Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования «Новосибирский национальный исследовательский государственный университет» (Новосибирский государственный университет, НГУ)

Высший колледж информатики

огласовано

Директор ВКИ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Окунев А.Г.

*подпись*

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ**

направление подготовки: 15.03.06 Мехатроника и робототехника

направленность (профиль): Мехатроника и робототехника

Форма обучения: очная

Разработчики:

Лях Татьяна Викторовна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель программы:

д.т. н., Назаров А.Д. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск, 2020

**Содержание**

[1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы 3](#_Toc21097778)

[2. Место дисциплины в структуре образовательной программы 3](#_Toc21097779)

[3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося 3](#_Toc21097780)

[4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий 4](#_Toc21097781)

[5. Перечень учебной литературы 6](#_Toc21097782)

[6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины 6](#_Toc21097784)

[7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине 7](#_Toc21097785)

[8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине 7](#_Toc21097786)

[9. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине 7](#_Toc21097787)

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

| Результаты освоения образовательной программы  (компетенции) | В результате изучения дисциплины обучающиеся должны: | | |
| --- | --- | --- | --- |
| знать | уметь | владеть |
| **ОПК 3 Владение современными информационными технологиями, готовность применять современные средства автоматизированного проектирования и машинной графики при проектировании систем и их отдельных модулей, а также для подготовки конструкторско-технологической документации, соблюдать основные требования информационной безопасности** | методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области проектировании систем и их отдельных модулей  *- теоретические основы объектно-ориентированного программирования (абстракция, инкапсуляция, иерархия, модульность, типизация, параллелизм, сохраняемость).* | применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области проектировании систем и их отдельных модулей;  *- использовать основные парадигмы объектно-ориентированного программирования при разработке собственных программных архитектур.* | Навыками программирования,отладки и тестирования прототипов программно-технических комплексов задач в области проектировании систем и их отдельных модулей.  *- технологиями разработки обектно-ориентированных архитектур программного обеспечения.* |
| **ПК 2 способность составлять математические модели мехатронных и робототехнических систем, их подсистем и отдельных элементов и модулей, включая информационные, электромеханические, гидравлические, электрогидравлические, электронные устройства и средства вычислительной техники** | основные методы проектирования и разработки программного обеспечения, необходимого для обработки информации и управления в мехатронных и робототехнических системах;  *- средства языка C++, поддерживающие основные принципы объектно-ориентированного подхода.* | проектировать и разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах;  - *использовать средства поддержки процедурного и объектно-*  *ориентированного стиля программирования при разработке программного обеспечения* | навыками проектирования разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах;  - *методами разработки устойчивого, надежного и оптимизированного программного обеспечения с использованием современного С++.* |

# 2. Место дисциплины в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин бакалавриата:

• «Императивное программирование»

• «Декларативное программирование»

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

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

Трудоемкость дисциплины: 3-й семестр: 4 з.е. (144 ч),

4-й семестр: 5 з.е. (180 ч).

Форма промежуточной аттестации: 3семестр – дифференцированный зачет,

4 семестр – экзамен

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вид деятельности | Семестр | |
| 3 | 4 |
| 1 | Лекции, ч | 32 | 32 |
| 2 | Практические занятия, ч |  |  |
| 3 | Лабораторные работы ч | 32 | 32 |
| 4 | Занятия в контактной форме, ч  , из них | 66 | 68 |
| 5 | из них аудиторных занятий, ч | 64 | 64 |
| 6 | в электронной форме, ч | - | - |
| 7 | консультаций, час. | - | -2 |
| 8 | промежуточная аттестация, ч | 2 | 2 |
| 9 | самостоятельная промежуточная аттестация, ч | - | 36 |
| 10 | Самостоятельная работа, час. | 78 | 76 |
| 11 | Всего, ч | 144 | 180 |

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

***3 семестр***

Лекции (32 ч)

|  |  |
| --- | --- |
| Наименование темы и их содержание | Объем,  час |
| Раздел 1. Введение в объектно-ориентированное программирование. | |
| 1. Основные принципы объектно-ориентированного программирования. Основные принципы ООП. 1.1. Предпосылки к возникновению объектно-ориентированной порадигмы.  1.2 Абстрагирование. Инкапсуляция. Модульность. Иерархия. Типизация. Параллелизм. Сохраняемость | 2 |
| 2. Объектно-ориентированная модель. Понятие объекта. Отношения между объектами.  2.1. Понятие объекта. Состояние, поведение и идентичность.  2.2. Связь, ассоциация, агрегация, композиция. | 2 |
| 3. Классы.  3.1. Определение класса. Типы данных.  3.2. Отношения между классами. Ассоциация. Агрегация. Наследование. Использование. Инстанцирование. | 2 |
| Раздел 2. Средства языка С++ |  |
| 4. Средства процедурного программирования С++. Основные отличия С++ от языка С.  4.1. Порядок компиляции и сборки.  4.2. Классы  4.3. Ссылки.  4.4. Константы.  4.5. Операторы new и delete.  4.6. Типы-перечисления.  4.7. Пространства имен.  4.8. Параметрический полиморфизм. | 4 |
| 5. Средства объектно-ориентированного программирования С++  5.1. Структура класса.  5.2. Средства инкапсуляции в С++.  5.3. Конструкторы и деструкторы.  5.4. Конструктор копирования.  5.5. Типы памяти в С++. Средства для управления временем жизни объекта.  5.6. Конструирование и уничтожение объектов и массивов объектов.  5.7. Списки инициализации.  5.8. Перегрузка операторов С++.  5.9. Механизм дружественности.  5.10. Статические поля классов.  5.11. Внутренние классы. | 8 |
| 6. Наследование в С++  6.1. Наследование.  6.2. Порядок конструирования объекта.  6.3. Модификаторы доступа при наследовании.  6.4. Виртуальный полиморфизм.  6.5. Абстрактные классы и интерфейсы.  6.6. Механизм исключений.  6.7. Множественное наследование. Его проблемы.  6.8. Приведение типов.  6.9. Типы наследования в С++. | 6 |
| 6. Средства обобщенного программирования на C++  6.1. Шаблоны классов.  6.2. Шаблоны функций.  6.3. Специализация шаблонов.  6.4. Наследования и шаблоны.  6.6. Вариативные шаблоны. | 4 |
| 7. Введение в стандартную библиотеку С++.  7.1. Потоки ввода вывода.  7.2. Контейнеры.  7.3. Алгоритмы.  7.4. Итераторы.  7.5. Аллокаторы. | 4 |
| Итого; | 32 |

***4 семестр***

Лекции (32 ч)

|  |  |
| --- | --- |
| Наименование темы и их содержание | Объем,  час |
| Раздел 1. Современный С++ | |
| 1. Семантика перемещения. 1.1. Rvalue-ссылки.  1.2. Конструкторы перемещения.  1.3. Перемещающий оператор присваивания.  1.4. Std::move.  1.5. Правило трех (пяти).  1.6. Принцип прямой передачи. Использование std::forward. | 6 |
| 2. Принцип SFINAE.  2.1. Объяснение принципа SFINAE.  2.2. SFINAE при частичной спецификации шаблона.  2.3. Примеры.  2.4. std::type\_traits  2.5. std::enable\_if  2.6. Альтернативы: отправка тэгов (tag dispatching) и сильные типы (strong types) | 6 |
| 3. Лямбда-выражения.  3.1. Определение лямбда-выражений.  3.2. Методики захвата данных.  3.3. std::function. | 4 |
| 4. Умные указатели.  4.1. Принципы работы умных указателей.  4.2. Std::auto\_ptr  4.3. Std::unique\_ptr  4.4. Std::shared\_ptr  4.5. Std::weak\_ptr | 2 |
| 5. Нововведения стандарта С++20.  5.1. Ограничения и концепции. | 2 |
| Раздел 2. Многопоточное программирование на С++ |  |
| 5. Введение в многопоточность в С++  5.1. Разница между процессом, потоком и волокном.  5.2. Запуск, остановка и оповещение о событиик потока в c++. Std::thread.  5.3. Присоединяемые и не присоединяемые потоки.  5.4. Синхронизация потоков.  5.5. Проблема голодания.  5.6. Честное распределение между потоками.  5.7. Мьютексы  5.8. lock\_guard  5.9. condition\_variable  5.10. Асинхронные вызовы. std::async.  5.11. Std::future и std::promise.  5.12. Атомарные операции.  5.13. Программирование без блокировок.  5.14. Барьеры памяти. | 10 |
| Раздел 3. Анализ и разбор объектных файлов. |  |
| 3.1. Этапы сборки.  3.2. Структура объектных файлов.  3.3. Утилита objdump. Утилита readelf. | 2 |
| Итого: | 32 |

Лабораторные занятия, 3-й семестр (32 ч)

|  |  |
| --- | --- |
| Содержание практического занятия | Объем, час |
| Лабораторная работа №1. Тип данных. Что такое переменная. Типы памяти в Си. Введение в классы. Разбор заданий. | 2 |
| Лабораторная работа №2. Структура класса. Ссылки. Конструктор и деструктор. Разбор заданий. | 2 |
| Лабораторная работа №3. Параметрический полиморфизм. Управление динамической памятью. Оператор const. | 2 |
| Лабораторная работа №4. Конструктор копирования. Списки инициализации. | 2 |
| Лабораторная работа №5. Перегрузка операторов. | 2 |
| Лабораторная работа №6. Введение в наследование. Protected. | 2 |
| Лабораторная работа №7. Исключения. | 2 |
| Лабораторная работа №8. Виртуальный полиморфизм. | 2 |
| Лабораторная работа №9. Ключевое слово explicit. Enum и enum-класс. | 2 |
| Лабораторная работа №10. Абстрактные классы и интерфейсы. | 2 |
| Лабораторная работа №11. Множественное наследование. | 2 |
| Лабораторная работа №12. Приведение типов. | 2 |
| Лабораторная работа №13. Шаблоны: введение. | 2 |
| Лабораторная работа №14. Вариативные шаблоны. Std::tuple. | 2 |
| Лабораторная работа №15. Constexpr и auto. | 2 |
| Лабораторная работа №16. Контрольная. | 2 |
| Итого: | 32 |

Лабораторные занятия, 4-й семестр (32 ч)

|  |  |
| --- | --- |
| Содержание практического занятия | Объем, час |
| Лабораторная работа №1. Rvalue-ссыки. Введение в семантику переммещения. | 2 |
| Лабораторная работа №2. Std::move. | 2 |
| Лабораторная работа №3. Std::forward. | 2 |
| Лабораторная работа №4. Введение в принцип SFINAE. | 2 |
| Лабораторная работа №5. Использование std::type\_traits и std::enable\_if | 2 |
| Лабораторная работа №6. Аллокаторы в stl. Allocator\_traits. | 2 |
| Лабораторная работа №7. Лямбды и их constexpr версии | 2 |
| Лабораторная работа №8. Использование std::function. | 2 |
| Лабораторная работа №9. Умные указатели. Использование и самостоятельная реализация. | 2 |
| Лабораторная работа №10. Многопоточность: std::thread. | 2 |
| Лабораторная работа №11. Многопоточность: std::mutex. Std::lock\_guard | 2 |
| Лабораторная работа №12. Многопоточность: std::async. | 2 |
| Лабораторная работа №13. Многопоточность: std::futute и std::promise. | 2 |
| Лабораторная работа №14. Аллокаторы в stl. Allocator\_traits. | 2 |
| Лабораторная работа №15. Анализ и разбор объектных файлов. Использование утилит objdump и readelf. | 2 |
| Лабораторная работа №16. Контрольная. | 2 |
| Итого: | 32 |

Самостоятельная работа студентов

3 семестр 78 ч.

|  |  |
| --- | --- |
| Перечень занятий на СРС | Объем, час |
| Изучение темы дисциплины по учебной литературе, учебным пособиям, поиск в интернете. Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Объектно-ориентированное программирование выложены на странице курса в сети Интернет . | 20 |
| Подготовка к лабораторным работам, к текущему контролю знаний. Разбор решенных задач, самостоятельное решение задач. | 50 |
| Подготовка к дифференцированному зачету.. Повторение теоретического материала по вопросам, совпадающим с темами лекций. | 8 |
| **Итого:** | **78** |

4 семестр 114 ч.

|  |  |
| --- | --- |
| Перечень занятий на СРС | Объем, час |
| Изучение темы дисциплины по учебной литературе, учебным пособиям, поиск в интернете. Изучение предлагаемых теоретических разделов в соответствии с настоящей Программой. Учебно-методические материалы по дисциплине «Объектно-ориентированное программирование выложены на странице курса в сети Интернет . | 26 |
| Подготовка к лабораторным работам, к текущему контролю знаний. Разбор решенных задач, самостоятельное решение задач. | 50 |
| Подготовка к экзамену. Повторение теоретического материала по вопросам, совпадающим с темами лекций. | 36 |
| **Итого:** | **114** |

# 5. Перечень учебной литературы

***5.1.Основная литература***

1. Б. Страуструп Язык программирования C++, спец. изд., 3 изд. /Пер. с англ. — СПб.; М.: «Невский Диалект» — «Издательство БИНОМ», 2004 г., 2011 г. ISBN 978-5-7989-0425-9, 0-201-70073-5, 5-7940-0064-3, 5-7989-0223-4, 5-7940-0064-3

***5.2.Дополнительная литература***

2. Г. Буч, Объектно-ориентированный анализ и проектирование с примерами приложений. 3-е изд. /Пер. с англ. «Вильямс». 2010 г. ISBN 978-5-8459-1401-9, 0-201-89551-X

3. Скотт Мейерс, Эффективное использование C++. 50 рекомендаций по улучшению ваших программ и проектов, Пер. с англ. - ДМК, 2006. ISBN: 5-469-01213-1, 0-201-92488-9

4. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. — М. ДМК, 2000.

# 6. 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.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту, социальные сети.

***6.1 Современные профессиональные базы данных:***

Не используются.

***6.2. Информационные справочные системы***

- Справочный сайт по языку C++ и стандартной библиотеке (на английском языке), свободный доступ: <http://www.cplusplus.com/>

- Справочный сайт по языку C++ и стандартной библиотеке (многоязычная версия), свободный доступ: <http://www.cppreference.com/>

- Справочные материалы по системе автоматического модульного тестирования Google test (на английском языке), свободный доступ: <https://github.com/google/googletest/tree/master/googletest/docs>

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

**7.1 Перечень программного обеспечения**

Для обеспечения реализации дисциплины «Объектно-ориентированное программирование» используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office, Microsoft Visual Studio 2019 и Jet Brains CLion 2019.

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

Для реализации дисциплины «Объектно-ориентированное программирование» используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, лабораторных занятий, 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.

Оборудование:

1. Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления) – для проведения лекционных и практических занятий.

2. Компьютерный класс (с выходом в Internet) - для организации самостоятельной работы обучающихся.

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».

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

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

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

***Текущий контроль успеваемости*** по дисциплине «Объектно-ориентированное программирование» осуществляется во время проведения лабораторных занятий в следующей форме:

* Решение и сдача лабораторных работ. Выполнение задания без замечаний засчитываются учащемуся. В случае обнаружения недочетов в решении учащегося лабораторная работа не зачитывается до тех пор, пока эти недочеты не будут исправлены. В случае сдачи задания позже установленного срока задание не за-считывается.
* В конце семестра проводится тестирование в форме контрольной работы (теста с вопросами, подразумевающими открытую форму ответа). Контрольная состоит из 10-15 задач и теоритических вопросов и проводится в виде письменного теста, за каждое правильное решение начисляется от 1 до 2 баллов в зависимости от сложности вопроса.

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

С целью контроля прогресса решения лабораторных заданий студенты сохраняют исходный код в процессе работы над практическими заданиями в системе контроля версий на основе технологии git или mercurial на общедоступных репозиториях в сети интернет. Выбор репозитория осущестлвяется на усмотрение студента по согласованию с преподавателем.

Решенные и успешно сданные практические работы совместно с ответами на вопросы контрольной формируют портфолио обучающегося. Для получения оценки «удовлетворительно» за работу в семестре необходимо сдать не менее 2-х задач и набрать за контрольную не меньше 70% баллов. Для получения оценки «хорошо» необходимо сдать 3 задачи и набрать не менее 85% баллов на контрольную. Для получения оценки «отлично» необходимо сдать 4 задачи и набрать не менее 95% баллов за контрольную.

***Промежуточная аттестация*** по дисциплине «Объектно-ориентированное программирование» проводится по завершению каждого периода ее освоения (семестра) в форме дифференцированного зачета и экзамена.

Количество набранных баллов за сдачу практических работ и коллоквиума является основным критерием при выставлении оценки во время дифференцированного зачета и является одним из условий прохождения промежуточной аттестации.

Дифференцированный зачет и экзамен проходит в устной форме в формате собеседования по темам курса. В процессе сдачи зачета и экзамена студенту могут задаваться дополнительные задания по теме вопросов в форме написания фрагмента кода демонстрирующего определенный механизм языка программирования или технику объектно-ориентированного программирования на C++.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

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

|  |  |  |
| --- | --- | --- |
| **Код компетенции** | **Результат обучения по дисциплине** | **Оценочное средство** |
| ОПК-3 | Знать теоретические основы объектно-ориентированного программирования (абстракция, инкапсуляция, иерархия, модульность, типизация, параллелизм, сохраняемость). | Дифференцированный зачет  Экзамен |
| Уметь использовать основные парадигмы объектно-ориентированного программирования при разработке собственных программных архитектур. | Дифференцированный зачет  Экзамен |
| Владеть технологиями разработки обектно-ориентированных архитектур программного обеспечения. | Дифференцированный зачет  Экзамен |
| ПК-2 | Знать средства языка C++, поддерживающие основные принципы объектно-ориентированного подхода. | Дифференцированный зачет  Экзамен |
| Уметь использовать средства поддержки процедурного и объектно-  ориентированного стиля программирования при разработке программного обеспечения | Дифференцированный зачет  Экзамен |
| Владеть методами разработки устойчивого, надежного и оптимизированного программного обеспечения с использованием современного С++. | Дифференцированный зачет  Экзамен |

Таблица 9.2

|  |  |
| --- | --- |
| **Критерий** | **Оценка** |
| **Дифференцированный зачет**  Студент проявил уверенное знание теоретических основ объектно-ориентированного программирования.  Демонстрировал понимание и способность использовать объектно-ориентированный подход при разработке программного обеспечения. Демонстрировал глубокое знание нотаций UML, способность читать диаграммы и описывать в нотации UML собственную программную архитектуру с применением декорирования.  Уверенное владение всем спектром возможностей одной из применявшихся на курсе сред разработки, а также дополнительных утилит для анализа, отладки и оптимизации.  **Экзамен**  Продемонстрировал уверенное знание следующих тем, изученных в семестре на лекциях и практических семинарах:  - основные средства поддержки процедурного стиля программирования в языке программирования C++;  - основные средства языка C++, поддерживающих принципы объектно-ориентированного подхода;  - основы стандартной библиотеки С++;  - средства обобщенного программирования на C++ (шаблоны);  Демонстрировал умение применять полученные знания при разработке программного обеспечения.  При решении заданий продемонстрировал обширные умения использовать современные средства С++.  Присутствовал на всех лекциях и семинарах (допускается отсутствие с уважительной причиной) | *Отлично* |
| **Дифференцированный зачет**  Студент проявил обширные знания теоретических основ объектно-ориентированного программирования.  Демонстрировал понимание и способность использовать объектно-ориентированный подход при разработке программного обеспечения. Допускает непринципиальные неточности при использовании нотаций UML. Способен читать диаграммы и описывать в нотации UML собственную программную архитектуру.  Владеет возможностями отладки и реализации многомодульных приложений в выбранной среде разработки.  **Экзамен**  Допускает непринципиальные неточности при ответе на вопросы по темам, изученным в семестре на лекциях и практических семинарах  - основные средства поддержки процедурного стиля программирования в языке программирования C++;  - основные средства языка C++, поддерживающих принципы объектно-ориентированного подхода;  - основы стандартной библиотеки С++;  - средства обобщенного программирования на C++ (шаблоны);  Демонстрировал умение применять полученные знания при разработке программного обеспечения.  При решении заданий продемонстрировал умение использовать основные современные средства С++.  Присутствовал на всех лекциях и семинарах (допускается отсутствие с уважительной причиной). | *Хорошо* |
| **Дифференцированный зачет**  Студент проявил поверхностное знание теоритических основ объектно-ориентированного программирования.  Студент испытывал трудности при использовании UML-нотаций и чтении диаграмм.  **Экзамен**  Демонстрировал пробелы в знаниях при ответах на вопросы по следующим темам, изученным в семестре на лекциях и практических семинарах:  - основных средств поддержки процедурного стиля программирования в языке программирования C++;  - основных средств языка C++, поддерживающих принципы объектно-ориентированного подхода;  - основ стандартной библиотеки С++;  - средств обобщенного программирования на C++ (шаблоны);  Ошибки, допущенные при ответах на вопросы, не касались принципиального понимания основ С++.  Демонстрировал умение применять имеющиеся знания при разработке программного обеспечения.  Присутствовал на всех лекциях и семинарах (допускается отсутствие с уважительной причиной). | *Удовлетворительно* |
| **Дифференцированный зачет**  Студент не знает теоретических основ объектно-ориентированного программирования.  Не знает UML-нотации, не умеет читать диаграммы.  **Экзамен**  Демонстрировал принципиальное непонимание основ С++. Не отвечает на дополнительные и наводящие вопросы. | *Неудовлетворительно* |

***Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения***

Примеры вопросов к контрольным:

1. Перечислите все средства языка С++, реализующие принцип инкапсуляции.
2. Реализуйте свой аналог умного указателя std::unique\_ptr.
3. Реализуйте класс, обеспечивающий принцип RAII при работе с потоками.

**Лист актуализации рабочей программы дисциплины**

**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ »**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика внесенных изменений (с указанием пунктов документа) | Дата и №  протокола Ученого совета ВКИ НГУ | Подпись  ответственного |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |