**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Проектирование программного обеспечения

Software Design

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 053904

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Цель изучения дисциплины: ознакомление студентов с основными принципами проектирования крупных программных продуктов и комплексов.  
Задачи обучения:  
1. Приобретение знаний об архитектуре программного обеспечения в целом, роли архитектуры в жизненном цикле программного обеспечения, целях и содержании деятельности архитектора.  
2. Приобретение навыков объектно-ориентированного проектирования, проектирования пользовательских интерфейсов, проектирования распределённых и многопоточных приложений.  
3. Изучение основных архитектурных стилей, типовых приёмов проектирования.  
4. Получение представления о языке UML и других формальных языках как основных средствах описания архитектуры ПО.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для студентов 4 курса бакалавриата, освоивших программу курсов «Программирование» и «Информатика» (или аналогичных им).  
Максимальная эффективность программы будет обеспечена при условии, что студент имел опыт разработки промышленных или исследовательских программных продуктов.

**1.3. Перечень результатов обучения (learning outcomes)**

Результатами обучения являются знание содержания программы курса и представление о возможностях применения ее разделов в практической деятельности, а именно:  
1. Умение самостоятельно выбрать подход к проектированию программной системы, создать или описать имеющуюся архитектуру программного продукта.  
2. Знание достоинств и недостатков основных существующих подходов к проектированию ПО.  
3. Владение типовыми шаблонами проектирования объектно-ориентированных программ.  
4. Владение языком UML на уровне, достаточном для создания архитектурных диаграмм и понимания диаграмм, созданных коллегами.  
5. Умение оформить и изложить результат проектирования программного обеспечения в виде архитектурного документа или набора документов.  
6. Владение принципами проектирования надёжных и отказоустойчивых систем.  
7. Владение принципами проектирования распределённых приложений.  
8. Владение принципами проектирования многопоточных приложений.

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Основной формой обучения проектированию программного обеспечения являются лекционные занятия в аудитории.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 7 | 30 |  | 2 |  |  |  |  | 2 | 2 |  |  |  | 43 |  | 29 |  | 4 | 3 |
|  | 2-100 |  | 2-100 |  |  |  |  | 2-100 | 2-100 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  | 2 |  |  |  |  | 2 | 2 |  |  |  | 43 |  | 29 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обученияочная | | | | | | |
| Семестр 7 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1 | 1. Введение в проектирование ПО.    1. Программа и программное обеспечение. Понятие архитектуры. Роль проектирования и архитектора в производстве ПО. Архитектура и качество ПО. Архитектура и жизненный цикл ПО    2. Архитектура и требования. Декомпозиция. Шаблон “Model-View-Controller”. Базовые принципы проектирования. Восходящее и нисходящее проектирование. Design Document. | лекции | 4 |
| по методическим материалам | 4 |
| 2 | 1. Объектно-ориентированное программирование    1. Объектно-ориентированный подход. Состояние, поведение и идентичность объекта. Принципы объектно-ориентированного проектирования. Наследование и композиция. Принципы SOLID. Закон Деметры. | лекции | 2 |
| по методическим материалам | 2 |
| 3 | 1. Язык UML    1. Модели в различных областях человеческой деятельности. Роль моделирования в процессе разработки. UML и его назначение. Функциональные требования и диаграммы использования. Система, действующие лица и варианты использования. Включение и расширение. Диаграммы активностей.    2. Диаграммы классов и моделирование предметной области. Моделирование взаимодействия объектов. Способы моделирования свойств. Ассоциации. Обобщение. Зависимости. Диаграммы компонент и диаграммы размещения. UML в процессе разработки. Моделирование данных.    3. Диаграммы состояний и их применение для описания поведения реактивных систем. Диаграммы последовательности и взаимодействия. | лекции | 6 |
| по методическим материалам | 6 |
| 4 | 1. Проектирование пользовательских интерфейсов    1. Проектирование слоя UI. Вопросы удобства использования. Сбор статистики использования элементов UI и метрики. | лекции | 2 |
| по методическим материалам | 2 |
| 5 | 1. Шаблоны проектирования    1. Архитектурные шаблоны и стили. Понятие архитектурного шаблона и стиля. Многоуровневая архитектура. Pipes and filters. Интерпретатор. Blackboard. Publish-Subscribe. Событийно-ориентированные системы.    2. Структурные шаблоны. Рекурсивная композиция, использование полиморфизма и композиции при едином базовом интерфейсе. Шаблоны Composite и Decorator. Выделение алгоритма в объект, шаблон Strategy. Сокрытие реализации, шаблоны Wrapper и Adapter. Проектирование удаленного и отложенного выполнения. Шаблон Proxy. Оптимизация числа объектов. Шаблон Flyweight.    3. Порождающие шаблоны. «Ленивая» инициализация, шаблон Singleton. Клонирование объектов, шаблон Prototype. Фабрика, шаблоны Abstract Factory и Factory Method. Поэтапное конструирование объекта, шаблон Builder.    4. Поведенческие шаблоны. Шаблон «Chain of Responsibility». Шаблон «Context». Архитектуры, управляемые событиями. Шаблон «Listener». Отложенное выполнение. Шаблон «Command». Инкапсуляция поведения. Шаблон «Strategy». Подходы к реализации поведения, основанного на модели конечного автомата. Шаблон «State». Шаблон «Visitor». | лекции | 8 |
| по методическим материалам | 8 |
| 6 | 1. Проектирование распределённых приложений.    1. Шаблоны интеграции приложений. Особенности интеграционных решений. Способы интеграции приложений. Message-oriented middleware. Шаблоны интеграционных решений.    2. Проектирование распределённых приложений. Задача сетевого взаимодействия. Семиуровневая модель OSI. Internet Protocol (IP). Transport Control Protocol (TCP). Процедура открытия соединения в TCP. Состояния TCP соединения. Сокеты. Проектирование сетевого взаимодействия на основе TCP.   Блокирующее и неблокирующее чтение. Кодирование сообщений в поток. Организация транспортных сессий. Контроль разрыва соединения. Контроль чтения и записи. Решение типовых проблем. | лекции | 4 |
| по методическим материалам | 4 |
| 7 | 1. Проектирование надёжных приложений, тестирование.    1. Качество и тестирование. Что такое тестирование. Психология тестирования. Тестирование в процессе разработки. Типы тестов. Unit-тесты. TDD. Динамическое и статическое тестирование. Функциональное и структурное тестирование, метод «черного» и «белого» ящика. Принципы функционального тестирования. Разбиение на классы эквивалентности. Анализ граничных условий. Отрицательное тестирование. Дефекты и управление дефектами. Составление плана тестирования. | лекции | 2 |
| по методическим материалам | 2 |
| 8 | 1. Проектирование многопочных приложений    1. Метрики производительности приложения. Потоки и процессы. Проблема параллельного доступа. Атомарные операции. Блокировка доступа к ресурсу. Модель памяти многопоточного приложения. Примитивы синхронизации. Монитор. Семафор. Барьер. Типовые задачи параллельного программирования. Производители и потребители. Читатели и писатели. Ключевое слово volatile. «Облегченная» синхронизация. Проблема «ленивой» инициализации в случае нескольких потоков. Антишаблон Double-Checked Locking. Шаблоны проектирования многопоточных приложений. Шаблоны «Thread Local», «Scheduler», «Thread Pool». | лекции | 2 |
| по методическим материалам | 2 |
| 9 | Экзамен | промежуточная аттестация | 2 |

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Посещение практических занятий, выполнение домашних заданий.

**3.1.2 Методическое обеспечение самостоятельной работы**

Основная и дополнительная литература.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Устный экзамен проводится в соответствии с заранее опубликованным списком вопросов, покрывающих рассмотренные на лекционных занятиях темы. Студент должен в ходе экзамена ответить на два вопроса, на подготовку ответа на которые даётся не менее 40 минут, а также на несколько дополнительных вопросов, без подготовки. Количество и содержание дополнительных вопросов – на усмотрение преподавателя, принимающего экзамен. Каждый ответ оценивается по шкале от 0 (нет ответа) до 5 (очень хороший ответ), результирующая оценка получается следующим образом:

1. Оценки за ответы на два основных вопроса усредняются, результат усреднения делится на два.
2. Оценки за ответы на дополнительные вопросы усредняются, результат усреднения делится на два и складывается с оценкой, полученной в п.1.
3. Если результирующая оценка
   1. в диапазоне от 2.5 до 3.5, за экзамен ставится «удовлетворительно»;
   2. в диапазоне от 3.5 до 4.5, за экзамен ставится «хорошо»;
   3. в диапазоне от 4.5 до 5, за экзамен ставится «отлично».

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

***Пример списка вопросов для устного экзамена:***

1. Понятие архитектуры, роль архитектуры в жизненном цикле ПО
2. Понятие декомпозиции. Модульность, информационная закрытость, связность, сцепление, сложность.
3. Понятия класса и объекта, принципы выделения абстракций предметной области.
4. Принципы SOLID.
5. Моделирование, визуальные модели.
6. Анализ и моделирование требований, диаграмма случаев использования UML, диаграмма активностей UML.
7. Проектирование структуры системы, диаграмма классов UML, диаграмма объектов UML.
8. Диаграммы пакетов UML, диаграммы компонентов UML, диаграммы размещения UML.
9. Моделирование данных, языки ER, ORM.
10. Моделирование поведения, диаграммы конечных автоматов UML.
11. Диаграммы активностей UML, временные диаграммы UML.
12. Диаграммы последовательностей UML, диаграммы коммуникации UML.
13. Проектирование пользовательских интерфейсов: понятие User Experience, моделирование персонажей.
14. Методы прототипирования пользовательских интерфейсов.
15. Понятие архитектурного стиля, трёхзвенная архитектура, Model-View-Controller.
16. Структурный и объектно-ориентированный стили, уровневые архитектурные стили.
17. Пакетная обработка, каналы и фильтры, Blackboard.
18. Publish-Subscribe, Event-based-стили, мобильный код, интерпретатор.
19. Паттерны «Компоновщик», «Декторатор», «Стратегия».
20. Паттерны «Адаптер», «Прокси», «Фасад».
21. Паттерны «Фабричный метод», «Шаблонный метод», «Абстрактная фабрика».
22. Паттерны «Одиночка», «Прототип», «Строитель».
23. Паттерны «Посредник», «Команда», «Цепочка ответственности».
24. Паттерны «Наблюдатель», «Состояние», «Посетитель», «Хранитель».
25. Методы интеграции приложений: обмен файлами, общая база данных, RPC.
26. Взаимодействие через обмен сообщениями, понятие канала.
27. Маршрутизаторы сообщений, фильтры сообщений, разветвители, агрегаторы, преобразователи порядка.
28. Брокеры сообщений, преобразователи сообщений, понятие канонической модели данных, Enterprise Service Bus.
29. Архитектура распределённых систем: понятие распределённой системы, виды сущностей и взаимодействий.
30. Архитектуры распределённых систем: Layered, Tiered, тонкий клиент.
31. Межпроцессное сетевое взаимодействие, модель OSI.
32. Стек протоколов TCP/IP, marshaling, протоколы «запрос-ответ».
33. Протокол HTTP.
34. Удалённые вызовы процедур (RPC).
35. Удалённые вызовы методов (RMI).
36. Веб-сервисы, SOAP, REST.
37. Понятие тестирования, виды тестирования.
38. Тестирование в процессе разработки. Unit-тесты. TDD.
39. Динамическое и статическое тестирование. Функциональное и структурное тестирование, метод «черного» и «белого» ящика. Дефекты и управление дефектами.
40. Составление плана тестирования.
41. Потоки и процессы. Проблема параллельного доступа.
42. Атомарные операции. Модель памяти многопоточного приложения.
43. Примитивы синхронизации: Монитор. Семафор. Барьер.
44. Типовые задачи параллельного программирования: Производители и потребители. Читатели и писатели. «Облегченная» синхронизация.
45. Проблема «ленивой» инициализации в случае нескольких потоков. Антишаблон Double-Checked Locking.
46. Шаблоны проектирования многопоточных приложений: «Thread Local», «Scheduler», «Thread Pool».

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

**Анкета для оценки обучающимися качества учебного процесса**

1. Статус обучающегося: студент/магистрант/аспирант факультета, курса
2. ФИО преподавателя
3. Как вы оцениваете содержание курса:
   1. Полезно, интересно
   2. Не очень полезно, рассказываются и так известные вещи
   3. Другое (укажите, что именно)
4. Как вы оцениваете домашние задания:
   1. Слишком высокая нагрузка
   2. Низкая нагрузка, можно было задавать больше
   3. В самый раз
   4. Комментарий в свободной форме относительно домашних заданий:
5. Как Вы оцениваете работу преподавателя?
   1. Очень высоко
   2. Высоко
   3. Средне
   4. Низко   
      Что именно не нравилось:
6. Ваши пожелания, соображения и/или рекомендации

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К проведению лекционных занятий должны привлекаться преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Не требуется

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Требуется стандартно оборудованная аудитория.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Доска для письма маркером, мультимедийный проектор.

**3.3.3 Характеристики специализированного оборудования**

Не требуется.

**3.3.4 Характеристики специализированного программного обеспечения**

Не требуется.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Маркеры для доски, губка

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы.- 2-е изд. - СПб. ; М. : Символ-Плюс, 2007. - 298 с.  
4. Приемы объектно-ориентированного проектирования] = Паттерны проектирования : производственно-практическое издание / Э. Гамма [и др.]. - М. ; СПб. ; Нижний Новгород : Питер, 2008-2016. - 366 с. : ил. - (Библиотека программиста).

**3.4.2 Список дополнительной литературы**

К. И. Вигерс, Дж. Битти. Разработка требований к программному обеспечению. БХВ-Петербург, 2016, 736 С.

**3.4.3 Перечень иных информационных источников**

Не требуются.

**Раздел 4. Разработчики программы**

Литвинов Юрий Викторович, кандидат технических наук, доцент кафедры системного программирования, y.litvinov@spbu.ru  
Брыксин Тимофей Александрович, кандидат технических наук, доцент кафедры системного программирования, t.bryksin@spbu.ru