

«

»

“ ”

“ ”

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Технология разработки программного обеспечения**

: 09.04.01

: 1, : 12

		1	2
1	()	2	3
2		72	108
3	, .	42	28
4	, .	18	0
5	, .	18	0
6	, .	0	18
7	, .	18	6
8	, .	2	2
9	, .	4	8
10	, .	30	80
11	(, ,)		
12			

(): 09.04.01

1420 30.10.2014 . , : 25.11.2014 .

: 1,

(): 09.04.01

, 6 20.06.2017

, 6 21.06.2017

:

,

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОК.1 способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ОК.5 использованием на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ОК.6 способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ОК.8 способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ОПК.1 способность воспринимать математические, естественнонаучные, социально-экономические и профессиональные знания, умением самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе в новой или незнакомой среде и в междисциплинарном контексте; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ОПК.5 владение методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях; <i>в части следующих результатов обучения:</i>
1.
Компетенция ФГОС: ПК.19 способность к применению современных технологий разработки программных комплексов с использованием CASE-средств, контролировать качество разрабатываемых программных продуктов; <i>в части следующих результатов обучения:</i>
1.
10. , , SWEBOOK, ,
2.
4.
5. - ,
6.
7.
1.
6.
Компетенция ФГОС: ПК.6 пониманием существующих подходов к верификации моделей программного обеспечения (ПО); <i>в части следующих результатов обучения:</i>
1.
1.
Компетенция ФГОС: ПК.7 применением перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий; <i>в части следующих результатов обучения:</i>
1. ,

	(
	,	
	,	
)	
.1. 1		
1.находить и адаптировать для решения профессиональных задач теоретические, практические и технологические артефакты из новых областей знаний		
.1. 1		
2.способность осваивать и использовать новейшие достижения области профессиональной деятельности		;
.5. 1		
3.источники информации, необходимой для профессиональной деятельности		
.5. 1		
4.участвовать в коллективной разработке программного обеспечения в различных формах организации и самоорганизации коллектива		;
.6. 1		
5.управлять коллективной разработкой программного обеспечения в различных формах организации и самоорганизации коллектива		;
.6. 1		
6.методы верификации программного обеспечения		;
.6. 1		
7.применять методы верификации программного обеспечения		
.7. 1		
8.программные средства, среды поддержки и разработки для перспективных направлений профессиональной деятельности		
.8. 1		
9.принципы сопровождения и конфигурирования программного обеспечения		;
.19. 1		
10.знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения		;
.19. 10		SWEBOOK,
11.этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOOK, стандартов по процессам жизненного цикла ПО		;
.19. 2		
12.инструменты и методы управления требованиями		;
.19. 4		

2.	UP	0	2	1, 11, 18, 3	UP. : , , - , . : - , , , , , . .
3.		0	2	1, 18, 3	. - (XP). (agile) SCRUM. Agile UP, ICONIX. SCRUM - .. . SCRUM- . - . - .

4. UML	0	2	1, 11, 18, 3	<p>UML,</p> <p>· : - , , , , ; , , - , ; , , - , - · : , , , , · . - , · - , : , , , · , , . (-). · . : - , - : , · .</p>
5.	0	2	1, 11, 12, 18, 3, 6	<p>· - , · , : , , - , , , · , · . : , - - , , , , - , · .</p>

5.	4	4	10, 11, 14, 2, 9	<p> : - : - (,), . ; - (,) - - ; - ; - , , , (); - , , . - . </p>
----	---	---	------------------	--

3.4

	, .			
: 2				
:				

6.	0	2	11, 18, 4, 5, 9	<p>().</p> <p>:</p> <p>(),</p> <p>UP.</p> <p>().</p> <p>UP</p>
:				
9.	0	2	13, 16	
10.	0	2	16, 9	<p>(TDD-Test Driven Development).</p>

11.	0	2	16, 17, 2, 8, 9	() model-view-controller (MVC). : session, worker thread
12.	0	2	16, 7, 9	

4.

: 1				
1		2, 4, 5	15	3
<p>- - (, , -) - (,) - - : / . . . ; . . . [. . .]: - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. - . . . : () []: 080800]/ . . . ; . . . - . . . , [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - . . .</p>				
2		10, 11, 12, 6, 8, 9	5	0
<p>. . . : . . . []: - , [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. - . . .]: () [080800]/ , [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - . . .</p>				
3		10, 11, 12, 6, 8, 9	10	1

<p> http://elibrary.nstu.ru/source?bib_id=vtls000220170. - () [080800] / . ; [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - </p>				
: 2				
1		10, 11, 12, 2, 4, 5, 6, 8, 9	50	8
<p> UP(RUP) (1 . .). UP(RUP) (4) (9). UP(RUP) : . . . [] : ; [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. - </p>				
2		2, 4, 5	10	0
<p> ; [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. - () [080800] / . ; [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - </p>				
3		10, 11, 12, 6, 8, 9	10	0
<p> ; [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. - () [080800] / . ; [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - </p>				
4		11, 13, 16, 17, 18, 2, 4, 5, 7, 8, 9	10	0
<p> 3.4 : [] : / . ; ; [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000220170. () [080800] : [; ; [2014]. - : http://elibrary.nstu.ru/source?bib_id=vtls000208496. - </p>				

5.

(. 5.1).

5.1

	-
	:YandexDisk vk.com/cprog_cs; :Skype;
	:"
	:Skype
	:YandexDisk vk.com/cprog_cs; :Skype;
	:"

5.2

1	
<p>Краткое описание применения: Метод проектов: Лаб. работы выполняются в виде единого проекта "нарастающим итогом", тема которого выбирается в начале курса</p>	

2	
<p>Краткое описание применения: Практические занятия в семестре 1 проводятся в форме тренинга под руководством преподавателя по разработке документов жизненного цикла ПО</p>	

6.

(),

-

15-

ECTS.

. 6.1.

6.1

: 1		
<i>Практические занятия:</i>	0	50
<i>РГЗ:</i>	0	30
<i>Зачет:</i>	0	20
: 2		
<i>Лабораторная:</i>	10	20
<i>Курсовой проект:</i>	0	40
<i>Экзамен №1:</i>	0	40
<i>Экзамен №2:</i>	0	40

			/		
.1	1.		+		
.5	1.		+		
.6	1.		+		
.8	1.		+		
.1	1.		+		
.5	1.		+		
.19	1.				+
	10. , , SWEBOK, ,				+
	2.				+
	4.				+
	5. - ,				+
	6.				+
	7.	+		+	+
	1.	+		+	+
	6.		+		+
.6	1.		+		
	1.				+
.7	1. ,		+		

7.

1. Липаев В. В. Программная инженерия. Методологические основы : [учебник для вузов по направлению "Бизнес-информатика" (080700)] / В. В. Липаев ; Гос. ун-т - высш. шк. экономики. - М., 2006. - 605, [1] с.

2. Буч Г. Язык UML : руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон ; [пер. с англ. Н. Мухина]. - М., 2007. - 493 с. : ил.

1. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо ; [пер. с англ. В. Горбункова]. - СПб. [и др.], 2002. - 492 с. : ил.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Гради Буч ; пер. с англ. под ред. И. Романовского и Ф. Андреева. - М., 1998. - 558 с. : ил. - Тит. л. парал. рус., англ..
3. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем : Учебное пособие для вузов / С. А. Орлов. - СПб., 2003. - 473 с. : ил.
4. Вязовик Н. А. Программирование на Java. Курс лекций / Н. А. Вязовик ; Интернет ун-т информ. технологий. - М., 2003. - 586 с. : ил., схемы

1. Романов Е. Л. Архитектура и прикладные протоколы клиент-серверных приложений [Электронный ресурс] : электрон. учеб.-метод. комплекс / Е. Л. Романов; Новосиб. гос. техн. ун-т. – Новосибирск, 2015. – Режим доступа: <http://dispace.edu.nstu.ru/didesk/course/show/5379>. - Загл. с экрана.

2. ЭБС НГТУ : <http://elibrary.nstu.ru/>

3. Архипенков С. Лекции по управлению программными проектами [Электронный ресурс] / С. Архипенков. – Москва, 2009. — 128 с. – Режим доступа : http://www.arkhipenkov.ru/resources/sw_project_management.pdf. – Загл. с экрана.

4. ЭБС «Издательство Лань» : <https://e.lanbook.com/>

5. ЭБС IPRbooks : <http://www.iprbookshop.ru/>

6. ЭБС "Znanium.com" : <http://znanium.com/>

7. :

8.

8.1

1. Романов Е. Л. Программная инженерия [Электронный ресурс] : электронный учебно-методический комплекс / Е. Л. Романов ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000220170. - Загл. с экрана.
2. Пустовалова Н. В. Программная инженерия (метрическая теория программ) [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 080800 Прикладная информатика] / Н. В. Пустовалова, Г. И. Кайгородцев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000208496. - Загл. с экрана.

8.2

1 NetBeans IDE

2 СОСОМО® П.2000.4

3 StarUML

9. -

1	(- , ,)	

1	(Internet)	Internet

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ____ ” _____ 2017 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Технология разработки программного обеспечения

Образовательная программа: 09.04.01 Информатика и вычислительная техника, магистерская
программа: Прикладные информационные системы и технологии

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Технология разработки программного обеспечения приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОК.1 способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень	у1. способность осваивать и использовать новейшие достижения области профессиональной деятельности	Разработка программной архитектуры Шаблоны проектирования и конструирования	Курсовой проект, разделы 2.3,3	
ОК.5 использованием на практике умений и навыков в организации исследовательских и проектных работ, в управлении коллективом	у1. участвовать в коллективной разработке программного обеспечения в различных формах организации и самоорганизации коллектива	Оценка трудоемкости и стоимости программного проекта Разработка бизнес-плана программного проекта Разработка прототипа графического интерфейса Управление программными проектами. Общие положения	Курсовой проект, разделы 4.3	
ОК.6 способность проявлять инициативу, в том числе в ситуациях риска, брать на себя всю полноту ответственности	у1. управлять коллективной разработкой программного обеспечения в различных формах организации и самоорганизации коллектива	Оценка трудоемкости и стоимости программного проекта Разработка бизнес-плана программного проекта Разработка прототипа графического интерфейса Управление программными проектами. Общие положения	Курсовой проект, разделы 1.4,1.5,2.2	
ОК.8 способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы	з1. принципы сопровождения и конфигурирования программного обеспечения	Разработка программной архитектуры Тестирование ПО Технология программирования. Разработка, управляемая тестированием Управление программными проектами. Общие положения Шаблоны проектирования и конструирования	Курсовой проект, разделы 2.3,4.1	
ОПК.1 способность воспринимать математические, естественнонаучные, социально-экономические и профессиональные знания, умением самостоятельно приобретать, развивать и применять их для решения нестандартных задач, в том числе в новой или	у1. находить и адаптировать для решения профессиональных задач теоретические, практические и технологические артефакты из новых областей знаний	UML как средство поддержки процесса проектирования Архитектурное проектирование Детальное проектирование Жизненный цикл ПО Унифицированный процесс разработки UP Функциональное проектирование Экстремальные и гибкие методологии разработки ПО	Курсовой проект, разделы 1.1,1.5,2.3,3.1	

незнакомой среде и в междисциплинарном контексте				
ОПК.5 владение методами и средствами получения, хранения, переработки и трансляции информации посредством современных компьютерных технологий, в том числе в глобальных компьютерных сетях	31. источники информации, необходимой для профессиональной деятельности	UML как средство поддержки процесса проектирования Архитектурное проектирование Детальное проектирование Жизненный цикл ПО Унифицированный процесс разработки UP Функциональное проектирование Экстремальные и гибкие методологии разработки ПО	Курсовой проект, разделы 1.1,1.5,2.3,3.1	
ПК.19/ПТ способность к применению современных технологий разработки программных комплексов с использованием CASE-средств, контролировать качество разрабатываемых программных продуктов	31. знать принципы построения архитектуры программного обеспечения и виды архитектур программного обеспечения	Разработка программной архитектуры		Экзамен, вопросы 6-8
ПК.19/ПТ	32. инструменты и методы управления требованиями	Анализ бизнес-процессов и разработка видения проекта Моделирование предметной области Разработка требований Функциональное проектирование		Экзамен, вопросы 4-5
ПК.19/ПТ	34. типовые метрики программного обеспечения	Оценка трудоемкости и сроков Оценка трудоемкости и стоимости программного проекта Технология программирования. Оценка кода		Экзамен, вопросы 17,18
ПК.19/ПТ	35. компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними	Архитектурное проектирование Детальное проектирование Разработка программной архитектуры		Экзамен, вопросы 6-8
ПК.19/ПТ	36. основные методы измерения и оценки характеристик программного обеспечения	Оценка трудоемкости и сроков		Экзамен, вопросы 18
ПК.19/ПТ	37. технологии программирования	Тестирование ПО Технология программирования. Оценка кода Технология программирования.	РГЗ	Зачет Экзамен, вопросы 23-30

		Разработка, управляемая тестированием Шаблоны проектирования и конструирования		
ПК.19/ПТ	з10. этапы, технологические процессы, артефакты унифицированного процесса разработки ПО, содержание свода знаний о программной инженерии SWEBOOK, стандартов по процессам жизненного цикла ПО	UML как средство поддержки процесса проектирования Унифицированный процесс разработки UP Управление программными проектами. Общие положения Функциональное проектирование		Экзамен, вопросы 10-14
ПК.19/ПТ	у1. использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Шаблоны проектирования и конструирования	РГЗ	Зачет Экзамен, вопросы 27-30
ПК.19/ПТ	уб. применять методологии разработки программного обеспечения	UML как средство поддержки процесса проектирования Анализ бизнес-процессов и разработка видения проекта Архитектурное проектирование Детальное проектирование Жизненный цикл ПО Моделирование предметной области Оценка трудоемкости и сроков Разработка бизнес-плана программного проекта Разработка модели прецедентов Разработка требований Унифицированный процесс разработки UP Управление программными проектами. Общие положения Управление рисками Функциональное проектирование Экстремальные и гибкие методологии разработки ПО	Курсовой проект, разделы 1.4-1.6,2,4	
ПК.6/НИ пониманием существующих подходов к верификации моделей программного обеспечения (ПО)	з1. методы верификации программного обеспечения	Разработка требований Функциональное проектирование	Курсовой проект, разделы 2.1	
ПК.6/НИ	у1. применять методы верификации программного обеспечения	Тестирование ПО		Экзамен, вопросы 21-23

ПК.7/НИ применением перспективных методов исследования и решения профессиональных задач на основе знания мировых тенденций развития вычислительной техники и информационных технологий	31. программные средства, среды поддержки и разработки для перспективных направлений профессиональной деятельности	Шаблоны проектирования и конструирования	Курсовой проект, разделы 3.1	
---	---	---	---------------------------------	--

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 1 семестре - в форме зачета в 2 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОК.1, ОК.5, ОК.6, ОК.8, ОПК.1, ОПК.5, ПК.19/ПТ, ПК.6/НИ, ПК.7/НИ.

Зачет проводится в устной по билетам

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 1 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

В 2 семестре обязательным этапом текущей аттестации является курсовой проект. Требования к выполнению курсового проекта, состав и правила оценки сформулированы в паспорте курсового проекта.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОК.1, ОК.5, ОК.6, ОК.8, ОПК.1, ОПК.5, ПК.19/ПТ, ПК.6/НИ, ПК.7/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание

курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт зачета

по дисциплине «Технология разработки программного обеспечения», 1 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: выбираются два вопроса из общего списка шаблонов проектирования. Ответ готовится в виде конспекта и должен содержать:

- Диаграмму классов шаблона
- Описание основных классов: данные, методы
- Общее описание поведения и алгоритмов шаблона
- Пример использования шаблона в разработке

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к зачету по дисциплине «Технология разработки программного обеспечения»

1. Шаблон проектирования Flyweight
2. Шаблон проектирования Double Buffering

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) _____ (дата)

2. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за зачет составляет 20 баллов. Соответственно, критерий оценки определяется в процентах к этому баллу:

- Ответ на билет для зачета считается **неудовлетворительным**, если отсутствует один из пунктов содержания, студент не может прокомментировать содержания, не дает определений основных понятий, оценка составляет менее 50% базовой

- Ответ засчитывается на **пороговом** уровне, если все пункты содержания присутствуют, студент минимально комментирует их содержание и не способен его развивать - оценка составляет не более 50% базовой
- Ответ засчитывается на **базовом** уровне, если все пункты содержания присутствуют в достаточном объеме, студент комментирует их содержание и способен его развивать - оценка составляет 50-80% базовой
- Ответ засчитывается на **продвинутом** уровне, если он соответствует базовому уровню, а также студент предлагает оригинальные решения поставленной задачи и отвечает на дополнительные вопросы по поводу его реализации - оценка составляет 80-100% базовой

3. Шкала оценки

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины

4. Вопросы к зачету по дисциплине «Технология разработки программного обеспечения»

Вариант шаблона проектирования

1. Builder
2. Factory
3. Prototype
4. Singleton
5. Adapter
6. Bridge
7. Composite
8. Decorator
9. Facade
10. Flyweight
11. Proxy
12. Command
13. Iterator
14. Mediator
15. Snapshot
16. Observer
17. State
18. Strategy
19. method template
20. visitor.
21. model-view-controller (MVC)
22. session
23. transaction
24. Single Threaded Execution
25. Two-phase Termination
26. Asynchronous Task
27. Lock Object, Read/Write Lock
28. Scheduler
29. Double Buffering
30. Producer-consumer

Паспорт расчетно-графического задания (работы)

по дисциплине «Технология разработки программного обеспечения», 1 семестр

1. Методика оценки

Расчетно-графическое задание выполняется в виде разработки одного из шаблонов проектирования. Пояснительная записка должна содержать:

- Учебно-методический материал по шаблону (учебники, учебные пособия, лекционный материал – краткое описание)
 - Материал по практике применения шаблона (форумы, тематические сайты)
 - Описание разработки
 - Результаты тестирования
- Все сторонние материалы должны быть снабжены корректно оформленными библиографическими ссылками.

При защите расчетно-графического задания (работы) оцениваются следующие виды работ:

- Постановка и анализ задачи;
- Разработка структур данных для представления предметной области и алгоритмов;
- Объем и качество программного кода;
- Тестирование программного кода;
- Оформление текста программы и результатов работы

2. Критерии оценки

Согласно положению о балльно-рейтинговой системе НГТУ, базовый балл рейтинга за РГР определен в рабочей программе (16 баллов). Соответственно, критерий оценки определяется в процентах к этому баллу:

- РГР считается **невыполненной**, если отсутствуют необходимые разделы описания программы, либо программа не работает, либо студент не в состоянии объяснить принципов ее работы (алгоритма) – оценка составляет <25% базовой;
- РГР засчитывается на **пороговом** уровне, разработка не полностью соответствует заданию, имеются ошибки при тестировании, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50% базовой
- РГР засчитывается на **базовом** уровне, если функционал разработки соответствует заданию, отсутствуют ошибки тестирования, пояснительная записка оформлена в целом грамотно - оценка составляет 50-80% базовой
- РГР засчитывается на **продвинутом** уровне, если функционал реализован полностью и эффективно, в пояснительной записке отражены все аспекты

структурного, функционального проектирования и тестирования - оценка составляет 80-100% базовой

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины (16 баллов).

4. Примерный перечень тем РГЗ(Р)

1. **Шаблон MVC. Сетевая (локальная) игра типа «Морской бой», «Домино».** Модель хранит структуру данных игры - расстановку кораблей, и над ней выполняются методы по управлению игрой. Контроллер определяет порядок ходов, проверяет возможность выполнения хода. Внешнее представление (view) связано с контроллером двунаправленным интерфейсом: контроллер управляет отображением элементов игрового поля, выводит текстовые сообщения, получает от представления события – клик по элементу поля, начало новой игры, завершение и т.п.. Варианты игры:
 - локальная: одна модель, один контроллер, два представления для игроков;
 - сетевая: одна модель, один контроллер, два представления для игроков. Представления связываются с приложением, содержащим контроллер и модель. Контроллер обеспечивает множественные соединения и синхронизацию;
 - сетевая: две модели, два контроллера, два представления. Контроллеры поддерживают соединение и передают команды: сделан ход, синхронизация моделей, сброс и начальная установка игры.
2. **Шаблон MVC (классический).** Представление имеет ряд текстовых полей, отображающих параметры модели физического объекта. На каждый параметр представление отдельно подписывается к модели на событие, связанное с его изменением. Интерфейс подписки идентичен для всех параметров. Для каждого параметра создается оригинальный контроллер, обрабатывающий команду изменения значения параметра со стороны представления. Модель описывается набором зависимостей, параметры которой могут быть *входными, выходными (результатами) и выходными с возможностью задания начальных значений (инициализацией)*.
3. **Шаблон – прототип. Класс таблицы с произвольной структурой столбцов.** Хранимые данные разных типов - целые, вещественные, дата, время, GPS-координаты создаются на основе абстракции данных с функционалом: имя класса, парсинг из строки и вывод в строку, клонирование, сравнение и сложение с объектом того же типа. Строка таблицы определяется набором объектов-прототипов для столбцов. Сама строка также клонируется. Таблица состоит из вектора строк-имен, строки-прототипа и строк самой таблицы. Функционал: создание таблицы, добавление столбцов, добавление строк, сортировка по столбцу с заданным номером, сложение строк, сохранение и

- загрузка из файла. Программный и интерфейс и оконное приложение. Тестирование на больших данных - импорт из Excel, генерация тестовых таблиц.
4. *Шаблон – приспособленец. Работа с деревом версий текстового файла.* Файл редактируется по словам. Класс словаря содержит хэш-таблицу адаптеров со ссылками на оригинальные слова в виде *ключ – само слово*. Адаптер содержит количество ссылок на слово из всех версий текста. Каждая версия текста – вектор ссылок на адаптеры. При добавлении или вставке слова в версию текста оно ищется в фабрике, при нахождении счетчик в адаптере увеличивается на 1. Иначе добавляется в словарь с новым адаптером. При удалении счетчик ссылок уменьшается. Изменение слова рассматривается как последовательность операций удаления и вставки. Вся структура данных сериализуется в файл. Протестировать шаблон на сказке *Репка*.
 5. *Шаблон – композиция для древовидной системы. Графический редактор с группировкой/разгруппировкой элементов, изменением размеров, переносом на передний/задний план, перемещением объектов, сохранением картинки в файл.* Абстрактный класс элемента, группы элементов и конкретных графических объектов - окружность, полигон, строка текста. Ограничивающий прямоугольник, селекция объектов по точке и прямоугольнику.
 6. *Шаблон Command.* Группа команд обработки объекта с общим интерфейсом *Do/ReDo/Undo*. Производный класс запоминает параметры, необходимые для выполнения прямой и обратной команды. Класс – менеджер команд поддерживает очередь команд ограниченной длины, текущий обрабатываемый объект, методы *Do/ReDo/Undo*, выбирая их из очереди и вызывая соответствующие методы в объектах-командах. **Графический редактор** набором команд редактирования графических объектов - создать объект, переместить, изменить размер, удалить, переместить на передний и задний план.
 7. *Шаблон – пул потоков.* Класс потока представляет собой поток, запускаемый при создании объекта. Содержит ссылку на исполняемый код через *Runnable*, код завершения и ссылку на менеджер пула. Код потока содержит цикл, в котором засыпает, после пробуждения исполняемый назначенный код, уведомляет менеджер о своем завершении и засыпает. Менеджер потоков содержит вектор потоков-исполнителей. При обращении к менеджеру с запросом, содержащим код исполнения и код завершения, выбирается поток из пула, заполняется данными и пробуждается. Если свободного потока нет, то запрос ставится в очередь. Провести сравнительное тестирование для потока запросов с пулом и при запуске потоков обычным образом.
 8. *Шаблон – прокси (фильтр).* Класс с интерфейсом текстового потока при конструировании делегируется к однотипному объекту источнику и **отфильтровывает набор слов**, передаваемый при конструировании, используется внутренняя очередь символов для отложенного распознавания. Протестировать на цепочке фильтров для разных наборов слов.
 9. *Шаблон – сессия* для соединения на сокетах в виде библиотеки для клиента/сервера. Клиент и сервер используют синхронный обмен *запрос-ответ*. При первоначальном установлении соединения клиент получает уникальный идентификатор, сервер создает дескриптор соединения. Клиент нумерует передаваемые сообщения, сервер сохраняет номер и ответ на последнее

переданное сообщение. Сервер периодически или после каждого изменения сохраняет дескрипторы в файл. Клиент также запоминает в файле идентификатор сессии, номер и последнее переданное сообщение. Обеспечить сохранность последовательности (отсутствие пропадания и дублирования) сообщений при перезагрузке клиента и сервера. Реализовать модель банкомата и платежной системы.

10. *Модель параллельных запросов к серверу от группы потоков.* Множество потоков может посылать независимые запросы к серверу через единственное соединение. Запрос ставится в очередь, нумеруется, клиентский процесс засыпает. Поток передачи от клиента выбирает сообщения из очереди и передает в соединение. Поток приема на сервере, приняв очередное сообщение, запускает поток исполнения запроса, по завершении которого в очередь ответов ставится ответ, в котором сохраняется порядковый номер запроса. Поток передачи ответов на сервере выбирает сообщения из очереди и передает в соединение. Поток приема ответов определяет по номеру в сообщении поток, передавший запрос и пробуждает его. Ответное сообщение находится в запросе, переданном потоком, и выводится им. Промоделировать *группу потоков, посылающих случайные слова, которые сервер переворачивает со случайной задержкой.*
11. *Шаблон – кэш объектов.* Разработать класс - кэш объектов с возможностью изменения размера кэша, сбора статистики и применения различных стратегий вытеснения (FIFO, LRU, RAND). Использовать для кэширования слов текстового файла при последовательном чтении. Кэш – хеш-таблица с ключом-словом и необходимыми параметрами для моделирования, например, номер последнего обращения для LRU. При чтении очередного слова проверяется его наличие в кэше. При отсутствии производится замещение. Для выбранного файла строится зависимость доли попадания в кэш как функция от размера кэша и способа вытеснения. Сравнить результаты для файлов с разным содержимым - художественное произведение, технический текст.
12. *Шаблон – итератор.* Операции над итератором: установка на первый, последний, следующий, предыдущий, по логическому номеру, извлечение через итератор, вставка на позицию итератора, замещение, удаление. Структура данных: двухуровневый массив ссылок (двумерный массив), двоичное дерево, дерево с данными в конечных вершинах, список с массивом ссылок [9]. Использование нескольких итераторов. *Замечание по теме:* для корректной реализации операции удаления в основном классе использовать вектор созданных итераторов. При удалении одним из них элемента остальные, которые на него ссылаются, генерируют исключение при очередном обращении. Рассмотреть другие варианты корректного разделения.

Разработка классов, использующих внутренние потоки для промежуточной буферизации данных. Для моделирования прикладных процессов использовать потоки, которые засыпают на случайный момент времени, после чего читают/записывают очередную порцию данных постоянного или случайного размера. Предусмотреть сбор статистики в классах буферизации – средний объем данных в буфере.

13. Класс буферизованного ввода в реальном времени. При конструировании получает параметр – физический поток данных с интерфейсом *InputStream*.

Использует внутренний циклический буфер или односвязный список блоков, содержащих массив байтов фиксированной размерности (буферный пул). Создает поток, который читает байты из входного потока и записывает в циклический буфер. При заполнении циклического буфера засыпает. Метод чтения извлекает из циклического буфера очередной байт, возвращает -1 при окончании данных в потоке-источнике, блокируется при отсутствии данных в циклическом буфере.

14. Класс – *PipedStream* с циклическим буфером данных. Класс *PipedOutputStream* имеет циклический буфер, в который пишет поток байтов, блокируя текущий поток при заполнении буфера. Класс *PipeInputStream* получает ссылку на *PipedOutputStream* и при чтении данных либо блокируется при их отсутствии, либо извлекает данные, деблокируя поток записи.
15. Класс отложенной записи. При открытии файла классом с присоединенным интерфейсом *OutputStream* создается *ByteOutputStream*, в который пишутся данные потока. При закрытии объект, содержащий имя файл и байтный массив ставится в очередь, из которой фоновый поток их извлекает и пишет файлы. Сравнить среднее время записи в обычный и *отложенный файл*.

2.

Паспорт экзамена

по дисциплине «Технология разработки программного обеспечения», 2 семестр

1. Методика оценки

Экзамен проводится в устной форме по билетам. Билет формируется по следующему правилу: первый (теоретический) вопрос выбирается из диапазона, второй вопрос – практическая задача на проектирование. Выбирается вариант предметной области и одна из компонент проектирования. В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы по другим компонентам проектирования выбранной предметной области.

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Технология разработки программного обеспечения»

Вопрос 1. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.

Вопрос 2. Система мониторинга междугородных автобусных перевозок. Карта автодорог, маршрут, график движения по маршруту, планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций. Модель классов предметной области

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись) _____ (дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент не ответил на теоретический или практический вопрос, оценка составляет *<30% базовой*.
- Ответ на экзаменационный билет (тест) засчитывается на **пороговом** уровне, если студент при ответе на теоретический вопрос дает определение основных понятий, может показать причинно-следственные связи явлений, при решении практической задачи допускает **непринципиальные** ошибки, например, структурные, оценка составляет *30-60% от базовой*.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при

дает развернутый ответ на теоретический вопрос с несущественными ошибками и решает практическую задачу с несущественными ошибками, оценка составляет *60-80% от базовой*.

- Ответ на экзаменационный билет (тест) билет засчитывается на **продвинутом** уровне, если студент дает развернутый ответ на теоретический вопрос и дополнительные вопросы, решает практическую задачу и вносит изменения в решение по дополнительным требованиям, оценка составляет *80-100% от базовой*.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины, т.е. 40 баллов за экзамен.

4. Вопросы к экзамену по дисциплине «Технология разработки программного обеспечения»

Теоретический вопрос

1. Сущность программной инженерии (ПИ). Связь с computer science. Особенности в сравнении и другими инженерными дисциплинами. Свод знаний и ПИ SWEBOK
2. Жизненный цикл (ЖЦ) программного продукта и проекта. «Легкие» и «тяжелые» модели процессов разработки ПО. Этапы и технологические процессы (дисциплины) ЖЦ. Результаты этапов и основные документы. Каскадная, итеративная и спиральная модели.
3. Унифицированный процесс UP. Фазы жизненного цикла: исследование, анализ, реализация, внедрение. Содержание и результаты фаз. Итерация и ее рабочие потоки: требования, анализ, проектирование, реализация, тестирование, их содержание.
4. Фаза исследования. Основные дисциплины и артефакты. Дисциплина «анализ предметной области», бизнес-анализ. Диаграммы потоков данных, деятельности. Моделирование предметной области.
5. Фаза анализа и проектирования. Дисциплина «анализ требований». Способы извлечения и фильтрации требований. Бизнес-требования, бизнес-требования, системные требования, функциональные требования. Разработка и управление требованиями. Документ «спецификация требований к ПО ». Диаграммы прецедентов.
6. Фаза анализа и проектирования. Понятие архитектуры, ее многомерность. Основные методы проектирования и их особенности: структурное, функциональное, объектно-ориентированное, компонентное, проектирование на основе структур данных. Классы анализа. Виды классов: граница, управление, сущность. Диаграммы устойчивости. Архитектурные аспекты технологического процесса проектирования (по SWEBOK)
7. Фаза анализа и проектирования. Дисциплина проектирование (design). Ключевые моменты проектирования по SWEBOK: параллелизм, контроль и обработка событий, распределение компонентов, обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости, взаимодействие и представление (MVC), сохраняемость данных (доступность «долгоживущих» данных).

8. Фаза анализа и проектирования. Многоуровневая архитектура клиент-серверных приложений. Тонкие и толстые клиенты. Локальное и сетевое взаимодействие слоев через интерфейсы и протоколы. Совместное использование кода различными типами клиентов.
9. Фаза анализа и проектирования Проектирование графического интерфейса (GUI). Основные аспекты. Архитектурное проектирование, основанное на GUI. Факторы, характеризующие GUI: производительность, человеческие ошибки, обучение, субъективное восприятие, запоминание, поиск, визуализация, навигация
10. Виды моделей. Сущность UML как средства моделирования. Структура UML, статическая и динамическая составляющие модели. Составные элементы: сущности, отношения, диаграммы. Виды сущностей: структурные сущности – класс, интерфейс, кооперация, прецедент, активный класс, компонент, узел; поведенческие сущности – взаимодействия, деятельности, автоматы; группирующая сущность – пакет, аннотационная сущность – примечание.
11. UML. Виды отношений: зависимость, ассоциация, агрегация, композиция, включение, обобщение, реализация. Отношения. Связи – отношения между объектами. Направленность связи, Сообщения. Диаграммы объектов. Ассоциации – отношения между классами. Свойства ассоциации: имя, кратность, навигация, атрибуты. Рефлексивные ассоциации, деревья и сети. Классы атрибутов ассоциаций (классы-ассоциации). Зависимости. Зависимости использования «use», «call», «parameter», «send» и «instantiate». Зависимости абстракции. Зависимости доступа.
12. UML. Принятые деления: классификатор-экземпляр, интерфейс-реализация. Расширения: ограничения, стереотипы. Классификация диаграмм. Диаграммы классов (объектов). Диаграммы взаимодействий, коммуникационные диаграммы.
13. UML. Диаграммы деятельности. Технология сетей Петри. Параллелизм. Поток управления, узел действия, ребро, узел управления, объектный узел, буферизация и в объектном узле. Объектные узлы – параметры, состояния объектных узлов. Контакты. Прерывающие ребра. Контакты исключений. Потоки объектов. Их аналоги в программировании
14. UML. Диаграммы состояний. Конечные автоматы.
15. Экстремальное и гибкое программирование. Манифест экстремального программирования (XP). Гибкие (agile) технологии. SCRUM. Agile UP, ICONIX.
16. SCRUM как технологический фреймворк.. Терминология. Спринт. Митинг. Собственник проекта. Команда. SCRUM-мастер. Беклог проекта и спринта. Планирование спринта. Диаграмма сгорания. Оценка трудоемкости. Покер-планирование.
17. Оценка программного кода. Метрики кода. Метрики количественные, сложности потока управления и потока данных, метрики ООП, прагматические метрики. Средства оценки качества программного кода.
18. Метрика проекта. Оценка сроков на основании трудоемкости (по Боэму). Оценка на основе собственного опыта. Метод PERT. Оценка на основе функциональных точек. Оценка по отраслевым данным. Метод COSOMO II

19. Управление проектами как инженерная дисциплина. Особенности управления программными проектами. Роль и место УПП в программной инженерии (ПИ). Компоненты организационного (менеджмент) и технологического (исполнение) планирования в УПП.
20. Определение и характеристики риска. Шкалы оценивания последствий и вероятности. Способы идентификации. Реакция на риски. Наиболее вероятные риски по Бозму и Архипенкову. Качественные оценки рисков. Количественные оценки: анализ чувствительности, дерево решений, имитационное моделирование. Управление, направленное на снижение рисков. Вероятностный характер оценивания, его природа. Последствия «агрессивного» планирования. Исходные данные для оценивания, характеристики проекта, используемые в оценивании. Оценка сроков на основании трудоемкости (по Бозму).
21. Понятие программной ошибки. Философия ошибок. Классификационные характеристики ошибок. Отладка, инспекция и тестирование как этапы поиска ошибок. Ошибки вычислений и преобразований. Ошибки структурирования кода. Ошибки форматов входных данных. Ошибки форматов внутренних данных и соглашений по данным. Ошибки сборки, конфигурирования, размещения. Ошибки использования ресурсов. Ошибки, связанные с ограничениями по ресурсам. Ошибки реактивности и производительности. Ошибки параллелизма и синхронизации. Ошибки распределенных систем и протоколов. Ошибки пользовательского интерфейса.
22. Структурное тестирование. Тестирование операторов, условий (решений) и путей. Комбинационное тестирование. Пример. Функциональное тестирование. Классы эквивалентности по входным данным. Проектирование тестового покрытия. Классы эквивалентности по граничным условиям. Примеры.
23. Отладка и инспекция программного кода. Сущность отладки. Приемы отладки. Инспекция как неформальный анализ программного кода. Логический и временной анализ. Методы инспектирования. TDD – разработка, управляемая тестированием.
24. Шаблоны проектирования. Производящие шаблоны builder, factory, prototype, singleton. Шаблоны проектирования. Структурные шаблоны adapter, bridge, composite, decorator, facade, flyweight, proxy
25. Шаблоны проектирования. Поведенческие шаблоны command, iterator, mediator, snapshot, observer, state, strategy, method template, visitor.
26. Шаблоны проектирования. Системные шаблоны model-view-controller (MVC), session, transaction
27. Шаблоны проектирования. Шаблоны параллелизма Single Threaded Execution, Two-phase Termination, Asynchronous Task, Lock Object, Read/Write Lock, Scheduler, Double Buffering, Producer-consumer

Практический вопрос

Для произвольно выбранного варианта лабораторных работ выполнить черновую разработку одной из компонент:

1. Модель классов предметной области

2. Общая архитектура системы
3. Диаграмма состояний одной из сущностей
4. Диаграмма деятельности, потоков данных для бизнес-процесса
5. Перечень требований (функциональные, бизнес-требования, системные требования, бизнес-правила, атрибуты качества) в виде дерева требований

Варианты заданий к практическому вопросу – предметная область проектирования.

1. Система продажи билетов в кинотеатре. Клиент, кассир, смена, билетер, администратор, планирование сеансов, продажа билетов кассиром, бронирование и продажа через Интернет, финансовые отчеты, план зала.
1. Система продажи театральных билетов. Приложение кассира - множество точек продажи, приложение распространителя, бронирование через Интернет, связь с платежными системами, план зала (мета-уровень описания), спектакли, репертуар.
2. Система автоматизации диспетчерской службы такси. Диспетчер, водитель, клиент, директор, прием заказов, ведение очередей, ручное распределение заказов, приложение водителя, мониторинг прохождения заказа.
3. Система автоматизированного заказа такси через Интернет. Серверное приложение для автоматического распределения заказов с учетом нагрузки, web-приложение и мобильное приложение для клиентов, приложение водителя, приложение администратора для форсмажорных и конфликтных ситуаций. Автоматическое распределение заказов на основе расстояния для клиента и других критериев, предложение свободных заказов водителю, голосование за заказ, мониторинг прохождения заказа. Виды адресов с привязкой к GPS-координатам: почтовый, место, корпоративный (фирма, организация).
4. Система ведения корпоративной адресной базы для мобильных клиентов. Типы адресов: служебный - подразделение, корпус, кабинет, домашний - почтовый. Типы контактов: электронная почта, телефон, социальная сеть, адрес. Административная структура организации. Хранение списка контактов, обмен контактами, иерархическая многомерная адресная книга с каталогами (тегами), общая и личная адресные книги.
5. Мессенджер мобильных клиентов аналогичный WhatsUp. Регистрация по номеру мобильного телефона. Передача сообщений, файлов, синхронизация адресных книг, иерархическая многомерная адресная книга с каталогами (тегами), поиск по общей адресной книге, личная адресная книга с собственной системой каталогов (тегов)
6. Мессенджер с прямой связью клиентов. Сервер контактов, регистрация, сохранение IP-адресов клиента и статистики пребывания, авторизация с сообщением текущего IP-адреса, дозвон по списку IP-адресов, прямая связь с передачей сообщений и файлов, локальные адресные книги, обмен адресами.
7. Система заказных грузоперевозок по городу. Клиент, диспетчер, магазин, водитель. Прием и оформление заказов, отчеты и сопроводительные документы, распределение заказов диспетчером. Два вида заказов: точка-точка и развоз товаров со склада по клиентам. Приложение водителя: просмотр заказов, мониторинг проведения заказа, планирование последовательности

исполнения для развоза, времени доставки. Приложение диспетчера: прием и оформление заказа, распределение, планирование доставки. Параметры заказа – вес и габариты грузов. Транспортные средства и водители. Оплата доставки авансом и при выполнении заказа.

8. Система продажи билетов на междугородные автобусы. Планирование рейсов, расписание, чартерные рейсы, типы автобусов, планы рассадки, водители, кассиры, смены, визуализация рассадки, приобретение билетов в кассе и в кассовых терминалах, бронирование через Интернет, сводные отчеты по маршруту и дате.
9. Система мониторинга междугородных автобусных перевозок. Карта автодорог, маршрут, график движения по маршруту, планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций.
10. Система планирования междугородных транспортных перевозок. Транспортные средства - тип, тоннаж, вместимость. Населенные пункты, дорожная сеть, прием заявок, поиск подходящего транспорта с учетом его местонахождения и порожнего прогона, составление маршрута, оформление и проводка заявок, отчеты по периоду времени, транспортному средству, водителю.
11. Система мониторинга междугородных транспортных перевозок. Населенные пункты, дорожная сеть, маршрут, планирование движения по маршруту, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций, отслеживание заправки, отчеты по расходу топлива и рабочему времени водителей.
12. Справочная система наличия товаров. Многоуровневая система категорий и марок товара. Мета-система классификационных признаков и их значений, например, вес, цвет, производитель, объем памяти, наличие GPS и т.п.. Торговые сети, торговые точки с привязкой к GPS-координатам. Ассортимент в торговой точке, количество товара. Приложение пользователя: поиск по местоположению, по условиям, сформированным для признаков. Приложение торговой точки – редактирование ассортимента. Приложение администратора – редактирование категорий и марок, классификационных признаков.
13. Логистическая система интернет-магазина с пунктами выдачи и доставкой по городу. Прием заказов, оформление заявок поставщикам, уведомление клиентов, отслеживание работы курьеров, отчеты по пунктам выдачи. Многоуровневая система категорий и марок товара. Мета-система классификационных признаков и их значений, например, вес, цвет, производитель, объем памяти, наличие GPS и т.п.. Наличие товара на складе. Отслеживание балансов по каждому виду товара: заказано, наличие на складе, в заявках к поставщикам. Принятие товара на складе, формирование комплектов заказов для пунктов выдачи и курьеров.
14. Система электронного документооборота учебного процесса в ВУЗе. Структура учебного заведения, факультеты, кафедры, группы, студенты, преподаватели. Сторонние организации. Приказы о зачислении/отчислении, назначении тем выпускных и курсовых работ, распределение на практику, распоряжения по подразделениям. Прохождение приказа: создание, визирование, утверждение, нумерация, рассылка.

15. Система книговыдачи школьной библиотеки с использованием QR-кодов. Систематический и алфавитный каталог книг с учетом экземпляров, рекомендованные учебники по предметам, структура учебного процесса: класс, ученик, предметы, преподаватели, сторонние лица. Формирование комплектов учебников, выдача литературы на абонемент. QR-коды читательских билетов, экземпляров книг, выдача и прием, ведение формуляра с историей, уведомление о просрочках.
16. Система книговыдачи библиотеки с использованием QR-кодов. Алфавитный каталог книг с учетом экземпляров, многомерный иерархический тематический каталог или система тегов. Выдача литературы на абонемент. QR-коды читательских билетов, экземпляров книг, ведение формуляра с историей, уведомление о просрочках. Ведение очередей на дефицитные книги - артефакты, уведомления об очередности.
17. Система управления кафе/баром. План зала, закрепление официантов за столиками, планирование смен. Меню – категории, позиции, описание, привязка к кухне или бару. Проведение заказа: закрепление столика за официантом, выбор по меню, частичный заказ, заявки в бар и на кухню, повторение заказа, итоговый расчет. Приложение посетителя, официанта, бармена – планшет, администратора – desktop.
18. Система бронирования мест для клубных мероприятий. План концертов. Анонсы. Стоимость столиков. План зала - мета-уровень описания, настройка под конкретный клуб. Билеты – столики, танцпол. Электронная предоплата. Бронирование. Приложение кассира. Мобильное или web-приложение клиента.
19. Система бронирования мест в гостинице. Мета-уровень описания конкретной гостиницы – расположение и типы номеров, поэтажные планы, список услуг, фото общие и отдельных номеров, расценки. Бронирование через интернет, визуализация свободных/занятых, расчет стоимости, квитанции, отчеты по периодам. Бронирование индивидуальное и групповое. Балансы по занятым, свободным и забронированным номерам по датам и категориям. Заселение, продление проживания, дополнительные услуги, частичный и итоговый расчет.
20. Система планирования взаимосвязанных работ. Учет сотрудников и их занятости запланированными работами. Определение работ в виде цепочек заданий с параллельными ветвями. Распределение заданий по работникам с учетом их занятости. Сдвиг сроков зависимых работ при задержке выполнения одной из них. Приложения сотрудника, руководителя подразделения, генерация отчетов.
21. Система мониторинга обслуживания по заявкам. Система с предварительным сбором заявок и обслуживанием. Категории и виды работ, исполнители, возможность выполнения ими работ по категориям и видам (квалификация). Прием заявок, планирование исполнения, распределение по исполнителям. Оперативное планирование времени исполнения заявок, отслеживание времени исполнения, коррекция времени при задержках с уведомлением клиентов, отказы. Мобильный клиент сотрудника, приложение диспетчера.
22. Система поддержки технологии SCRUM для удаленной работы. Поддержка основных элементов технологии SCRUM - исполнители (команда), лидер, собственник проекта, истории пользователей (задачи), спринты, беклоги,

мониторинг исполнения задач, диаграммы сгорания. Митинги, покер-планирование в режиме чата.

23. Система многомерной организации документов. В качестве документов могут выступать короткие заметки (записи в БД), файлы документов, изображений, звуковых. Поддерживается дерево редактирования версий файла. Создается произвольное количество систем классификаций (каталогов ссылок на файлы) по набору ключевых параметров, например, изображения могут классифицироваться по размеру, цветовой палитре, содержанию, наличию определенных предметов, звуковые файлы по исполнителю, стилю, наличию музыкальных инструментов.

5.

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт курсового проекта

по дисциплине «Технология разработки программного обеспечения», 2 семестр

1. Методика оценки

Курсовой проект выполняется в форме разработки пакета документов и прототипа программной системы по одному из 4 вариантов (см. выбор темы). Пояснительная записка должна содержать:

- Материал по разделам проектирования (см. ниже).
- Выводы
- Список литературы

При защите курсового проекта оцениваются следующие характеристики работы:

- Объем и полнота разработки перечисленных разделов
- Оформление текста пояснительной записки, логика, стиль и грамотность изложения

Выбор темы

1. Может быть выбрана в соответствии с темой магистерской диссертации. Не обязательно это компонента, напрямую входящая в работу. Можно описать проект, имеющий отношение к данной области, или решающий вспомогательную задачу. Например, стенд для моделирования основного функционала проекта и т.п..
2. Тема может быть выбрана из заданий к л.р. семестра 1. При этом бригада может выполнять одну тему, но должны быть выделены компоненты или функционал, например, «Система продажи театральных билетов. Серверная компонента». Тогда часть материала по проектированию может быть общей.
3. Тема может быть аналогично типовой, но предложена самим студентом, например, из выполненных им индивидуальных проектов или проектов, в которых он принимал участие.
4. Для индивидуальной темы объем и содержание разделов может быть изменено по согласованию с преподавателем, например, **полный бизнес-анализ за счет исключения конструирования.**

Разработка должна быть доведена до уровня прототипа: каркас проекта – основные интерфейсы, абстрактные классы, производные классы без функционального наполнения + БД с минимальным (тестовым) наполнением + реализация одного-двух существенных прецедентов (объем проекта порядка 10 пакетов и 100 классов)

Раздел 1. Бизнес-анализ и моделирование предметной области (анализ предметной области)

Исходные данные к проекту. Постановка задачи. История вопроса. Описание окружения. Основные требования (неформальное изложение)

1.1 Анализ предметной области (аналитический обзор).

Первичный сбор материалов (литература, интернет-источники) по всем этапам разработки, ключевым моментам, используемым средствам, существующим решениям. Классификация материала по темам: **существующие аналоги (прототипы), средства разработки, библиотеки, стандарты, проблемы, графический интерфейс, алгоритмы** (в пределах 10 рубрик по 5-15 ссылок в каждой).

Анализ существующих решений и прототипов: выбор классификационных признаков, например, виды поддерживаемого функционала, наличие открытого кода, свободное/коммерческое ПО, объем разработки, год выпуска, стоимость. Краткая аннотация по каждому прототипу. Сводная таблица характеристик. Определение характеристик собственной разработки («экологическая ниша», преимущества).

Сюда же: анализ средств разработки, готовых частных решений, фреймворков по той же схеме: аннотированный обзор, классификационные признаки, сводная таблица характеристик, обоснование выбора. Анализ должен производиться с точки зрения основных требований со стороны системы разработки (например, платформенная независимость, совместимость).

1.2 Системная инженерия (системный анализ)

Краткое описание системы и ее окружения. Список заинтересованных лиц (стейкхолдеров), в т.ч. разработчики, службы сопровождения и эксплуатации, заказчик, пользователи, руководитель ВКРБ, секретарь ГЭК (приемка), «злоумышленники». Описание их интересов в системе и целей, из которых следуют эти интересы. Определение основных **архитектурных видов (аспектов) и архитектурных точек зрения** (см. 6.1. Системная инженерия в папке «Программная инженерия»).

1.3 Глоссарий

Содержательная (не масло масляное) описание **наиболее важных** элементов функционала, архитектуры, реализации, особых или жаргонных терминов, используемых в разработке.

1.4 Бизнес-анализ

- 1. бизнес-требования** – обоснование полезности проекта, особенности проекта, обеспечивающие его привлекательность, предполагаемые отличия от аналогов, проблемы предметной области и способы их решения в проекте, возможности коммерческого использования, способы монетизации.
- 2. границы проекта** – перечень бизнес-процессов, поддерживаемых и не поддерживаемых системой.

3. **перечень пользователей проекта**, их квалификация, типовые задачи, которые ими решаются, специфические требования
4. словесное описание бизнес-процессов или использование диаграмм BPMN (см. папку «Бизнес-анализ» в программной инженерии)

1.5 Видение проекта

Основные данные фазы исследования, включенные в видение:

1. название проекта
2. цели проекта
3. результаты проекта
4. допущения и ограничения
5. ключевые участники и заинтересованные стороны
6. ресурсы проекта
7. сроки
8. риски
9. критерии приемки
10. обоснование полезности проекта

1.6 Моделирование предметной области

1. Модель предметной области как таковой (если проект встраивается в существующие бизнес-процессы)
2. Представление предметной области в программной системе
3. Классы бизнес-объектов на этапе проектирования

Раздел 2. Системная аналитика и архитектурное проектирование (структурное проектирование)

2.1 Анализ требований

1. Диаграммы прецедентов
2. Ключевые сценарии
3. Иерархический список требований, в т.ч. вносимых на более поздних этапах проектирования (GUI, архитектура, системные требования, форматы)

2.2 Графический интерфейс

1. Подробная диаграмма экранов, количественные данные в таблице (ссылки, активные элементы (списки, кнопки, поля, события))
2. Графический дизайн
3. Подсказки, справки в программе, ориентация на различных пользователей, программная документация
4. Обсуждение вопросов поиска, навигации, визуализации данных.

2.3. Архитектура

1. Ключевые вопросы архитектурного решения (основные идеи)
2. Компоненты архитектуры: слои, интерфейсы, протоколы, сервисы, подсистемы
3. Конкретные решения, паттерны, стили, фреймворки (и реализуемые ими перечисленные выше компоненты)
4. стандарты представления и обмена данными (форматы, протоколы)
5. Описание поведения в наиболее существенных или критических состояниях или сценариях.

Описание архитектуры согласно определению:

6. **значимые решения** по поводу организации ПС
7. **структурные элементы и их интерфейсы**, при помощи которых компонуется система
8. **поведение** - взаимодействие между этими элементами
9. **компоновка элементов** в иерархию подсистем
10. **стиль архитектуры** который направляет эту организацию

Краткое описание специфики архитектуры с каждой их сторон ее представления (модель 4+1): функциональное, логическое, процедурное, реализация, развертывание.

Раздел 3. Детальное проектирование и конструирование (функциональное проектирование)

3.1 Детальное проектирование

Архитектурные решения по перечисленному списку ключевых моментов проектирования:

1. ключевые идеи и концепции: абстрагирование, связность и соединение, модульность и декомпозиция, инкапсуляция, разделение интерфейса и реализации
2. параллелизм
3. контроль и обработка событий
4. распределение компонентов
5. обработка ошибок и исключительных ситуаций и обеспечение отказоустойчивости;
6. взаимодействие и представление (Model-View-Controller - MVC)
7. сохраняемость данных (доступность «долгоживущих» данных)
8. виды представления архитектуры - структурное, поведенческое, логическое, физическое, реализация кода
9. архитектурные стили
10. шаблоны проектирования
11. методы проектирования: нисходящее проектирование, модульное, абстрагирование, итеративность, функционально-ориентированное или структурное проектирование, объектно-ориентированное проектирование, проектирование на основе структур данных, компонентное проектирование

Архитектурные классы и диаграммы классов: архитектурные слои: представление, контроллеры, бизнес-модель, DAO (ORM), БД, коммуникации (сеть, протоколы), валидность, исключения.

Основные документы по проектированию:

1. Структура БД
2. ORM- или DAO-объекты
3. Основные бизнес-объекты
4. Форматы файлов (сохранение данных, импорт/экспорт, конфигурирование и настройка)
5. Описание протоколов: базовый протокол (web-сокеты, http, tcp/ip), форматы сообщений, процедуры обмена, синхронный/асинхронный, симметричный/несимметричный, буферизация и мультиплексирование данных, параллелизм, синхронизация.
6. Форматы собственных API проекта
7. Диаграммы состояний – автоматные модели для описания поведения сущностей системы, состояния протокола и т.п..

3.2 Конструирование

Прототип программного проекта в среде разработки в виде набора классов-пустышек. Пакеты: базовые интерфейсы и сервисы, паттерны (параллелизм, исключения, валидность данных), файлы и сериализация, коммуникация к БД, DAO (ORM-классы), бизнес-модель, контроллеры, представление (View), коммуникации с сетью (линейные и протокольные компоненты). Определяются абстракции (абстрактные БД и интерфейсы), а также классы-наследники с соответствующим наполнением. Класс содержит основные данные (свойства), а также заголовки наиболее существенных методов. Ориентировочно в прототипе должно быть порядка 50-70 классов и интерфейсов и порядка 10 пакетов.

По прототипу должна быть справка: количество пакетов, классов, свойств, методов. Эти данные используются при оценке проекта. Данные можно представить в табличной форме

Раздел 4. Тестирование, сопровождение, управление проектом

4.1 Тестирование

1. Модульное тестирование (примеры модульных тестов, возможно в табличной форме)
2. Интеграционное тестирование (описание тестовых сценариев, тестовых наборов, результаты)
3. Специфические тестирование (например, производительности)

4.2. Документация по установке и использованию

1. Руководство по сборке проекта
2. Руководство по установке, конфигурированию, настройке, в т.ч. описание конфигурационных и настроечных файлов и процедур

3. Краткое руководство пользованием, примеры базовых сценариев

4.3. Управление проектом

1. Оценка трудоемкости и сроков исполнения проекта по СОСОМО и по «смете». Во втором случае из проектной документации берутся данные о количестве компонент разработки (таблиц БД, полей, запросов, окон, событий) и берется **примерная** оценка трудоемкости на единицу
2. Список исполнителей проекта, иерархическая структура работ, расписание

Выходные документы

1. Пояснительная записка к КП
2. Исходный код проекта
3. Презентация

Справочные материалы

<http://yadi.sk/d/eKfzLUkMDambr> - «Программная инженерия»

<http://yadi.sk/d/a8aFNiArHc4jn> - «Сетевые протоколы», архитектура клиент-серверных приложений, протоколы

2. Критерии оценки

- проект считается **невыполненным**, если отсутствуют необходимые разделы описания программы, либо программа не работает, либо студент не в состоянии объяснить принципов ее работы (алгоритма) – оценка составляет <50% баллов;
- проект засчитывается на **пороговом** уровне, разработка не полностью соответствует заданию, имеются ошибки при тестировании, пояснительная записка оформлена со значительными структурными, стилистическими и грамматическими ошибками - оценка составляет не более 50-65% баллов
- проект засчитывается на **базовом** уровне, если функционал разработки соответствует заданию, отсутствуют ошибки тестирования, пояснительная записка оформлена в целом грамотно - оценка составляет 65-85% баллов
- проект засчитывается на **продвинутом** уровне, если функционал реализован полностью и эффективно, в пояснительной записке отражены все аспекты структурного, функционального проектирования и тестирования - оценка составляет 85-100% баллов

3. Шкала оценки

В общей оценке по дисциплине баллы за проект учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Перечень тем курсового проекта

1. Система продажи билетов в кинотеатре (клиент, кассир, билетер, директор, планирование сеансов, резервирование мест кассиром, бронирование через Интернет, финансовые отчеты, план зала, анонсы).

2. **Система продажи театральных билетов.** Приложение кассира (множество точек продажи), приложение распространителя, бронирование по интернет, связь с платежными системами, план зала (мета-уровень описания), репертуар, анонсы.
3. **Система автоматизации диспетчерской службы такси** (диспетчер, водитель, клиент, директор, прием заказов, ведение очередей, ручное распределение заказов, приложение водителя)
4. **Система автоматизированного заказа такси через Интернет** (серверное приложение для автоматического распределения заказов с учетом нагрузки, web-приложение и мобильное приложение для клиентов, приложение водителя, приложение администратора для форсмажорных и конфликтных ситуаций)
5. **Система ведения корпоративной адресной базы для мобильных клиентов** (хранение список контактов, обмен контактами, доп. данные по контактам, иерархическая многомерная общая адресная книга).
6. **Мессенджер мобильных клиентов** аналогичный WhatsUp. Регистрация по номеру мобильного телефона. Передача сообщений, файлов, синхронизация адресных книг, поиск абонента по общей адресной книге.
7. **Мессенджер с прямой связью клиентов.** Сервер контактов, регистрация, авторизация с сообщением текущего IP-адреса, прямая связь с передачей сообщений и файлов.
8. **Система заказных грузоперевозок по городу** (клиент, диспетчер, магазин, водитель, прием и оформление заказов, отчеты и сопроводительные документы, распределение заказов диспетчером,).
9. **Система продажи билетов на междугородные автобусы.** Планирование рейсов, визуализация раскладки, платежные терминалы, бронирование через Интернет, сводные отчеты по маршруту, дате.
10. **Система мониторинга междугородных автобусных перевозок** (Планирование рейсов, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций).
11. **Система планирования междугородных транспортных перевозок.** Транспортные средства (тип, тоннаж, вместимость), БД населенных пунктов, дорожной сети, прием заявок, поиск подходящего транспорта с учетом его местонахождения и порожнего прогона, оформление и проводка заявок, отчеты по периоду времени, транспортному средству, водителю.
12. **Система мониторинга междугородных транспортных перевозок** (Планирование поездок, GPS-навигация транспортных средств, отслеживание графика движения по маршруту, обработка аварийных ситуаций, отслеживание заправки, отчеты по расходу топлива и рабочему времени водителей).
13. **Система мониторинга и анализа дорожной обстановки** и движения городского транспорта на основе данных сайта Новосибирскгортранса. (Периодический опрос сервера, сбор данных о состоянии транспортных средств на линии, статистические отчеты по маршруту, временному промежутку – день недели, часы, интервалам движения).
14. **Логистическая система интернет-магазина** с пунктами выдачи и доставкой по городу. Прием заказов, оформление заявок поставщикам, уведомление клиентов, отслеживание работы курьеров, отчеты по пунктам выдачи).
15. **Система авансовых платежей за услуги ЖКХ** (жилец – управляющая компания – поставщик, ведение лицевого счета жильцов, управляющих компаний и поставщиков (эл. энергия, тепло, водоснабжение)), ввод показаний, списывание денежных средств

по факту потребленных услуг с учетом текущего тарифа, подсчет общедомовых расходов.

16. **Система электронного документооборота учебного процесса** в ВУЗе (приказы о зачислении/отчислении, назначении тем дипломов и магистерских, распределение на практику, распоряжения по подразделениям, прохождение приказа: создание, визирование, утверждение, нумерация, рассылка. Мета-уровень описания структуры).
17. **Система книговыдачи библиотеки с использованием QR-кодов.** Систематический и алфавитный каталог книг с учетом экземпляров, генератор QR-кодов, ведение списка читателей, выдача и прием, ведение формуляра (с историей), уведомление о просрочках.
18. **Система книговыдачи библиотеки с использованием штрих-кодов.** Систематический и алфавитный каталог книг с учетом экземпляров, генератор штрих-кодов, ведение списка читателей, выдача и прием, ведение формуляра (с историей), уведомление о просрочках
19. **Система управления кафе/баром** (электронное меню, заказ, расчет, касса, клиент, официант, администратор, интеграция с «1С-Предприятие», электронные платежи, чек).
20. **Система бронирования мест для клубных мероприятий.** План концертов. Анонсы. Стоимость столиков. План зала (мета-уровень описания, настройка под конкретный клуб). Электронная предоплата. Бронирование. Приложение кассира. Мобильное приложение.
21. **Система бронирования мест в гостинице.** Мета-уровень описания конкретной гостиницы – расположение и типы номеров, поэтажные планы, список услуг, фото общие и отдельных номеров, расценки. Бронирование через интернет, визуализация свободных/занятых, расчет стоимости, квитанции, отчеты по периодам.
22. **Система планирования взаимосвязанных работ.** Учет сотрудников и их занятости запланированными работами. Определение работ в виде цепочек заданий с параллельными ветвями. Распределение заданий по работникам с учетом их занятости. Сдвиг сроков зависимых работ при задержке выполнения одной из них. Приложения сотрудника, руководителя подразделения, генерация отчетов.
23. **Система мониторинга обслуживания по заявкам.** Система с предварительным сбором заявок и обслуживанием (врач, электрик, сантехник...). Составление плана исполнения заявок, отслеживание времени исполнения, коррекция времени при задержках с уведомлением клиентов, отказы. Мобильный клиент сотрудника, приложение диспетчера.
24. **Система поддержки технологии SCRUM для удаленной работы.** Разработка основных элементов технологии SCRUM (митинги, беклоги, диаграммы сгорания, контроль исполнения, покер планирование) для разработчиков в режиме удаленной работы.
25. **Система многомерной организации документов.** В качестве документов могут выступать короткие заметки (записи в БД), файлы документов, изображений, звуковых. Поддерживается дерево редактирования версий файла. Создается произвольное количество систем классификаций (каталогов ссылок на файлы) по набору ключевых параметров (например, изображения могут классифицироваться по размеру, цветовой палитре, содержанию, наличию определенных предметов, звуковые файлы по исполнителю, стилю, наличию муз. инструментов).

5. Перечень вопросов к защите курсового проекта

Защита курсового проекта состоит в верификации и тестировании документов и прототипа. С этой целью преподаватель задает вопросы по исполнению отдельных сценариев в проекте разработки, оценивая возможные нестыковки и обнаруживаемые ошибки. Аналогично проверяется полнота проектирования. При необходимости проект может быть отправлен на доработку, которая оформляется в виде отдельного раздела к пояснительной записке.