

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Технология разработки программного обеспечения

по направлению подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки:
Математическое моделирование и информационные системы

Форма обучения
Очная

Квалификация
Бакалавр

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОП
К.И. Лившиц

Председатель УМК
С.П. Сущенко

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.

ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.

ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

ПК-2 Способен формализовать и алгоритмизировать поставленную задачу, написать программный код, а также верифицировать работоспособность программного обеспечения и исправить дефекты.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.1 Обладает навыками объектно-ориентированного программирования для решения прикладных задач в профессиональной деятельности.

ИОПК-2.2 Проявляет навыки использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.

ИОПК-2.3 Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи.

ИОПК-2.4 Демонстрирует умение адаптировать существующие математические методы для решения конкретной прикладной задачи.

ИОПК-4.1 Обладает необходимыми знаниями в области информационных технологий, в том числе понимает принципы их работы

ИОПК-4.2 Применяет знания, полученные в области информационных технологий, при решении задач профессиональной деятельности

ИОПК-4.3 Использует современные информационные технологии на всех этапах решения задач профессиональной деятельности

ИОПК-4.4 Демонстрирует умение составлять научные обзоры, рефераты и библиографии по тематике научных исследований.

ИОПК-5.1 Обладает необходимыми знаниями алгоритмов, принципов разработки алгоритмов и компьютерных программ

ИОПК-5.2 Разрабатывает алгоритмы и компьютерные программы для решения задач профессиональной деятельности

ИПК-2.1 Осуществляет построение формальной модели и алгоритма для поставленной задачи, написание программного кода с использованием языков программирования, верификацию работоспособности программного обеспечения и исправление дефектов.

ИПК-2.2 Осуществляет оформление программного кода в соответствии с установленными требованиями, разработку процедур верификации работоспособности и измерения характеристик программного обеспечения, разработку тестовых наборов данных.

ИПК-2.3 Осуществляет работу с системой контроля версий, рефакторинг и оптимизацию программного кода.

2. Оценочные материалы текущего контроля и критерии оценивания

Типовые задания для проведения текущего контроля успеваемости по дисциплине

Примеры лабораторных работ

Задание №1

Запрограммируйте шаблонный класс, реализующий стек. Класс должен поддерживать следующие операции:

1. Помещение объекта в стек;
2. Извлечение объекта из стека;
3. Получение размерности стека.

В случае попытки вызова операции извлечения объекта из стека при условии, что стек пуст, должно генерироваться исключение класса **EStackEmpty**(наследник класса EStackException). Данный класс должен содержать публичный метод char* what(), возвращающий диагностическое сообщение

Задание №2

На основе паттерна Наблюдателя, написать программу для слежения за состоянием выбранного файла.

Ограничимся двумя характеристиками:

1. Существует файл или нет;
2. Каков размер файла.

Программа будет выводить на консоль уведомление о произошедших изменениях в файле.

Существует несколько ситуаций для наблюдаемого файла

1. Файл существует, файл не пустой - на экран выводится факт существования файла и его размер.
2. Файл существует, файл был изменен - на экран выводится факт существования файла, сообщение о том, что файл был изменен и его размер.
3. Файл не существует - на экран выводится информация о том, что файл не существует.

В главной программе создаем объект для отслеживания состояния конкретного файла,

затем определяется объект наблюдатель, после чего связываем Наблюдателя с Источником.

Можно использовать бесконечный цикл, в котором будем обновлять состояние объекта Источника каждые 100 миллисекунд (например так std::this_thread::sleep_for(std::chrono::milliseconds(100));).

Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Текущий контроль успеваемости проводится во время сдачи лабораторных работ.

Каждая работа оценивается оценками «зачтено»/ «не зачтено» в соответствии со следующими критериями:

- предложенные реализации программ являются корректными и решают поставленную задачу;
- студент уверенно отвечает на вопросы, связанные как по предложенной реализации, так и по технологиям, примененным к предложенному решению;
- дает полные ответы на вопросы по теории из соответствующего раздела курса;
- умеет исправлять ошибки и оперативно вносить изменения в программу.

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

Типовые задания для проведения промежуточной аттестации по дисциплине
Контрольные вопросы по курсу

1. Базовые принципы проектирования (SOLID). **Принцип единственной ответственности.**
2. Базовые принципы проектирования (SOLID). Принцип открытости/закрытости.
3. Базовые принципы проектирования (SOLID). **Принцип открытости-закрытости**
4. Базовые принципы проектирования (SOLID). Принцип подстановки Барбары Лисков.
5. Базовые принципы проектирования (SOLID). **Принцип разделения интерфейсов.**
6. Базовые принципы проектирования (SOLID). Принцип инверсии зависимости.
7. Паттерны проектирования. Абстрактная фабрика.
8. Паттерны проектирования. Шаблонный метод.
9. Паттерны проектирования. Наблюдатель.
10. Паттерны проектирования. Адаптер. Строитель.
11. Паттерны проектирования. MVC.

Примеры задач:

Задача 1.

Паттерн Наблюдатель. Его назначение, архитектура. Рассмотрите предложенную легенду, примените к ней паттерн наблюдатель. Решение представить в виде кода с подробными объяснениями.

Легенда

В вузе есть разные библиотеки по разным дисциплинам. Когда студент зачислен на дисциплину, ему рекомендуется присоединиться к соответствующей библиотеке. Библиотека содержит собрание книг. Каждый раз, когда новая книга добавляется в библиотеку, она уведомляет всех участников.

Участвующие классы:

Книга - имеет идентификатор и имя;

Библиотека - есть коллекция книг;

Студент - имеет идентификатор и имя;

Некоторый вуз - имеет набор студентов по разным дисциплинам и соответствующую

библиотеку по каждой дисциплине.

Задача 2.

Паттерн стратегия. Его назначение, архитектура. Рассмотрите предложенную легенду, примените к ней паттерн стратегия. Решение представить в виде кода с подробными объяснениями.

Легенда

Рассмотрим индикатор выполнения – это окно, которое приложение может использовать для индикации хода длительности операции (например, процесса установки). Обычно это прямоугольное окно, которое постепенно заполняется слева направо цветом выделения по мере выполнения операции. У него есть диапазон и текущая позиция. Диапазон представляет собой всю продолжительность операции, а текущая позиция представляет прогресс, достигнутый приложением в завершении операции. Диапазон и текущая позиция используются для определения процента индикатора выполнения, который нужно заполнить цветом выделения. Существуют различные направления заполнения, такие как справа налево, сверху вниз и снизу вверх, также с заданным направлением заливки можно использовать различные типы заливок, такие как непрерывная заливка, прерывистая заливка или заливка на основе узора.

Задание.

Для пользовательского приложения реализовать возможность настройки индикатора выполнения с конкретным классом-заполнителем.

Методические материалы для проведения промежуточной аттестации по дисциплине.

Студент допускается к экзамену в том случае, если в течение семестра успешно сдал все лабораторные работы по курсу на оценку «**зачтено**».

Экзамен проводится в письменной форме по билетам. Билет содержит теоретический вопрос и практическую задачу. Студент письменно готовит ответ на вопросы в билете, решение практической задачи, после чего, в устной форме объясняет/защищает преподавателю подготовленный материал.

Студент допускается к экзамену в том случае, если в течение семестра успешно сдал все лабораторные работы по курсу.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Список вопросов для оценки остаточных знаний

1. Базовые принципы проектирования (SOLID). Принцип единственной ответственности.

2. Базовые принципы проектирования (SOLID). Принцип открытости/закрытости.

3. Базовые принципы проектирования (SOLID). Принцип открытости-закрытости

4. Базовые принципы проектирования (SOLID). Принцип подстановки Барбары Лисков.

5. Базовые принципы проектирования (SOLID). Принцип разделения интерфейсов.

6. Базовые принципы проектирования (SOLID). Принцип инверсии зависимости.

7. Паттерны проектирования. Абстрактная фабрика.

8. Паттерны проектирования. Шалонный метод.

9. Паттерны проектирования. Наблюдатель.

10. Паттерны проектирования. Адаптер. Строитель.

Информация о разработчиках

Андреева Валентина Валерьевна, к.т.н., доцент, ТГУ, ИМПКН, каф. компьютерной безопасности, доцент.