

Федеральное государственное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Т. В. Снытникова

**ТЕХНОЛОГИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОЕКТИРОВАНИЯ И ПРОГРАММИРОВАНИЯ**

Учебно-методическое пособие по изучению дисциплины
для студентов бакалавриата по направлению подготовки
09.03.03 Прикладная информатика

Калининград
Издательство ФГБОУ ВО «КГТУ»
2025

УДК 004.451(075)

Рецензент

кандидат технических наук, доцент,
доцент кафедры прикладной информатики ФГБОУ ВО
«Калининградский государственный технический университет»
О. М. Топоркова

Снытникова, Т. В.

Технология объектно-ориентированного проектирования и программирования: учеб.-метод. пособие по изучению дисциплины для студентов бакалавриата по направлению подготовки 09.03.03 Прикладная информатика / Т. В. Снытникова. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2025. – 22 с.

В учебно-методическом пособии приведен тематический план по дисциплине и даны методические указания по её самостоятельному изучению, подготовке к лабораторным занятиям, подготовке и сдаче экзамена, выполнению самостоятельной работы.

Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины направления подготовки 09.03.03 Прикладная информатика.

Табл. 5, список лит. – 4 наименования

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики 11 ноября 2024 г., протокол № 4

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 21 января 2025 г., протокол № 1

УДК 004.451(075)

© Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Калининградский государственный
технический университет», 2025 г.
© Снытникова Т. В., 2025 г.

ОГЛАВЛЕНИЕ

1. Введение.....	4
2. Тематический план.....	6
3. Содержание дисциплин.....	7
4. Методические указания по проведению лабораторных занятий.....	12
5. Методические указания по выполнению самостоятельной работы.....	12
6. Методические указания по проведению занятий и освоению дисциплины.....	13
7. Требования к аттестации по дисциплине.....	14
7.1 Текущая аттестация.....	14
7.2. Промежуточная аттестация по дисциплине.....	16
8. Заключение.....	20
9. Библиографический список.....	20

1. ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления подготовки 09.03.03 Прикладная информатика, изучающих дисциплину «Технология объектно-ориентированного проектирования и программирования».

Целью освоения дисциплины «Технология объектно-ориентированного проектирования и программирования» является: развитие у студентов базовых навыков программирования на языке высокого уровня (Python) в объектно-ориентированной парадигме программирования, создания объектно-ориентированных программ и алгоритмизации

Задача изучения дисциплины: приобретение студентом способности формулировать требования, проектировать и разрабатывать программное обеспечение на языках высокого уровня.

В результате изучения дисциплины студент должен

знать:

- основные подходы проектирования компьютерного программного обеспечения;
- анализ требований к программному обеспечению;
- основные подходы организации исследовательских и проектных работ;
- правила оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования;

уметь:

- применять методы алгоритмизации, языки и технологии программирования, для создания и использования продуктов и услуг;
- разрабатывать технические спецификации на программные компоненты и их взаимодействие;
- применять основные подходы организации исследовательских и проектных работ;
- применять правила оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования;

владеть:

- способами управления процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе навыками разработки алгоритмов и программ для их практической реализации;
- навыками проектирования программного обеспечения;
- навыками организации исследовательских и проектных работ;

– навыками оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования.

Дисциплина «Технология объектно-ориентированного проектирования и программирования» входит в состав профессионального модуля части, формируемой участниками образовательных отношений, образовательной программы бакалавриата по направлению подготовки 09.03.03 Прикладная информатика.

Для успешного освоения дисциплины, в соответствии с учебным планом, ей предшествуют такие дисциплины, как: «Программирование на языках высокого уровня», «Численные методы».

Результаты освоения дисциплины могут быть использованы при выполнении выпускной квалификационной работы, а также в дальнейшей профессиональной деятельности.

Далее в пособии представлен тематический план, содержащий перечень изучаемых тем и лабораторных занятий, мероприятий текущей аттестации и отводимое на них аудиторное время (занятия в соответствии с расписанием) и самостоятельную работу. При формировании личного образовательного плана на семестр следует оценивать рекомендуемое время на изучение дисциплины, возможно, вам потребует больше времени на выполнение отдельных заданий или проработку отдельных тем.

В разделе «Содержание дисциплины» приведены подробные сведения об изучаемых вопросах, по которым вы можете ориентироваться в случае пропуска каких-то занятий, а также методические рекомендации преподавателя для самостоятельной подготовки, каждая тема имеет ссылки на литературу (или иные информационные ресурсы), а также контрольные вопросы для самопроверки.

Раздел «Требования к аттестации по дисциплине» содержит описание обязательных мероприятий контроля самостоятельной работы и усвоения разделов или отдельных тем дисциплины. Далее изложены требования к завершающей аттестации – экзамену.

Помимо данного пособия, студентам следует использовать материалы, размещенные в соответствующем данной дисциплине разделе ЭИОС, в которые более оперативно вносятся изменения для адаптации дисциплины под конкретную группу.

2. ТЕМАТИЧЕСКИЙ ПЛАН

Общая трудоемкость дисциплины составляет 6 зачетные единицы (ЗЕТ), т. е. 216 академических часов контактной (лекционных и лабораторных занятий) и самостоятельной работы студента, в том числе связанной с текущей и промежуточной (заключительной) аттестацией по дисциплине.

Формы аттестации по дисциплине:

- очная форма, шестой семестр – курсовой проект, экзамен;
- заочная форма, седьмой семестр – курсовой проект, экзамен.

Тематический план лекционных занятий для очной и заочной форм обучения приведен в таблице 1.

Таблица 1 – Тематический план лекционных занятий

Номер темы	Тема лекционного занятия	Объем учебной работы	
		очная форма, ч	заочная форма, ч
Тема 1	Жизненный цикл программного продукта, модели жизненного цикла, сферы их применения	2	1
Тема 2	Основы объектно-ориентированного представления программных систем	2	1
Тема 3	Язык UML	2	1
Тема 4	Создание объектно-ориентированного программного обеспечения. Паттерны проектирования	2	–
Тема 5	Объектно-ориентированное программирование на Python	12	3
Тема 6	Паттерны проектирования	4	2
Тема 7	Тестирование объектно-ориентированных программ	4	1
Тема 8	Параллельная обработка данных	4	1
	Итого	32	10

3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Содержательно структура дисциплины представлена восемью тематическими блоками (темами).

Тема 1. Жизненный цикл программного продукта, модели жизненного цикла, сферы их применения

Перечень изучаемых вопросов:

Цели и задачи дисциплины. Место дисциплины в структуре образовательной программы. Планируемые результаты освоения дисциплины.

Определение понятия «жизненный цикл программного обеспечения». Основные этапы жизненного цикла программного продукта: анализ требований, проектирование, реализация (кодирование), тестирование, внедрение, эксплуатация и сопровождение, завершение проекта.

Модели жизненного цикла: каскадная модель (Waterfall), итеративная модель, V-образная модель, спиральная модель (Spiral Model), Agile-методологии (Scrum, Kanban). Сферы применения различных моделей жизненного цикла.

Рекомендуемая литература: [5]; [6].

Контрольные вопросы:

1. Назовите основные этапы жизненного цикла программного продукта?
2. В чем заключается отличие между каскадной моделью и итеративным подходом?

Когда предпочтительнее использовать каждую из них?

3. Опишите особенности применения Agile-методов в разработке программного обеспечения. Каковы преимущества и ограничения этого подхода?

4. Какие риски могут возникнуть на разных этапах жизненного цикла программного продукта? Как можно минимизировать эти риски?

Тема 2. Язык UML

Перечень изучаемых вопросов:

Введение в язык UML. Основные концепции: объекты, классы, ассоциации, агрегации, композиции, пакеты, компоненты, узлы. Диаграммы UML: типы диаграмм и основные элементы. Применение UML в разработке ПО: проектирование архитектуры системы, документирование требований и спецификаций, коммуникация внутри команды разработчиков.

Структурные диаграммы: классовые диаграммы, компонентные диаграммы, пакетные диаграммы, объектные диаграммы.

Поведенческие диаграммы: диаграммы состояний, диаграммы активности, диаграммы последовательности, диаграммы коммуникации.

Рекомендуемая литература: [2].

Контрольные вопросы:

1. Перечислите основные типы диаграмм UML и кратко опишите их назначение.
2. Чем отличается ассоциация от агрегации и композиции в UML?
3. Как используются диаграммы состояний в проектировании программного обеспечения?
4. В чем разница между диаграммой последовательности и диаграммой коммуникации?

Тема 3. Создание объектно-ориентированного программного обеспечения.

Паттерны проектирования

Перечень изучаемых вопросов:

Определение понятия «паттерн». Исторический контекст возникновения паттернов проектирования. Преимущества использования паттернов при разработке ПО.

Порождающие паттерны: Singleton, Factory Method, Abstract Factory, Builder, Prototype.

Структурные паттерны: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.

Паттерны поведения: Observer, Strategy, Command, Iterator, Mediator, Memento, State, Template Method, Visitor.

Основные принципы SOLID и их связь с паттернами проектирования.

Рекомендуемая литература: [1]; [3, гл. 1].

Контрольные вопросы:

1. Назовите три основных категории паттернов проектирования. Приведите примеры паттернов из каждой категории.
2. Объясните принцип работы паттерна Singleton. Когда его следует применять, а когда лучше избегать?
3. Почему соблюдение принципов SOLID способствует лучшему использованию паттернов проектирования?

Тема 4. Объектно-ориентированное программирование на Python.

Перечень изучаемых вопросов:

Введение в объектно-ориентированное программирование (ООП). Основные концепции ООП: абстракция, инкапсуляция, наследование, полиморфизм. Отличия процедурного программирования от объектно-ориентированного.

Создание классов в Python: атрибуты класса и экземпляра, методы класса и экземпляра, конструкторы (`__init__`), деструктор (`__del__`).

Основные аспекты инкапсуляции в Python. Соглашения об именовании. Использование свойств (properties): декораторы `@property` и `@<property_name>.setter` для управления чтением и записью значений.

Жизненный цикл объектов в Python: сборка мусора, управление памятью, метод `__new__`.

Создание и организация модулей. Понятие модуль в Python: импортирование, пространства имен, пакетирование модулей.

Наследование и полиморфизм. Понятие наследования: родительские и дочерние классы, переопределение методов. Полиморфизм в Python. Множественное наследование.

Абстрактные классы и перегрузка операторов: модуль `abc` и абстрактные классы, перегрузка операторов.

Исключения: механизмы обработки исключений в Python, пользовательские исключения, логика обработки ошибок.

Структуры данных. Встроенные структуры данных в Python. Списки (list), кортежи (tuple), словари (dict), множества (set). Очереди (deque) и стеки. Генераторы списков и словарей.

Работа с функциями. Функции первого порядка. Анонимные функции (lambda). Декораторы. Генераторы.

Ввод/вывод файлов. Форматы файлов: текстовые, бинарные. Работа с путями к файлам. Библиотека `pathlib`.

Строки. Операции над строками. Сериализация и десериализация (pickle, json).

Рекомендуемая литература: [3, гл. 2–8]; [4].

Контрольные вопросы:

1. Что такое класс в Python, и какие основные компоненты он содержит?
2. Чем отличается метод класса от метода экземпляра?
3. Что такое наследование и зачем оно используется в ООП?
4. Объясните разницу между множественным и одиночным наследованием.
5. Как работает механизм обработки исключений в Python?
6. Что такое абстрактный класс и как его создать в Python?

Тема 5. Паттерны проектирования

Перечень изучаемых вопросов:

Паттерны: итератор, декоратор, Observer, Strategy, Command, состояние (State), Singleton.

Паттерны: Adapter, Facade, Flyweight, Abstract Factory, компоновщик, шаблонный метод (Template Method).

Рекомендуемая литература: [3, главы 10–12].

Контрольные вопросы:

1. Какую цель преследует паттерн Singleton в Python, и каким образом его можно реализовать?
2. Какой паттерн позволяет динамически добавлять функциональные возможности объектам в Python, и как он называется?

Тема 6. Тестирование объектно-ориентированных программ.

Перечень изучаемых вопросов:

Введение в тестирование ПО. Особенности тестирования в Python: инструменты для тестирования: unittest, pytest, doctest. Принципы написания тестов в Python.

Типы шаблонов тестирования. Unittest: установка и настройка, написание простых юнит-тестов, организация тестов в модули и пакеты, тестирование исключений. Pytest: установка и настройка, преимущества pytest перед unittest, фикстуры и параметризация тестов, настройка отчетов и логирование. Doctest: как написать doctests в документации, запуск doctests и интеграция с другими инструментами тестирования.

Модульное тестирование: цели и задачи. Покрытие кода: coverage.py, анализ покрытия кода и улучшение тестов. Процесс разработки через тестирование.

Рекомендуемая литература: [3, гл. 13].

Контрольные вопросы:

1. Что такое юнит-тестирование и какие инструменты используются для его проведения в Python?
2. Назовите отличия между unittest и pytest. Когда предпочтительнее использовать каждый из них?
3. Как организовать тесты в модули и пакеты в Python?
4. Что такое покрытие кода и как его измеряют с помощью coverage.py?
5. Опишите суть подхода Test Driven Development (TDD).

Тема 7. Параллельная обработка данных

Перечень изучаемых вопросов:

Введение в параллельную обработку данных. Основные подходы к параллельной обработке: многопоточное выполнение, многопоточность, асинхронное программирование.

Python и параллельное программирование: ограничения Python (GIL – Global Interpreter Lock), возможности Python для параллельного выполнения задач.

Многопоточная обработка: понятие потока (threads), создание и управление потоками в Python. Модуль threading. Синхронизация потоков: блокировки, семафоры, события. Проблемы многопоточной обработки. Производительность многопоточных приложений.

Многопроцессорная обработка. Понятие процесса, создание и управление процессами в Python. Модуль multiprocessing. Межпроцессовая коммуникация: каналы, очереди, менеджеры. Производительность и масштабируемость.

Библиотека AsyncIO. Основы асинхронного программирования. AsyncIO для работы в сети.

Рекомендуемая литература: [3, гл. 14].

Контрольные вопросы:

1. Что такое поток (thread) и процесс в контексте параллельной обработки данных? В каких случаях предпочтительно использовать каждый из них?
2. Как происходит межпроцессовая коммуникация в Python? Какие механизмы для этого используются?
3. В чем основное различие между многопоточным и асинхронным подходами к параллельной обработке данных?

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Особое место в структуре дисциплины занимает лабораторный практикум, включающий в себя девять лабораторных работ (таблица 2).

Таблица 2 – Объем (трудоемкость освоения) и структура ЛЗ

Номер темы	Содержание лабораторного занятия	Очная форма, ч	Заочная форма, ч
1	Введение в библиотеку NumPy	2	0.3
1	Манипуляция над данными с помощью пакета Pandas	6	0.4
1	Визуализация с помощью библиотеки Matplotlib	4	0.3
2	Методология CRISP DM	4	—
3	Работа с xlsx, sav, csv-файлами	6	1
3	Парсинг web-страниц с помощью запросов xpath и BeautifulSoup	4	4
3	Автоматизация работы с публичными данными, доступными через общедоступные API	6	4
4	Подключить PySpark и Kaggle в Google Colab	6	—
4	Знакомство с Spark API	10	—
ИТОГО:		48	10

Лабораторные занятия по дисциплине проводятся в компьютерных классах, оснащенных персональными компьютерами с программным обеспечением.

Студент в ходе лабораторного практикума согласно методическим указаниям и заданию преподавателя изучает методологии, при проведении проектов анализа данных осваивает методы работы с данными в разных форматах, способы сбора и обработки больших данных, а также изучает инструментария для визуализации результатов обработки. Защита лабораторной работы проводится на основании выполненного задания к лабораторной работе, оформленного отчета, а также ответа на контрольные вопросы.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа студентов по дисциплине, а также работа в ЭИОС университета может проводиться в компьютерных классах кафедры прикладной информатики, оснащенных персональными компьютерами с выходом в сеть Интернет.

Самостоятельная работа студента включает в себя освоение теоретического учебного материала (в том числе подготовка к лабораторным занятиям, оформление работ, подготовка к защите лабораторных работ).

Наряду с проработкой лекционного материала и подготовкой к лабораторным занятиям, студенты обеих форм обучения обязаны выполнить курсовой проект.

В качестве задания для курсового проекта выбираются (по указанию преподавателя) два вопроса из примерного перечня типовых заданий на курсовой проект по дисциплине (п. 7.2).

Выполнение курсового проекта способствует

- формированию навыков исследовательской деятельности;
- овладению приемами самостоятельной работы на основе полученных знаний и навыков;
- овладению навыками использования различных инструментальных средств.

Целью выполнения курсового проекта является закрепление, углубление и обобщение знаний по дисциплине.

Курсовой проект предоставляется в распечатанном или электронном виде. Структурными элементами курсового проекта являются: титульный лист, лист задания, лист содержания, введение, основная часть, заключение, библиографический список, приложения.

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЗАНЯТИЙ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

Для успешного освоения дисциплины необходимо ознакомиться с основными принципами объектно-ориентированного проектирования и программирования, изучить наиболее распространенные паттерны программирования.

При разработке образовательной технологии организации учебного процесса основной упор сделан на соединение активной и интерактивной форм обучения. Интерактивная форма позволяет студентам проявить самостоятельность в освоении теоретического материала и овладении практическими навыками, формирует интерес и позитивную мотивацию к учебе.

При изучении дисциплины внимание студентов постоянно акцентируется не только на теоретическом аспекте объектно-ориентированного подхода разработки программного обеспечения, но и его практического применения.

Для планирования работы студента в начале семестра производится выдача тем для самостоятельного изучения, определяются источники информации и график проведения текущего контроля. В качестве источников информации рекомендуется наряду с учебными пособиями использовать периодические издания (журналы) из области профессиональной деятельности.

В ходе лекционных занятий студенту следует вести конспектирование учебного материала. При самостоятельном изучении заданных преподавателем тем рекомендуется вносить основные материалы по ним в тот же конспект лекций в соответствии с рекомендованным порядком следования учебного материала.

При проведении занятий в интерактивной форме важно участвовать в процессе обсуждения и решения поставленных задач, задавать преподавателю вопросы с целью уяснения теоретических положений, области их применения, разрешения спорных ситуаций.

На лекциях и лабораторных занятиях изложению нового материала предшествуют обсуждение предыдущей темы с целью восстановления и закрепления студентами изученного теоретического и практического материала и ответы на вопросы студентов. В конце лекции выделяется время для ответов на вопросы по текущему материалу и его обсуждению. На лабораторных занятиях используется разбор конкретных методов создания программ в объектно-ориентированной парадигме. Активность студентов и проявленные знания при обсуждении материала учитываются при текущей и промежуточной (заключительной) аттестации по дисциплине.

Самостоятельная работа студентов проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- формирования умений использовать специальную литературу.

7. ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

7.1 Текущая аттестация

Текущая аттестация (текущий контроль) проводится с целью оценки освоения теоретического учебного материала, в том числе в рамках самостоятельной работы студента (п. 6).

Контроль на лекциях и лабораторных занятиях производится в виде тестирования или устного опроса.

Типовые контрольные вопросы для устного опроса по темам приведены в п. 3 настоящего пособия. Тестовые задания представлены в фонде оценочных средств (приложение к рабочей программе дисциплины).

Положительная оценка («зачтено») по результатам каждого контроля (опроса) выставляется в соответствии с универсальной системой оценивания, приведенной в таблице 3. В случае получения оценки «не зачтено» студент должен пройти повторный контроль по данной теме в ходе последующих консультаций.

Таблица 3 – Система оценок и критерии выставления оценки при прохождении контроля (опроса)

Критерий	Система оценок			
	«не зачтено»	«зачтено»		
1. Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект

Текущий контроль в виде защиты лабораторных работ проводится на лабораторном практикуме, целью которого является формирование умений и навыков по сбору, хранению и анализу больших данных. Оценка результатов выполнения задания по каждой лабораторной работе производится при представлении студентом отчета по лабораторной работе и на основании ответов студента на вопросы по тематике лабораторной работы. Студент, самостоятельно выполнивший задание, продемонстрировавший знание использованных им технических средств, алгоритмов и языков программирования задачи, получает по лабораторной работе оценку «зачтено».

С целью контроля качества самостоятельной работы студентов запланировано выполнение и защита курсового проекта.

Система оценивания и критерии оценки курсового проекта приведены в таблице 4.

Таблица 4 – Система оценивания и критерии оценки курсового проекта

Критерий	Система оценок			
	2	3	4	5
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
1 Работа с информацией	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
2 Научное осмысление изучаемого явления, процесса, объекта	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
3 Освоение стандартных алгоритмов решения профессиональных задач	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

7.2 Промежуточная аттестация по дисциплине

Промежуточная (заключительная) аттестация по дисциплине предусматривает проведение экзамена (экзаменационного тестирования).

К экзамену допускаются студенты:

- выполнившие и защитившие все предусмотренные лабораторные работы (получившие положительную оценку по результатам лабораторного практикума);
- выполнившие курсовой проект

Экзамен может проводиться как в традиционной форме, так и в виде тестирования.

Типовые задания на курсовой проект:

1. Строительство. Объекты – управление проектами.
2. Производство мебели. Контрагенты (поставщики / заказчики), цепочки поставки.
3. Туристический оператор. Туры, ценовые предложения.
4. Медицинские услуги. Регистратура – управление прайс-листом, занятость манипуляционных / операционных.
5. Логистика транспортных компаний. Управление автопарком, маршруты доставки, клиенты.
6. Фитнес-центр. Абонементы, тренировки, расписание занятий.
7. Онлайн-магазин. Каталог товаров, корзина покупок, система скидок.
8. Система управления библиотекой. Книги, читатели, бронирование книг.
9. Кинотеатр. Сеансы фильмов, продажа билетов, залы кинотеатра.
10. Университетская система. Курсы, студенты, преподаватели, оценки.
11. Интернет-провайдер. Клиенты, тарифы, счета за услуги.
12. Автомастерская. Запись на ремонт, запчасти, сотрудники.
13. Аэропорт. Рейсы, пассажиры, багаж, регистрация.
14. Банк. Счета, транзакции, кредиты, депозиты.
15. Отель. Номера, бронирование, гости, услуги.
16. IT-компания. Проекты, разработчики, задачи, дедлайны.
17. Рекрутинговая фирма. Вакансии, кандидаты, работодатели, собеседования.
18. Фотостудия. Залы, клиенты, фотографии, реквизит.

Экзаменационный билет содержит два экзаменационных вопроса.

Примерный перечень экзаменационных вопросов по дисциплине:

1. Назовите основные этапы жизненного цикла программного продукта
2. В чем заключается отличие между каскадной моделью и итеративным подходом?

Когда предпочтительнее использовать каждую из них?

3. Опишите особенности применения Agile-методов в разработке программного обеспечения. Каковы преимущества и ограничения этого подхода?

4. Какие риски могут возникнуть на разных этапах жизненного цикла программного продукта? Как можно минимизировать эти риски?

5. Перечислите основные типы диаграмм UML и кратко опишите их назначение.
6. Как используются диаграммы состояний в проектировании программного обеспечения?
7. Назовите три основных категории паттернов проектирования. Приведите примеры паттернов из каждой категории.
8. Почему соблюдение принципов SOLID способствует лучшему использованию паттернов проектирования?
9. Класс в Python. Жизненный цикл объектов в Python.
10. Принцип наследование в ООП. Множественное и одиночное наследование.
11. Структуры данных. Встроенные структуры данных в Python.
12. Обработка исключений в Python.
13. Абстрактный класс в Python.
14. Что такое юнит-тестирование, и какие инструменты используются для его проведения в Python?
15. Модульное тестирование в Python.
16. Опишите суть подхода Test Driven Development (TDD).
17. Многопоточная обработка.
18. Многопроцессорная обработка.
19. Библиотека AsyncIO. Основы асинхронного программирования.

Тестовые задания для проведения экзаменационного тестирования приведены в фонде оценочных средств по дисциплине.

Система оценивания и критерии выставления оценок по экзамену (экзаменационному тестированию) приведена в таблице 5.

Таблица 5 – Система оценок и критерии выставления оценки по экзамену (экзаменационному тестированию)

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
1. Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полной полнотой знаний и системным взглядом на изучаемый объект
2. Работа с	Не в состоянии	Может найти	Может найти,	Может найти,

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
информацией	находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	необходимую информацию в рамках поставленной задачи	интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
3. Научное осмысление изучаемого явления, процесса, объекта	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
4. Освоение стандартных алгоритмов решения профессиональных задач	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

8. ЗАКЛЮЧЕНИЕ

Использование объектно-ориентированного подхода стало стандартом де-факто в разработке современного ПО. Этот метод позволяет создавать надежные, гибкие и поддерживаемые системы, которые легко адаптируются под изменяющиеся требования рынка и бизнеса.

Современные IT-системы часто требуют реализации сложной логики обработки данных, взаимодействия с внешними сервисами и управления множеством взаимосвязанных сущностей. ООП предоставляет инструменты для создания моделей, отражающих реальные объекты и процессы, что существенно упрощает разработку таких систем. Использование паттернов проектирования позволяет реализовать на базе ООП событийно-ориентированный подход программирования. Это открывает огромные возможности для создания эффективных, гибких и масштабируемых IT-решений.

Освоение дисциплины «Технология объектно-ориентированного проектирования и программирования» является одним из основополагающих шагов к формированию будущего специалиста в области прикладной информатики. Приобретенные в ходе изучения дисциплины знания, умения и навыки будут углубляться и совершенствоваться в процессе дальнейшего обучения и могут быть применены в профессиональной деятельности.

9. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

1. Крючкова, Е. Н. Объектно-ориентированное программирование: Архитектурное проектирование и паттерны программирования: учеб.-метод. пособие / Е. Н. Крючкова, С. М. Старолетов. – Барнаул: АлтГТУ, 2020. – 180 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/292790> (дата обращения: 30.05.2024). – Режим доступа: для авториз. пользователей.
2. Розенберг, Д. Применение объектного моделирования с использованием UML и анализ прецедентов [Электронный ресурс] / Д. Розенберг, К. Скотт; пер. с англ. – Москва: ДМК Пресс, 2007. – 160 с. <https://e.lanbook.com/book/1226/>.
3. Лотт, С. Объектно-ориентированный Python. Надежные объектно-ориентированные приложения и библиотеки на языке Python / С. Лотт, Д. Филлипс; 4-е изд. – Санкт-Петербург: Питер, 2024. – 704 с. ISBN 978-5-4461-1995-0.

Дополнительная литература

1. Кальб, Ирв. Объектно-ориентированное программирование с помощью Python / Ирв Кальб; перевод с англ. М. А. Райтмана. – Москва: Эксмо, 2024. – 512 с. – (Мировой компьютерный бестселлер). ISBN 978-5-04-186627-3.

Интернет-ресурсы

1. ГОСТ Р ИСО/МЭК 12207–2010. Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств. URL: <https://docs.cntd.ru/document/1200082859>.

2. Agile-манифест разработки программного обеспечения. URL: <https://agilemanifesto.org/iso/ru/manifesto.html> .

Локальный электронный методический материал

Татьяна Валентиновна Снытникова

ТЕХНОЛОГИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОЕКТИРОВАНИЯ И ПРОГРАММИРОВАНИЯ

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 1,5. Печ. л. 1,4.

Издательство федерального государственного бюджетного образовательного
учреждения высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1