоги промежуточной аттестации (дифференцированного зачёта) оцениваются по пятибалльной шкале:

- оценка «отлично» ставится при безупречном ответе на оба вопроса и отлично сделанном портфолио (продвинутый уровень освоения компетенций);

- оценка «хорошо» ставится при наличии небольших недочётов в ответе на вопросы и портфолио (базовый уровень освоения компетенций);

- оценка «удовлетворительно» ставится при отсутствии ответа на один из вопросов или при больших недочётах в портфолио (пороговый уровень освоения компетенций);

- оценка «неудовлетворительно» ставится при невозможности ответа на оба вопроса или отсутствии портфолио (компетенции не сформированы).

Вывод об уровне сформированности компетенций принимается преподавателем. Положительная оценка ставится, когда все компетенции освоены не ниже порогового уровня. Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

### Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Объектно-ориентированное программирование 1».

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	ПК 1.1	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	ПК 1.2 ПК 2.2	Отсутствие минимальных умений. Не умеет решать стандартные задачи. Имеют место	Продемонстрированы частично основные умения. Решены типовые за-	Продемонстрированы все основные умения. Решены все основные задания с негрубыми ошибками или с недочётами.	Продемонстрированы все основные умения. Решены все основные задания в полном объеме

		грубые ошибки.	дачи. Допущены негрубые ошибки.		без недочетов и ошибок.
--	--	----------------	---------------------------------	--	-------------------------

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

### Список тем заданий для самостоятельного решения

- Раздельная компиляция и пространства имен, структурные средства языка C++
- Перегрузка функций, указатели на функции, перечисления
- Классы в языке C++
- Иерархии классов, наследование
- Обобщенное программирование

### Вопросы на дифференцированный зачёт

#### *Категория 1 – Основы объектного подхода*

1. Эволюция методологий программирования. Парадигмы программирования
2. Основные принципы объектного подхода. Абстрагирование
3. Основные принципы объектного подхода. Инкапсуляция
4. Основные принципы объектного подхода. Модульность
5. Основные принципы объектного подхода. Иерархия
6. Основные принципы объектного подхода. Типизация
7. Объект с точки зрения ООП. Состояние. Поведение
8. Объект с точки зрения ООП. Идентичность и жизненный цикл объектов
9. Объект с точки зрения ООП. Взаимоотношения между объектами.
10. Классы. Природа классов. Мета модель. Инстанцирование.
11. Классы. Структура класса. Абстрактные классы и интерфейсы
12. Классы. Принцип подстановки Лисковской. Принцип разделения интерфейсов
13. Классы. Средства UML для построения диаграмм классов
14. Классы. Отношения между классами. Ассоциация и агрегация

15. Классы. Иерархии классов. Зависимость

*Категория 2 – Средства объектно-ориентированного программирования на C++*

1. Модель памяти и структура программы. Классы памяти. Ссылки
2. Средства абстракции C++. Структура класса. Статические члены и их инициализация
3. Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья
4. Модульность, отдельная компиляция, пространства имен, using директива.
5. Представление иерархических отношений. Наследование
6. Представление иерархических отношений. Агрегация. Зависимость по времени жизни
7. Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм
8. C++: средства реализации состояния объектов; реализация поведения
9. Перегрузка операторов
10. Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании
11. Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов
12. Исключения в C++. Обработка исключений. Умные указатели
13. Шаблоны классов и шаблоны функций. Специализация
14. Основы STL. Структура и назначение. Контейнеры
15. Основы STL. Аллокаторы и итераторы

**Пример билета на дифференцированный зачёт**

1. Вопрос из категории 1
2. Вопрос из категории 2

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 «Объектно-ориентированное программирование 1»  
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