

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗКОЙ РЕСПУБЛИКИ

КЫРГЫЗСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. И. РАЗАКОВА

Институт информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

«УТВЕРЖДАЮ»

Декан

ФИТ

(факультет/институт)

Кабаева Г.Дж.

(подпись)

« ____ » _____ 20 ____ г.

РАБОЧАЯ ПРОГРАММА ПО ДИСЦИПЛИНЕ

Б1.3.П2 ПРОЕКТИРОВАНИЕ ПО П»

(код, название)

<u>Направление:</u>		710400 «Программная инженерия»,	
<u>Профиль:</u>		Технология командной разработки ПО	
<u>Квалификация:</u>		бакалавр	
<u>Форма обучения:</u>		очная	
<u>Семестр</u>		3	
Всего кредитов		4	В часах: 120
Аудиторных, из них:			64
Лекции			-
Лабораторные/ Практические/ Семинарские			64
СРС			56
Форма отчетности			экзамен

Бишкек, 2022 г.

Лист согласования

Рабочая программа по дисциплине «**Проектирование ПО II**» разработана в соответствии с требованиями ГОС ВПО по подготовки бакалавров и предназначена для студентов, обучающихся по направлению 710400 «Программная инженерия» профилю/программе Технология командной разработки ПО

Автор/ы (составитель/и): Мусина И.Р.

Процесс рассмотрения и утверждения РПД	№ протокола	Подписи (печать)
<p>Рабочая программа дисциплины рассмотрена на заседании кафедры _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____</p> <p>от « _____ » _____</p> <p>20__ г.</p>	<p>Зав. профилирующей кафедры: _____</p> <p>(подпись)</p> <p>Ф.И.О. _____</p>
<p>*Рабочая программа дисциплины рассмотрена/согласована на заседании кафедры _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____</p> <p>от « _____ » _____</p> <p>20__ г.</p>	<p>Зав. не/профилирующей кафедры: _____</p> <p>Ф.И.О. _____</p>
<p>Рабочая программа дисциплины одобрена руководителем ООП по направлению _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>Дата: _____</p>	<p>Руководитель ООП: _____</p> <p>(подпись)</p> <p>Ф.И.О. _____</p>
<p>Рабочая программа дисциплины согласована на заседании Учебно-методической комиссии факультета/института _____</p> <p>_____</p> <p>(наименование учебного подразделения)</p>	<p>протокол № _____</p> <p>от « _____ » _____</p> <p>20__ г.,</p>	<p>Председатель УМК: _____</p> <p>(подпись)</p> <p>Ф.И.О. _____</p>
<p>**Рабочая программа дисциплины согласована _____ (или обсуждалась/рецензирована)</p> <p>_____</p> <p>(указать наименование предприятия/учреждения/организации)</p>	<p>Дата: _____</p> <p>согласования/обсуждения/рецензия</p>	<p>(должность) _____</p> <p>(подпись)</p> <p>Ф.И.О. _____</p>

Лист изменений и дополнений в РПД

№ п/п	Номер и название раздела РПД	Описание изменений/дополнений в РПД	Дата изменений	№ протокола заседания кафедры	Подписи (печать) преподавателя, зав. кафедрой
					<hr/> <hr/>

1. Аннотация дисциплины

Дисциплина «Проектирование ПО II» относится к обязательным дисциплинам вариативной части профессионального цикла. Она читается в весеннем семестре студентам второго курса.

Предметом изучения дисциплины «Проектирование ПО II» являются теоретические и практические вопросы проектирования сложных программных систем с использованием объектно-ориентированного подхода на основе применения диаграмм унифицированного языка UML. Кроме того, в рамках данного курса изучаются модели жизненного цикла ПО, вопросы анализа и формирования требований.

По итогам изучения дисциплины студенты сдают экзамен. Текущий контроль и самоконтроль усвоения курса осуществляется посредством выполнения студентами лабораторных и самостоятельных работ. Рубежный контроль – письменная работа (модуль I) и командная разработка ПО, включающая разработку требований и проектирование (модуль II)

2. Цели и задачи дисциплины

Главная цель освоения дисциплины – теоретическая и практическая подготовка студентов в области проектирования ПО.

Целями дисциплины являются закрепление и расширение будущими программными инженерами знаний об этапах жизненного цикла создаваемого программного обеспечения и подходах к созданию ПО, полученных при обучении дисциплины «Проектирование ПО I»; изучение объектно-ориентированного подхода к проектированию ПО.

Основные задачи дисциплины:

- Изучение языка моделирования UML;
- Приобретение практических навыков по анализу и формирований требований, а также проектированию ПО с помощью графических языков программирования (UML).

3. Пререквизиты и постреквизиты

Перечень дисциплин, усвоение которых необходимо для изучения заданной дисциплины:

- ✓ «Информатика»;
- ✓ «Проектирование ПО I»;

Постреквизиты курса:

- Б1.3.3 Процессы проектирования и разработки ПО;
- Б3.9 Проектирование ПО III.

4. Перечень компетенций, которыми должен овладеть бакалавр при изучении дисциплины данной ООП (знать, уметь, навыки)

В результате изучения дисциплины студент должен

будет знать:

- Основные этапы жизненного цикла ПО;
- диаграммные методологии проектирования ПО;
- язык моделирования UML;
- методологию объектно-ориентированной разработки;
- дополнительные средства поддержки жизненного цикла разработки ПО (CASE-технологии);

будет уметь

- проводить анализ предметной области;
- составлять техническое задание, спецификации требований;

- проектировать ПО и организовывать процесс конструирования ПО, осуществлять выбор программных и инструментальных средств для разработки, создания и отладки программного обеспечения;
- самостоятельно приобретать необходимые знания из предметной области;
- применять язык UML для построения моделей анализа и проектирования ПО;

будет владеть

- пониманием современных информационных технологий;
- пониманием тенденций развития современных ПО, особенностей их работы в условиях профессиональной деятельности;
- диаграммными методологиями проектирования программного обеспечения;
- навыками использования языка UML;
- CASE-средствами проектирования программного обеспечения.

В процессе освоения дисциплины у студентов развиваются следующие компетенции (согласно ГОСТ ВПО КР 2021г.):

Профессиональные:

-аналитическая деятельность:

- способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта (ПК-1);

-проектная деятельность:

- способен применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ПК-2);
- способен читать, понимать и выделять главную идею прочитанного исходного кода, документации (ПК-3);
- способен моделировать объектов (ПК-4);
- способен создавать программные интерфейсы (ПК-5).

-производственно-технологическая деятельность:

- способен использовать операционные системы, сетевые технологии, средства разработки программного интерфейса, применять языки и методы формальных спецификаций, систем управления базами данных (ПК-6);
- способен применять основные методы и инструменты разработки программного обеспечения (ПК-7);
- способен понимать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования) (ПК-8);

организационно-управленческая деятельность:

- способен понимать модели жизненного цикла, методы управления процессами разработки требований, проектирования, конструирования, тестирования, эволюции и сопровождения (ПК-9);

-сервисно - эксплуатационная деятельность:

- способен понимать основные концепции и модели эволюции и сопровождения программного обеспечения, особенности эволюционной деятельности с технической точки зрения, реинженеринг и рефакторинг (ПК-10).

Тематический план лабораторных занятий

Целью лабораторных занятий является получение бакалаврами практических навыков анализа требований, проектирования, а также реализация программного обеспечения.

№	Наименование работ	Содержание	Кол-во час.	Примечание
Модуль 1				
1	Разработка ТЗ на программный продукт	1. Проанализировать автоматизируемый объект; 2. Определить бизнес-требования; 3. Определить функциональные и нефункциональные требования.	4	Показать документ, подготовленный в WORD
2	Построение диаграммы USE CASE.	1. Выделить варианты использования и действующие лица. 2. Установить связи между вариантами использования.	4	Использовать одно из CASE-средств (MS Visio, он-лайн UMLредактор)
3	Построение диаграммы USE CASE со связями обобщения между актерами и между вариантами использования	В диаграмме USE CASE установить связи обобщения между актерами и между вариантами использования	4	Использовать одно из CASE-средств (MS Visio, он-лайн UMLредактор)
4	Описание вариантов использования	Описать каждый вариант использования из диаграммы использования (UCM)	4	Использовать шаблон для описания
5	Алгоритмическое проектирование. Построение диаграммы деятельности	Построить диаграммы деятельности для каждого варианта использования	4	Использовать одно из CASE-средств
6	Построение диаграммы деятельности для параллельных процессов	Построить диаграммы с распараллеливанием и слиянием процессов	4	Использовать одно из CASE-средств (MS Visio, он-лайн UMLредактор)
7	Построение диаграммы деятельности с дорожками	Построить диаграммы деятельности с распределением деятельностей по различным дорожкам (модулям и актерам)	4	Использовать одно из CASE-средств (MS Visio, он-лайн UMLредактор)
8	Модульная работа: разработка ТЗ и проектирование для контрольного примера	1. Разработать ТЗ 2. Построить диаграммы: варианты использования и деятельности.	4	Использовать одно из CASE-средств (MS Visio, он-лайн UMLредактор)
Модуль 2				
9	Проектирование архитектуры. Организация классов. Построение диаграммы классов. Атрибуты, методы, связи	Конструирование классов. Определение атрибутов и методов. Определение отношений между классами: ассоциация, зависимость, агрегация, композиция, обобщение	4	Использовать одно из CASE-средств

10	Построение диаграммы последовательности	Построение взаимосвязи между объектами классов через передачу сообщений	4	Использовать одно из CASE-средств
11	Построение диаграммы пакетов.	Объединение классов в пакеты. Установление связи между пакетами	4	Использовать одно из CASE-средств
12	Построение диаграммы состояний	Определение состояний, переходов, триггеров. Построение диаграммы состояний.	4	Использовать одно из CASE-средств
13	Проектирование БД	Спроектировать БД. Построение логической и физической модели БД.	4	Использовать любое средство моделирования БД (Visio, All fusion и т.д.)
14	Программная реализация проекта	Провести кодирования, используя любой язык программирования.	4	Использовать любое средство для кодирования
15	Тестирование программы. Разработка документации	Тестирование разработанного программного продукта методом черного ящика. Отладка. Разработка документации на продукт	4	Использовать любое средство для кодирования
16	Защита проекта (по самостоятельной работе)	Подготовить отчет по самостоятельной работе с демонстрацией программы	4	Отчет должен быть подготовлен в виде текстового документа.
ИТОГО			64	

5. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА

Темы СРС (в часах) - непрерывный процесс при кредитной системе обучения, нацеливающий на целенаправленное получение новых знаний, умений, навыков (компетенций) без непосредственного участия в этом процессе преподавателя.

№	Темы занятий	Задания на СРС	Цель и содержание заданий	Реком. литература	Форма контроля	Срок и сдачи	Макс балл
Модуль 1							
1	Модели жизненного цикла ПО (4 час.)	Рассмотреть различные виды ЖЦ ПО, выявить их достоинства и недостатки	Студент должен знать в каких случаях применять ту или иную модель ЖЦ ПО	[1]	Отчет по работе.	2-3 недели	2
2	Agile (гибкая) разработка ПО (4 час.)	Описать особенности гибкой разработки	Студент должен знать в каких условиях и как применять гибкую разработку ПО	[1,2]	Отчет по работе.	2-3 недели	2

3	Разработка технического задания для индивидуального проекта (6 час).	Разработать документ ТЗ для индивидуального проекта, согласованного с преподавателем. Построить UCM.	Студент должен уметь самостоятельно анализировать и выстраивать требования для проекта по разработке ПО	[6]	Отчет с предоставлением UCM в среде для построения UML диаграмм	4-5 недели	3
4	Документирование вариантов использования для индивидуального проекта (8 час)	Провести описание вариантов использования по шаблону в виде таблицы	Студент должен уметь самостоятельно правильно документировать варианты использования ПО	[7]	Отчет	6-7 недели	4
5	Построение диаграммы деятельности для индивидуального проекта (6 час).	Построить диаграммы деятельности для индивидуального проекта с использованием CASE-технологий	Умение самостоятельно применить CASE-технологии для алгоритмического проектирования	[3, 6]	Отчет с предоставлением диаграмм	6-8 недели	4
Итого по модулю I (28ч.)							15
Модуль 2							
1	Построение диаграммы классов для индивидуального проекта (4 час)	Построить диаграмму классов с использованием CASE-технологий	Умение самостоятельно применить CASE-технологии для проектирования классов для реального объекта	[3, 7]	Отчет с предоставлением диаграмм	9-10 недели	3
2	Проектирование архитектуры ПО для индивидуального проекта (4 час)	Построение диаграммы пакетов с использованием CASE-технологий	Умение самостоятельно применить CASE-технологии для проектирования архитектуры ПО реального объекта	[3,2,7]	Отчет с предоставлением диаграмм	11-12 недели	3
3	Разработка модели данных для индивидуального проекта (4 час)	Построение диаграммы ERD для разрабатываемого ПО с использованием CASE-технологий	Умение самостоятельно анализировать данные и моделировать БД для реального объекта	[3,7]	Отчет с предоставлением диаграмм	13-14 недели	2

4	Построение диаграмм последовательности и состояний для индивидуального проекта (6 час)	Построить диаграммы последовательности и состояний с использованием CASE-технологий	Студент должен уметь самостоятельно строить диаграммы последовательности и состояний для реального объекта	[1,2]	Отчет с предоставлением диаграмм	15-16 недели	3
5	Реализация (кодирование) для индивидуального проекта (10 час)	Реализовать проект в виде работающего программного продукта	Студент должен уметь самостоятельно реализовать проект и провести его валидацию	[1,2]	Полный отчет в рукописном виде	15-16 недели	4
Итого по модулю II:28 час.							15

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

ВОПРОСЫ К МОДУЛЮ I

1. CASE-технология. CASE-средства. CASE-системы.
2. Классификация CASE-средств.
3. Жизненный цикл программного обеспечения. Модели жизненного цикла программного обеспечения.
4. Причины возникновения ошибок при разработке программных средств.
5. Процесс разработки программного обеспечения с использованием CASE-средств.
6. Этап анализа в жизненном цикле программного обеспечения.
7. Методологические аспекты анализа целей и требований к разрабатываемому программному обеспечению.
8. Функционально-ориентированное (структурное) проектирование программного обеспечения и его особенности.
9. Диаграммные методологии проектирования программного обеспечения.
10. CASE-средства поддержки структурных методологий.
11. Методики объектно-ориентированного анализа и проектирования.
12. Классификация, основные этапы и задачи объектно-ориентированных методов анализа и проектирования.
13. Язык моделирования UML. Синтаксис. Семантика.
14. Язык моделирования UML – диаграммы вариантов использования и сценарии.
15. Связи в диаграмме вариантов использования.
16. Язык моделирования UML – диаграммы деятельности.
17. Дорожки в диаграмме деятельности.

ВОПРОСЫ К МОДУЛЮ II

1. Язык моделирования UML – диаграммы классов и их использование.
2. Связи в диаграмме классов.
3. Отношение ассоциации.
4. Отношения агрегации и композиции.
5. Отношение обобщения.
6. Отношения зависимости.
7. Диаграмма пакетов.
8. Язык моделирования UML – диаграмма последовательности.

9. Отношения в диаграмме последовательности.
10. Синхронные и несинхронные сообщения.
11. Язык моделирования UML – диаграмма кооперации.
12. Язык моделирования UML – диаграммы состояний.
13. Отношения в диаграмме состояний.
14. Триггерные и нетриггерные переходы.
15. Язык моделирования UML – диаграммы компонентов и диаграммы развертывания.
16. Методология объектно-ориентированной разработки RUP (Ration Unified Process).
17. Чистота кода.
18. Проектирование БД.
19. Тестирование ПО.

7. РЕЙТИНГОВЫЙ ЛИСТ ОЦЕНКИ ЗНАНИЙ СТУДЕНТОВ (с разбивкой по баллам по видам и формам оценивания)

Успешность изучения дисциплины в системе кредитных технологий оценивается суммой набранных баллов (из 100 возможных)

Количество модулей	Объем дисциплины в часах	Оценка в баллах		Сроки текущего контроля
		min	max	
1	Лаб. –32 ч.	10	15	8 неделя
	Сам. работа -28час	10	15	4 семестр
	Итого	20	30	
2	Лаб. –32ч.	10	15	16 неделя
	Сам. работа -28час.	10	15	4 семестр
	Итого	20	30	
Итоговый контроль		21	40	по расписанию экзаменов
	Сумма баллов	61	100	

8. ПЕРЕЧЕНЬ УЧЕБНОЙ ЛИТЕРАТУРЫ И УЧЕБНО-МЕТОДИЧЕСКИХ МАТЕРИАЛОВ ДЛЯ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ.

Основная литература

1. А. Орлов., Б.Я.Цилькер. «Технологии разработки программного обеспечения». Учебник. - СПб: Питер, 2016.
2. Йан Соммервилл «Инженерия программного обеспечения». Издат. Дом Москва, 2002г. (или на англ. яз. Изд- е от 2016г.)
3. Мусина И.Р., Семенов А.С. Процессы проектирования и разработки программного обеспечения: Учебно – методическое пособие /КГТУ им. И.Раззакова, Бишкек: ИЦ «Текник», 2016.-64с.

Дополнительная литература

4. Микки Госсе, Брайан Келлер, Аджой Кришнамурти и т.д. «Управление жизненным циклом приложений с Visual Studio 2010. - 2012.
5. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. — СПб.: Питер, 2018. — 352 с.: ил. — (Серия «Библиотека программиста»). ISBN 978-5-4461-0772-8/
6. Л.Г. Гагарина, Е.В. Кокорева, Б.Д.Виснадул. Технология разработки программного обеспечения. Москва, 2008.

7. М. М. Меженная, Т. В. Гордейчук, М. М. Борисик, О. С. Медведев, И.Ф. Киринович. Тестирование, оценка программного обеспечения. Учебно-методическое пособие.– Минск: БГУИР, 2016. – 64 с. : ил.

Интернет-ресурсы

8. Интернет Университет Информационных Технологий – <http://www.intuit.ru/>