

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Павловский филиал ННГУ

УТВЕРЖДЕНО

решением Ученого совета ННГУ

протокол № 6 от 31.05.2023 г.

Рабочая программа дисциплины

Объектно-ориентированный анализ и программирование

Уровень высшего образования

Бакалавриат

Направление подготовки / специальность

09.03.03 - Прикладная информатика

Направленность образовательной программы

Прикладная информатика в экономике и управлении

Форма обучения

очная, очно-заочная

г. Павлово

2023 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.01.01 Объектно-ориентированный анализ и программирование относится к части, формируемой участниками образовательных отношений образовательной программы.

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

| Формируемые компетенции (код, содержание компетенции) | Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции | | Наименование оценочного средства | |
|--|---|--|------------------------------------|-------------------------------|
| | Индикатор достижения компетенции (код, содержание индикатора) | Результаты обучения по дисциплине | Для текущего контроля успеваемости | Для промежуточной аттестации |
| ПК-6: Способен составлять технико-экономическое обоснование проектных решений и техническое задание на разработку ИС (ИИС) | <p>ПК-6.1: Демонстрирует знание методик технико-экономического обоснования проектных решений, связанных с созданием ИС (ИИС)</p> <p>ПК-6.2: Демонстрирует умение выполнять технико-экономические расчеты при обосновании проектных решений, составлять техническую документацию на разработку ИС (ИИС)</p> <p>ПК-6.3: Имеет практический опыт технико-экономического обоснования конкретного проектного решения и представления технической документации на разработку ИС (ИИС)</p> | <p>ПК-6.1:</p> <p>Знать основные принципы планирования в рамках проектной, организационно-управленческой, аналитической и научно-исследовательской деятельности;</p> <p>принципы управление документированием программного обеспечения;</p> <p>о роли процесса документирования в процессе создания программного обеспечения (ИС);</p> <p>виды программной документации, этапы документирования;</p> <p>российские и международные стандарты в области документирования ПО, их обозначение, классификаторы стандартов, структура или группы стандарта ЕСПД;</p> <p>Структуру и содержание документов по этапам жизненного цикла программного обеспечения;</p> <p>Общие принципы систем управления версиями при разработке ПО.</p> <p>ПК-6.2:</p> <p>Уметь описать состояние проблемы</p> | Задания | Зачёт: Контрольные вопросы |

| | | | | |
|--|---|---|--------|--|
| | | <p>по теме проекта, доказать актуальность темы</p> <p>составлять список литературных источников по научной теме;</p> <p>создавать документацию к ПО</p> <p>в соответствии с требованиями;</p> <p>Выбирать инструменты для документирования разработки ПО;</p> <p>Создать начальное техническое задание на разработку ПО;</p> <p>Применять прикладное и специальное программное обеспечения для документирования ПО.</p> <p>Искать информацию о научных и технологических достижениях в сети Интернет и из других источников.</p> <p>ПК-6.3:</p> <p>Владеть навыками создания плана-графика по реализации учебного проекта;</p> <p>основными принципами документирования программного обеспечения;</p> <p>методикой анализа необходимой документации в процессе создания программного обеспечения.</p> | | |
| <p>ПК-8: Способен разрабатывать лингвистическое, информационное и программное обеспечение ИС (ИИС) и сопровождающую его документацию</p> | <p>ПК-8.1: Демонстрирует знание современных языков и систем программирования, формализмов описания знаний на концептуальном и инфологическом уровнях, требований к технической документации на все виды обеспечения ИС (ИИС)</p> <p>ПК-8.2: Применяет современные языки и системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях при</p> | <p>ПК-8.1:</p> <p>Знает современные языки и системы программирования; формализмы описания знаний на концептуальном и инфологическом уровнях;</p> <p>требования к технической документации на все виды обеспечения ИС (ИИС).</p> <p>ПК-8.2:</p> <p>Способен применять современные языки и</p> | Задачи | <p>Зачёт:</p> <p>Контрольные вопросы</p> |

| | | | | |
|--|--|--|--|--|
| | разработке лингвистического, информационного и программного обеспечения ИИС и сопровождающей ее документации ПК-8.3: Имеет практический опыт разработки лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей ее документации | системы программирования, формализмы описания знаний на концептуальном и инфологическом уровнях при разработке лингвистического, информационного и программного обеспечения ИИС и сопровождающей его документации. ПК-8.3: Способен осуществлять разработку лингвистического, информационного и программного обеспечения конкретной ИС (ИИС) и сопровождающей его документации. | | |
|--|--|--|--|--|

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

| | очная | очно-заочная |
|--|--------------------------|--------------------------|
| Общая трудоемкость, з.е. | 2 | 2 |
| Часов по учебному плану | 72 | 72 |
| в том числе | | |
| аудиторные занятия (контактная работа): | | |
| - занятия лекционного типа | 16 | 8 |
| - занятия семинарского типа (практические занятия / лабораторные работы) | 16 | 8 |
| - КСР | 1 | 1 |
| самостоятельная работа | 39 | 55 |
| Промежуточная аттестация | 0 Зачёт | 0 Зачёт |

3.2. Содержание дисциплины

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

| Наименование разделов и тем дисциплины | Всего (часы) | в том числе | | | Самостоятельная работа обучающегося, часы |
|--|-----------------|--|---|-------|--|
| | | Контактная работа (работа во взаимодействии с преподавателем), часы из них | | | |
| | | Занятия лекционного типа | Занятия семинарского типа (практические занятия/лаборат орные работы), | Всего | |

| | | | | | часы | | | | | |
|---|-------------|------------------|-------------|------------------|-------------|------------------|-------------|------------------|-------------|------------------|
| | О Ф О | О З Ф О | О Ф О | О З Ф О | О Ф О | О З Ф О | О Ф О | О З Ф О | О Ф О | О З Ф О |
| Тема 1. Введение в ООП. Процедурное и объектно-ориентированное программирование | 2 | 1 | 2 | 1 | | | 2 | 1 | | |
| Тема 2. Классы и объекты | 11 | 11 | 2 | 1 | 3 | 2 | 5 | 3 | 6 | 8 |
| Тема 3. Работа с объектами | 11 | 12 | 2 | 1 | 3 | 1 | 5 | 2 | 6 | 10 |
| Тема 4. Конструкторы и деструкторы | 8 | 10 | 2 | 1 | 2 | 1 | 4 | 2 | 4 | 8 |
| Тема 5. Перегрузка операторов | 10 | 10 | 2 | 1 | 2 | 1 | 4 | 2 | 6 | 8 |
| Тема 6. Наследование и виртуальные функции | 10 | 10 | 2 | 1 | 2 | 1 | 4 | 2 | 6 | 8 |
| Тема 7. Шаблоны, паттерны проектирования | 9 | 7 | 2 | 1 | 2 | 1 | 4 | 2 | 5 | 5 |
| Тема 8. Введение в STL | 10 | 10 | 2 | 1 | 2 | 1 | 4 | 2 | 6 | 8 |
| Аттестация | 0 | 0 | | | | | | | | |
| КСР | 1 | 1 | | | | | 1 | 1 | | |
| Итого | 72 | 72 | 16 | 8 | 16 | 8 | 33 | 17 | 39 | 55 |

Содержание разделов и тем дисциплины

Тема 1. Язык C++ и ООП. Особенности языка C++. Программное обеспечение и стандарты языка C++. Среда разработки Visual Studio C++. Краткий обзор других распространенных сред разработки ПО на C++.

Тема 2. Объявление класса. Открытые и закрытые члены класса. Статические члены класса. Перегрузка методов.

Тема 3. Передача объектов аргументами. Возвращение результатом объектов. Указатель на объект. Указатели на члены класса. Ссылки на объекты. Массивы объектов. Динамическое выделение памяти под объекты. Дружественные функции и классы.

Тема 4. Создание и перегрузка конструктора класса. Правила оформления конструктора. Вызов конструктора. Конструктор копии. Деструкторы и правила их оформления. Использование деструкторов.

Тема 5. Внешняя операторная функция для переопределения бинарного оператора. Перегрузка операторной функции. Переопределение унарных операторов внешними функциями. Перегрузка операторов методами класса. Перегрузка оператора присваивания.

Тема 6. Наследование классов и типы наследования. Переопределение методов и виртуальные функции. Многоуровневое и многократное наследование. Конструкторы и деструкторы при наследовании. Чисто виртуальные методы и абстрактные классы. Существующие и несуществующие члены класса.

Тема 7. Обобщенные функции. Обобщенная функция с несколькими параметрами. Перегрузка обобщенных функций. Явная специализация обобщенной функции. Обобщенные классы. Явная специализация обобщенного класса. Значения параметров по умолчанию. Наследование обобщенных классов. Целочисленные обобщенные параметры. Порождающие паттерны (фабричный метод, абстрактная фабрика, одиночка, прототип, строитель). Структурные паттерны (адаптер, декоратор, заместитель, компоновщик, мост, приспособленец, фасад). Паттерны поведения (шаблонный метод, итератор, команда, наблюдатель, посетитель, посредник, состояние, стратегия, хранитель, цепочка обязанностей).

Тема 8. Общие сведения о библиотеке STL. Состав библиотеки: контейнеры, алгоритмы, итераторы. Базовые контейнеры STL.

Практические занятия /лабораторные работы организуются, в том числе, в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

На проведение практических занятий / лабораторных работ в форме практической подготовки отводится: очная форма обучения - 16 ч., очно-заочная форма обучения - 8 ч.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Цель самостоятельной работы - формирование навыков непрерывного самообразования и профессионального совершенствования.

Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.

Основные виды самостоятельной работы студентов:

- работа с основной и дополнительной литературой;
- изучение категориального аппарата дисциплины;
- самостоятельное изучение тем дисциплины;
- подготовка к зачёту, экзамену;
- работа в библиотеке;
- изучение сайтов по темам дисциплины в сети Интернет.

Работа с основной и дополнительной литературой

Изучение рекомендованной литературы следует начинать с учебников и учебных пособий, затем переходить к научным монографиям и материалам периодических изданий. Работа с литературой предусматривает конспектирование наиболее актуальных и познавательных материалов. Это не только мобилизует внимание, но и способствует более глубокому осмыслению материала, его лучшему запоминанию, а также позволяет студентам проводить систематизацию и сравнительный анализ изучаемой информации. Таким образом, конспектирование – одна из основных форм самостоятельного труда, которая требует от студента активно работать с учебной литературой и не ограничиваться конспектом лекций. Студент должен уметь самостоятельно подбирать необходимую литературу для учебной и научной работы, уметь обращаться с предметными каталогами и библиографическим справочником библиотеки.

Изучение категориального аппарата дисциплины.

Изучение и осмысление экономических категорий требует проработки лекционно-го материала, выполнения практических заданий, изучение словарей, энциклопедий, справочников.

Индивидуальная самостоятельная работа студента направлена на овладение и грамотное применение экономической терминологии в области компьютерного моделирования.

Самостоятельное изучение тем дисциплины

Особое место отводится самостоятельной проработке студентами отдельных разделов и тем изучаемой дисциплины. Такой подход вырабатывает у студентов инициативу, стремление к увеличению объема знаний, умений и навыков, всестороннего овладения способами и приемами профессиональной деятельности.

Изучение вопросов определенной темы направлено на более глубокое усвоение основных категорий экономической теории, понимание экономических процессов, происходящих в обществе, совершенствование навыка анализа теоретического и эмпирического материала.

Подготовка к зачёту, экзамену

Промежуточная аттестация студентов по дисциплине проходит в виде зачёта и экзамена, предусматривающего оценку. Условием успешного прохождения промежуточной аттестации является систематическая работа студента в течение семестра. В этом случае подготовка к зачёту, экзамену является систематизацией всех полученных знаний по данной дисциплине.

Рекомендуется внимательно изучить перечень вопросов к зачёту, экзамену, а также использовать в процессе обучения программу, учебно-методический комплекс, другие методические материалы.

Желательно спланировать троекратный просмотр материала перед зачётом, экзаменом. Во-первых, внимательное чтение с осмыслением, подчеркиванием и составлением краткого плана ответа. Во-вторых, повторная проработка наиболее сложных вопросов. В-третьих, быстрый просмотр материала или планов ответов для его систематизации в памяти.

Самостоятельная работа в библиотеке

Важным аспектом самостоятельной подготовки студентов является работа с библиотечным фондом.

Эта работа предполагает различные варианты повышения профессионального уровня студентов:

- а) получение книг для подробного изучения в течение семестра на научном абонементе;
- б) изучение книг, журналов, газет - в читальном зале;
- в) возможность поиска необходимого материала посредством электронного каталога;
- г) получение необходимых сведений об источниках информации у сотрудников библиотеки.

Изучение сайтов по темам дисциплины в сети Интернет

Ресурсы Интернет являются одним из альтернативных источников быстрого поиска требуемой информации. Их использование возможно для получения основных и дополнительных сведений по изучаемым материалам. Необходимо помнить об оформлении ссылок на Интернет-источники.

Для повышения эффективности самостоятельной работы студентов преподавателю целесообразно использовать следующие виды деятельности:

- консультации,
- выдача заданий на самостоятельную работу,
- информационное обеспечение обучения,
- контроль качества самостоятельной работы студентов.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Задания) для оценки сформированности компетенции ПК-6:

Во всех вариантах требуется создать шаблон некоторого целевого класса *A*, возможно, реализованный с применением некоторого серверного класса *B*. Это означает, что объект класса *B* используется как элемент класса *A*. В качестве серверного класса может быть указан либо класс, созданный программистом, либо класс из стандартной библиотеки – например, *std::vector*.

Варианты целевых или серверных классов, создаваемых программистом, приведены в табл. 1.

Таблица 1 Варианты целевых или серверных классов

| Имя класса | Назначение |
|-------------|---|
| Vect | одномерный динамический массив |
| List | двунаправленный список |
| Stack | стек |
| BinaryTree | бинарное дерево |
| Queue | односторонняя очередь |
| Deque | двусторонняя очередь (допускает вставку и удаление из обоих концов очереди) |
| Set | множество (повторяющиеся элементы в множество не заносятся; элементы в множестве хранятся отсортированными) |
| SparseArray | разреженный массив |

Если вместо серверного класса указан динамический массив, то это означает, что для хранения элементов контейнерного класса используется массив, размещаемый с помощью операции *new*.

Во всех вариантах необходимо предусмотреть генерацию и обработку исключений для возможных ошибочных ситуаций.

Во всех вариантах показать в клиенте *main()* использование созданного класса, включая ситуации, приводящие к генерации исключений. Показать инстанцирование шаблона для типов *int*, *double*, *std::string*.

Варианты заданий приведены в табл. 2.

Таблица 2 Варианты заданий

| <i>Вариант</i> | <i>Целевой шаблонный класс</i> | <i>Реализация с применением</i> |
|----------------|--------------------------------|---------------------------------|
| 1 | Vect | std :: list |
| 2 | List | – |
| 3 | Stack | динамический массив |
| 4 | Stack | Vect |
| <i>Вариант</i> | <i>Целевой шаблонный класс</i> | <i>Реализация с применением</i> |
| 5 | Stack | List |
| 6 | Stack | std :: vector |
| 7 | Stack | std :: list |
| 8 | BinaryTree | – |
| 9 | Queue | Vect |
| 10 | Queue | List |
| 11 | Queue | std :: list |
| 12 | Deque | Vect |
| 13 | Deque | List |
| 14 | Deque | std :: list |
| 15 | Set | динамический массив |
| 16 | Set | Vect |
| 17 | Set | List |
| 18 | Set | std :: vector |

| | | |
|----|-------------|-------------|
| | | |
| 19 | Set | std :: list |
| 20 | SparseArray | List |

Критерии оценивания (оценочное средство - Задания)

| Оценка | Критерии оценивания |
|---------------------|---|
| превосходно | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного и дополнительного материала |
| отлично | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного материала |
| очень хорошо | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание материала, допущено не более 2 неточностей не принципиального характера |
| хорошо | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, допущены неточности не принципиального характера, но обучающийся показывает систему знаний по теме своими ответами на поставленные вопросы |
| удовлетворительно | Задание выполнено не в полном объеме (решено более 50% поставленных задач), но обучающийся допускает ошибки, нарушена последовательность ответа, но в целом раскрывает содержание основного материала |
| неудовлетворительно | Задание выполнено не в полном объеме (решено менее 50% поставленных задач), обучающийся дает неверную информацию при ответе на поставленные задачи, допускает грубые ошибки при толковании материала, демонстрирует незнание основных терминов и понятий. |
| плохо | Задание не выполнено, обучающийся демонстрирует полное незнание материала |

5.1.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-8:

Вариант 1

1. Определить класс с именем *STUDENT*, содержащий следующие поля:

- Фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *STUDENT*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *STUDENT*; число записей задается в начале работы модуля;
- вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0; если таких студентов нет, вывести соответствующее сообщение.

Вариант 2

1. Определить класс с именем *STUDENT*, содержащий следующие поля:

- Фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *STUDENT*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *STUDENT*; число записей задается в начале работы модуля;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5; если таких студентов нет, вывести соответствующее сообщение.

Вариант 3

1. Определить класс с именем *STUDENT*, содержащий следующие поля:

- Фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *STUDENT*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *STUDENT*; число записей задается в начале работы модуля;
- вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2; если таких студентов нет, вывести соответствующее сообщение.

Вариант 4

1. Определить класс с именем *AEROFLOT*, содержащий следующие поля:

- название пункта назначения рейса;
- номер рейса;
- тип самолета.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *AEROFLOT*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *AEROFLOT*; число записей задается в начале работы модуля;
- вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с

клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 5

1. Определить класс с именем *AEROFLOT*, содержащий следующие поля:

- название пункта назначения рейса;
- номер рейса;
- тип самолета.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *AEROFLOT*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *AEROFLOT*; число записей задается в начале работы модуля;
- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры; если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 6

1. Определить класс с именем *WORKER*, содержащий следующие поля:

- фамилия и инициалы работника;
- название занимаемой должности;
- год поступления на работу.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *WORKER*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *WORKER*; число записей задается в начале работы модуля;

- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры; если таких работников нет, вывести на дисплей соответствующее сообщение.

Вариант 7

1. Определить класс с именем *TRAIN*, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *TRAIN*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *TRAIN*; число записей задается в начале работы модуля
- вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени; если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 8

1. Определить класс с именем *TRAIN*, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *TRAIN*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *TRAIN*; число записей задается в начале работы модуля

- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 9

1. Определить класс с именем *TRAIN*, содержащий следующие поля:

- название пункта назначения;
- номер поезда;
- время отправления.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *TRAIN*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *TRAIN*; число записей задается в начале работы модуля;
- вывод на экран информации о поезде, номер которого введен с клавиатуры; если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 10

1. Определить класс с именем *MARSH*, содержащий следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *MARSH*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *MARSH*; число записей задается в начале работы модуля;

- вывод на экран информации о маршруте, номер которого введен с клавиатуры; если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Вариант 11

1. Определить класс с именем *MARSH*, содержащий следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *MARSH*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *MARSH*; число записей задается в начале работы модуля;
- вывод на экран информации о маршрутах, которые начинаются или кончаются в пункте, название которого введено с клавиатуры; если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Вариант 12

1. Определить класс с именем *NOTE*, содержащий следующие поля:

- фамилия, имя, отчество;
- номер телефона;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *NOTE*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *NOTE*; число записей задается в начале работы модуля;

- вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

Вариант 13

1. Определить класс с именем *NOTE*, содержащий следующие поля:

- фамилия, имя, отчество;
- номер телефона;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *NOTE*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *NOTE*; число записей задается в начале работы модуля;
- вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение.

Вариант 14

1. Определить класс с именем *NOTE*, содержащий следующие поля:

- фамилия, имя, отчество;
- номер телефона;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *NOTE*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *NOTE*; число записей задается в начале работы модуля;

- вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

Вариант 15

1. Определить класс с именем *ZNAK*, содержащий следующие поля:

- фамилия, имя, отчество;
- знак Зодиака;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *ZNAK*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *ZNAK*; число записей задается в начале работы модуля;
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение.

Вариант 16

1. Определить класс с именем *ZNAK*, содержащий следующие поля:

- фамилия, имя, отчество;
- знак Зодиака;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *ZNAK*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *ZNAK*; число записей задается в начале работы модуля;

- вывод на экран информации о людях, родившихся под знаком, наименование которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение.

Вариант 17

1. Определить класс с именем *ZNAK*, содержащий следующие поля:

- фамилия, имя, отчество;
- знак Зодиака;
- день рождения (массив из трех чисел).

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *ZNAK*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *ZNAK*; число записей задается в начале работы модуля;
- вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры; если таких нет, выдать на дисплей соответствующее сообщение.

Вариант 18

1. Определить класс с именем *PRICE*, содержащий следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в руб.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *PRICE*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *PRICE*; число записей задается в начале работы модуля;

- вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать на дисплей соответствующее сообщение.

Вариант 19

1. Определить класс с именем *PRICE*, содержащий следующие поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в руб.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *PRICE*.

1. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *PRICE*; число записей задается в начале работы модуля;
- вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры; если такого магазина нет, выдать на дисплей соответствующее сообщение.

Вариант 20

1. Определить класс с именем *ORDER*, содержащий следующие поля:

- расчетный счет плательщика;
- расчетный счет получателя;
- перечисляемая сумма.

Определить методы доступа к этим полям и перегруженные операции извлечения и вставки для объектов типа *ORDER*.

2. Написать C++ программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из объектов типа *ORDER*;
- число записей задается в начале работы модуля;

- вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры; если такого расчетного счета нет, выдать на дисплей соответствующее сообщение.

Критерии оценивания (оценочное средство - Задачи)

| Оценка | Критерии оценивания |
|------------|--|
| зачтено | Все компетенции (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно» |
| не зачтено | Хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно» или «плохо» |

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

| Уровень сформированности компетенций (индикатора достижения компетенций) | плохо | неудовлетворительно | удовлетворительно | хорошо | очень хорошо | отлично | превосходно |
|--|---|--|--|---|---|--|--|
| | не зачтено | | | зачтено | | | |
| <u>Знания</u> | Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа | Уровень знаний ниже минимальных требований. Имели место грубые ошибки | Минимально допустимый уровень знаний. Допущено много негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет. | Уровень знаний в объеме, превышающем программу подготовки. |
| <u>Умения</u> | Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки | Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме | Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами | Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами. | Продемонстрированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнен | Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов |

| | | | | | | | |
|---------------|--|---|--|--|--|--|---|
| | | | | недочетами | | ы все задания в полном объеме | |
| <u>Навыки</u> | Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки | Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами | Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами | Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов | Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов | Продемонстрирован творческий подход к решению нестандартных задач |

Шкала оценивания при промежуточной аттестации

| Оценка | | Уровень подготовки |
|------------|---------------------|--|
| зачтено | превосходно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой |
| | отлично | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично». |
| | очень хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо» |
| | хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо». |
| | удовлетворительно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно» |
| не зачтено | неудовлетворительно | Хотя бы одна компетенция сформирована на уровне «неудовлетворительно». |
| | плохо | Хотя бы одна компетенция сформирована на уровне «плохо» |

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-6

1. Основные понятия ООП. Отличия объектно-ориентированной парадигмы от структурной. Основные элементы языка C++.
2. Виды объектно-ориентированных языков программирования: основанные на классах и на прототипах, со статической и динамической типизацией.

3. Наследование и иерархия классов. Виды иерархий классов. Примеры наследования.
4. Полиморфизм и его предназначение. Способы связывания сообщения с функцией. Примеры.
5. Шаблоны функций и их применение.
6. Шаблоны классов. Примеры.
7. Основные алгоритмы STL. Их временная сложность.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-8

1. Типы данных. Классификация типов данных. Примеры типов данных в языке C++. Базовые типы данных в языке C++. Хранение переменных базовых типов в оперативной памяти.
2. Инкапсуляция и её реализация в C++. Интерфейс и реализация класса. Области видимости.
3. Операции языка C++. Унарные, бинарные, тернарные операции. Порядок вычисления выражения.
4. Создание, копирование и удаление объектов. Конструкторы и деструкторы, порядок их вызова. Примеры.
5. Перегрузка операторов для встроенных и пользовательских типов данных. Примеры.
6. Библиотека ввода-вывода C++ как пример инкапсуляции. Перегрузка операторов побитового сдвига. Потоки ввода и вывода в стандартной библиотеке C. Буферизованные и небуферизованные потоки. Работа с файлами.
7. Принцип подстановки, подклассы и подтипы. Иерархия подтипов для объектов и указателей на объекты. Примеры.
8. Библиотека ввода-вывода C++ как пример наследования.
9. Перегрузка функций и аргументы по умолчанию. Решение проблемы неоднозначности имён функций.
10. Перегрузка и переопределение методов. Статическое и динамическое переопределение. В каких случаях они применяются?
11. Чистые виртуальные методы и абстрактные классы. Примеры.
12. Проблема обращения полиморфизма и её решение при помощи RTTI.
13. Компиляция шаблонов функций в C++. Проблема раздельной компиляции при реализации шаблонов и её решение.
14. Функциональные объекты в C++. Примеры их использования.
15. Понятие контейнера. Свойства контейнеров в STL, виды контейнеров в STL, операции над контейнерами. Линейные контейнеры и операции над ними. Примеры.

16. Понятие итератора. Итераторы как адаптеры контейнеров.
17. Категории итераторов, примеры итераторов разных категорий.
18. Связные списки. Операции над списком и их временная сложность.
19. Сортировка связного списка слиянием. Временная сложность алгоритма.
20. Векторы (массивы переменной длины). Операции над вектором и их временная сложность.
21. Двухнаправленные очереди. Операции над двухнаправленной очередью и их временная сложность.
22. Строки как контейнеры. Операции над строками и их временная сложность. Копирование строк.
23. Адаптированные классы: стеки. Операции над стеками и их временная сложность.
24. Адаптированные классы: очереди. Операции над очередями и их временная сложность.
25. Сбалансированные бинарные деревья. Пирамидальная сортировка. Временная сложность алгоритма.
26. Очереди с приоритетами. Операции над очередями с приоритетами и их временная сложность.
27. Ассоциативные контейнеры и их реализация. Множества и мультимножества. Временная сложность операций над множествами.
28. Отображения и мультиотображения. Операции над отображениями и мультиотображениями и их временная сложность.

Критерии оценивания (оценочное средство - Контрольные вопросы)

| Оценка | Критерии оценивания |
|------------|--|
| зачтено | Все компетенции (части компетенции), на формирование которой направлена дисциплина, сформированы на уровне не ниже «удовлетворительно» |
| не зачтено | Хотя бы одна часть компетенции сформирована на уровне «неудовлетворительно» или «плохо» |

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Барков И. А. Объектно-ориентированное программирование : учебник для вузов / Барков И. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2023. - 700 с. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-47113-3., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=864800&idb=0>.
2. Зыков Сергей Викторович. Объектно-ориентированное программирование : учебник и

практикум для вузов / С. В. Зыков. - 2-е изд. - Москва : Юрайт, 2025. - 151 с. - (Высшее образование). - ISBN 978-5-534-16941-6. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=925680&idb=0>.

3. Лозовский В. В. Объектно-ориентированное программирование / Лозовский В. В., Морозов В. А., Гололобов А. А. - Москва : РТУ МИРЭА, 2024. - 107 с. - Книга из коллекции РТУ МИРЭА - Информатика. - ISBN 978-5-7339-2120-4., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=896946&idb=0>.

Дополнительная литература:

1. Тузовский Анатолий Федорович. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. - Москва : Юрайт, 2024. - 213 с. - (Высшее образование). - URL: <https://urait.ru/bcode/537332> (дата обращения: 15.08.2024). - ISBN 978-5-534-16316-2 : 989.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=903277&idb=0>.

2. Унгер А. Ю. Объектно-ориентированное программирование: стандартная библиотека C++ / Унгер А. Ю. - Москва : РТУ МИРЭА, 2022. - 71 с. - Книга из коллекции РТУ МИРЭА - Инженерно-технические науки., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=861425&idb=0>.

3. Юрина Т. А. Объектно-ориентированное программирование : методические указания к курсовому проекту / Юрина Т. А. - Омск : СибАДИ, 2023. - 27 с. - Книга из коллекции СибАДИ - Информатика., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=866412&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционная система Microsoft Windows
2. Пакет прикладных программ Microsoft Office
3. Правовая система «Консультант плюс»
4. Правовая система «Гарант».
5. Интернет браузеры (Mozilla Firefox, Google Chrome)
6. Среда разработки программного обеспечения Lazarus.
7. Интегрированная среда разработки приложений Visual Studio C++.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами, специализированным оборудованием: Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: компьютерная техника с подключением к сети «Интернет», экран, проектор для вывода мультимедиа материалов на экран, динамики для воспроизведения звука, доска. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Специальные условия организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья
Организация обучения по дисциплине инвалидов и лиц с ограниченными возможностями

здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья при наличии таких обучающихся путем создания специальных условий для получения образования. Профессорско-преподавательский состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии). В соответствии с Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утв. Минобрнауки РФ 08.04.2014 АК-44/05вн при изучении дисциплины предполагается использование социально-активных и рефлексивных методов обучения, технологий социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При освоении дисциплины используются различные сочетания видов учебной работы с методами и формами активизации познавательной деятельности обучающихся для достижения запланированных результатов обучения и формирования компетенций. Форма проведения промежуточной аттестации для обучающихся-инвалидов и лиц с ограниченными возможностями здоровья устанавливается с учетом индивидуальных психофизиологических особенностей. По личной просьбе обучающегося с ограниченными возможностями здоровья, изложенной в форме письменного заявления, по дисциплине предусматриваются:

- замена устного ответа на письменный ответ при сдаче зачёта, экзамена;
- увеличение продолжительности времени на подготовку к ответу на зачёте, экзамене;
- при подведении результатов промежуточной аттестации студентов выставляется максимальное количество баллов за посещаемость аудиторных занятий.

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

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 09.03.03 - Прикладная информатика.

Автор(ы): Васин Дмитрий Юрьевич, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 17.11.2022, протокол № 5.