

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт экономики и предпринимательства

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Современные технологии разработки программного обеспечения

Уровень высшего образования
Магистратура

Направление подготовки / специальность
09.04.03 - Прикладная информатика

Направленность образовательной программы
Интернет-технологии в экономике

Форма обучения
очная, заочная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.07 Современные технологии разработки программного обеспечения относится к обязательной части образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1: Демонстрирует знание современных интеллектуальных технологий решения профессиональных задач ОПК-2.2: Демонстрирует умение обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач	ОПК-2.1: Уметь пользоваться различными средствами разработки Знать основные средства разработки современных технологий Владеть навыками выбора оптимальной среды создания интернет-приложений ОПК-2.2: Уметь проводить анализ и обосновывать выбор среды разработки Знать особенности сред разработки Владеть навыками выбора оптимальной среды разработки	Практическое задание	Экзамен: Кейс-задание Тест
ОПК-5: Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.1: Демонстрирует знание современных информационных технологий, аппаратных платформ и инструментальных программных средств при разработке и модернизации ИС ОПК-5.2: Демонстрирует умение планировать проекты по разработке и модернизации программного и аппаратного обеспечения ИС	ОПК-5.1: Уметь разрабатывать компоненты программных комплексов, использовать современные инструментальные средства и технологии программирования Знать среды современной разработки Владеть основами разработки приложений в разных средах ОПК-5.2:	Практическое задание	Экзамен: Кейс-задание Тест

		<p>Уметь настраивать среду под собственные нужды</p> <p>Знать особенности фреймворков</p> <p>Владеть навыками применения фреймворков для разработки</p>		
<p>ОПК-7: Способен использовать методы научных исследований и математического моделирования в области проектирования и управления информационными системами</p>	<p>ОПК-7.1: Демонстрирует знание методов научных исследований и математического моделирования при проектировании ИС</p> <p>ОПК-7.2: Демонстрирует умение осуществлять методологическое обоснование научного исследования</p>	<p>ОПК-7.1:</p> <p>Уметь использовать систему контроля версий</p> <p>Знать принцип командной разработки приложений</p> <p>Владеть навыками командной разработки</p> <p>ОПК-7.2:</p> <p>Знать особенности работы системы управления версиями</p> <p>Уметь распределять задачи между участниками команды</p> <p>Владеть методами отслеживания за версиями</p>	<p>Практическое задание</p>	<p>Экзамен:</p> <p>Кейс-задание</p> <p>Тест</p>

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная	заочная
Общая трудоемкость, з.е.	6	6
Часов по учебному плану	216	216
в том числе		
аудиторные занятия (контактная работа):		
- занятия лекционного типа	16	4
- занятия семинарского типа (практические занятия / лабораторные работы)	48	14
- КСР	2	2
самостоятельная работа	96	187
Промежуточная аттестация	54	9
	Экзамен	Экзамен

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе	
		Контактная работа (работа во взаимодействии с преподавателем), часы из них	Самостоятельная работа

			Занятия лекционного типа		Занятия семинарского типа (практические занятия/лабораторные работы), часы		Всего		обучающегося, часы		
	0 Ф 0	3 Ф 0	0 Ф 0	3 Ф 0	0 Ф 0	3 Ф 0	0 Ф 0	3 Ф 0	0 Ф 0	3 Ф 0	
Использование интегрированной среды разработки IDE	23	20.5	1	0.5	2	0	3	0.5	20	20	
Использование системы управления версиями Git	33	30.5	2	0.5	6	0	8	0.5	25	30	
Фреймворки для быстрой разработки интернет приложений	13	20.5	1	0.5	2	0	3	0.5	10	20	
Фреймворк Bootstrap.	26	47	4	1	12	6	16	7	10	40	
Фреймворк JQuery	29	52.5	4	0.5	14	2	18	2.5	11	50	
Фреймворк JQuery UI	36	34	4	1	12	6	16	7	20	27	
Аттестация	54	9									
КСР	2	2						2	2		
Итого	216	216	16	4	48	14	66	20	96	187	

Содержание разделов и тем дисциплины

Использование интегрированной среды разработки IDE
Использование системы управления версиями Git
Фреймворки для быстрой разработки интернет приложений
Фреймворк Bootstrap.
Фреймворк JQuery
Фреймворк JQuery UI

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Для обеспечения самостоятельной работы обучающихся используются:

- электронный курс "Современные технологии разработки программного обеспечения" (<https://e-learning.unn.ru/course/view.php?id=4396>).

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-2:

Задание 1.

С помощью стилей бустрап оформите формы и другие компоненты страницы

Задание 2.

Создайте страницу с кнопкой, по нажатию на которой открывается модальное окно. В окне разместите форму с двумя полями и кнопкой. Все стили и окно создайте с помощью фреймворка Bootstrap. Организуйте проверку вводимых данных.

5.1.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-5:

Задание 5.

Используйте верстку Bootstrap для создания страницы.

Задание 6.

Разместите на странице раскрывающийся горизонтальный блок

Задание 7.

Реализуйте адаптивную верстку для страницы из предыдущего задания.

Задание 8.

Используйте фреймворк JQuery для решения задач

1. JQuery Разместите на экране перечень новостей со ссылкой «читать дальше». При нажатии на ссылку открывается остаток статьи.

Подсказки

Блок div – заголовок, для него обрабатывается событие click

Блок div – тело статьи. Первоначально в css задано display:none.

1. Сделайте чекбокс и кнопку. Кнопка не активна, пока чекбокс не включен.

Кнопка не активна с помощью свойства disabled:true

```
<input type="button" disabled id="b1">
```

```
$(".check").click(function()
```

```
{
```

```
$("#b1").prop("disabled",false)
```

```
}
```

1. Расширьте предыдущую задачу – кнопка деактивируется при выключении чекбокс

`$(".check").prop("checked")` – возвращает true, если чекбокс активен

1. При наведении курсора мыши на абзац с `id=par2` он должен стать прозрачным (можно задавать любые значения прозрачности, но необходимо чтобы текст при этом был видим) в течении 3 секунд (3000 миллисекунд). При выведении курсора мыши за пределы абзаца он должен вернуть стандартные значения прозрачности.

5.1.3 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-7:

Задание 1.

Создайте проект в сервисе GitHub. Обеспечьте контроль версий.

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения. Решены все основные задачи с отдельным и несущественными недочетами, выполнены все задания в полном объеме	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продemonстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продemonстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка	Уровень подготовки
--------	--------------------

зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Кейс-задание) для оценки сформированности компетенции ОПК-2

Создайте страницу с версткой Bootstrap.



ТЕКСТОТЕКА

АГЕНТСТВО КОНТЕНТ-МАРКЕТИНГА

ТОЛЬКО ЛУЧШИЕ СТАТЬИ

Подпишись на рассылку и мы будем отправлять тебе раз в неделю наши лучшие статьи, полезные практические советы по контенту и маркетингу.

А еще ты сможешь лично увидеть контент-маркетинг в действии.

Имя

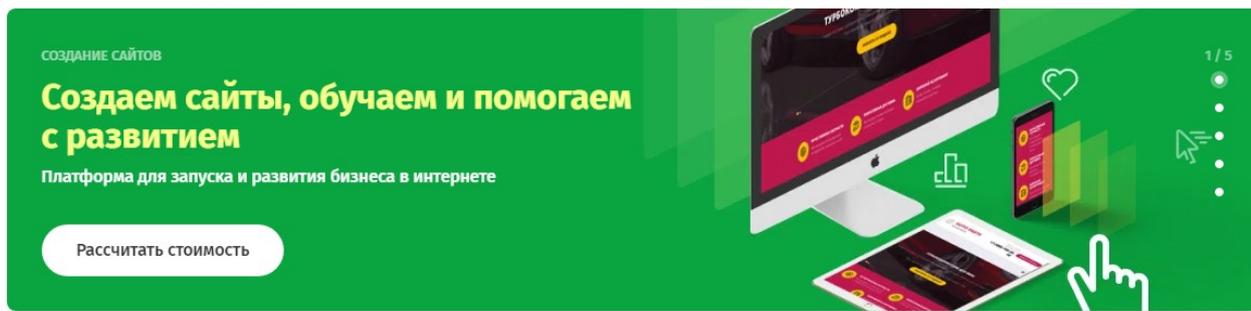
Почта

Подписаться

5.3.2 Типовые задания (оценочное средство - Кейс-задание) для оценки сформированности компетенции ОПК-5

Создать страницу с макетом. Рисунки можно подобрать любые, верстку можно менять на свое усмотрение, но слишком простая не приветствуется. Реализовать адаптивную верстку.

Одинаковые версии не принимаются. Выбирайте свои картинки, цвета, дизайн! Не забудьте добавить слайдер!



Создание сайтов

для малого и среднего бизнеса

<p>ТАРИФ</p> <p>Сайт-Визитка</p> <p>5 950 ₺</p> <p>Подробнее</p>	<p>ТАРИФ</p> <p>Лендинг</p> <p>7 950 ₺</p> <p>6 950 ₺</p> <p>Подробнее</p>	<p>АКЦИЯ ДО 31 АВГУСТА</p> <p>Сайт-Бизнес</p> <p>9 950 ₺</p> <p>от 8 950 ₺</p> <p>Подробнее</p>	<p>АКЦИЯ ДО 31 АВГУСТА</p> <p>Интернет-магазин</p> <p>14 950 ₺</p> <p>от 11 950 ₺</p> <p>Подробнее</p>
--	---	--	---

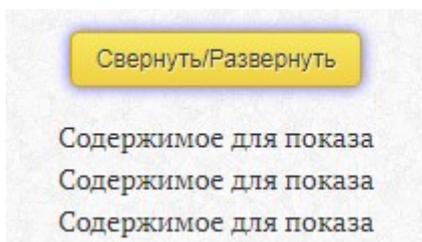
Для созданной страницы реализовать смещение и обводку блока при наведении

Создание сайтов

для малого и среднего бизнеса

<p>ТАРИФ</p> <p>Сайт-Визитка</p> <p>5 950 ₺</p> <p>Подробнее</p>	<p>ТАРИФ</p> <p>Лендинг</p> <p>7 950 ₺</p> <p>6 950 ₺</p> <p>Подробнее</p>	<p>АКЦИЯ ДО 31 АВГУСТА</p> <p>Сайт-Бизнес</p> <p>9 950 ₺</p> <p>от 8 950 ₺</p> <p>Подробнее</p>
--	---	--

Добавьте на страницу набор спойлеров, которые открываются по нажатию.



При нажатии на кнопку «Рассчитать стоимость» должно открыться модальное окно с формой заказа. Например.

Организуйте проверку полей формы – они не должны быть пустыми, телефон и email должны быть введены правильно.

5.3.3 Типовые задания (оценочное средство - Кейс-задание) для оценки сформированности компетенции ОПК-7

Задание 1

Создайте проект в сервисе GitHub. Обеспечьте контроль версий.

Методические указания

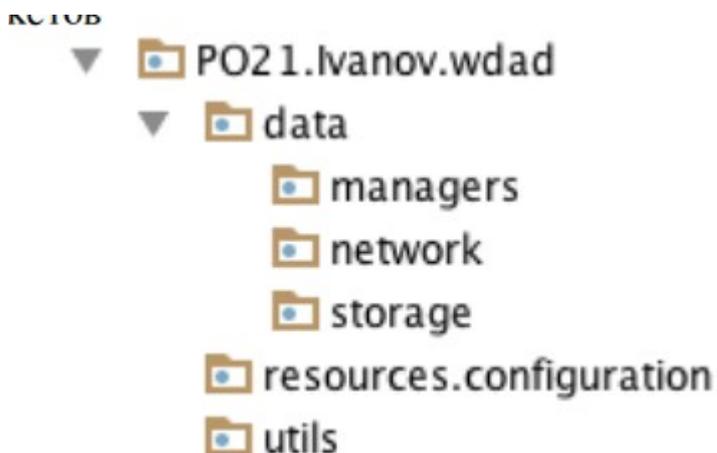
Зарегистрируйтесь в сервисе GitHub.

Установите VCS git на свой компьютер и настройте IDE для работы с данной системой.

Создайте новый проект «starting-monkey-to-human-path».

Создайте пакет «<группа>.<фамилия>.wdad».

Внутри него создайте следующую иерархию пакетов



Создайте новый репозиторий git и установите проект под версионный контроль.
Создайте новый удаленный репозиторий на GitHub из локального (init и commit & push).
Добавьте в соавторы (collaborator) пользователя WildChildCode.

Создайте новую ветку (branch) «initTask».

Добавьте следующие изменения в рамках этой ветки.

В пакете wdad создайте класс Application.

Этот класс должен содержать метод

public static void main(String[] args), выводящий в консоль фразу:

«I'm <Имя> <Фамилия>, and I'm not a monkey»

Зафиксируйте изменения в локальном и удаленном репозиториях (commit & push).

Коммит снабдите комментарием:

Added entry point

На страничке удаленного репозитория на основе последнего коммита создайте пул реквест (Сравниваются ветки master и initTask).

В комментарии при создании пул реквеста (вкладка write) добавьте текст в следующем формате:

“<Группа> <ФИО> инициализацию успешно прошел (-ла). Готов(-а) приступить к выполнению следующего задания”

Ссылку на пул реквест в формате

Группа

ФИО

Ссылка

постим в соответствующем топике группы в VK

Критерии оценивания (оценочное средство - Кейс-задание)

Оценка	Критерии оценивания
превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы

Оценка	Критерии оценивания
	одна компетенция сформирована на уровне «отлично»
очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-2

1. Обычно под термином "программное приложение" понимают:
 - конкретную программу
 - набор программных продуктов
 - инструменты для разработки программ
2. программное обеспечение это
 - инструменты разработки программных приложений
 - как одна программа, так и комплекс (набор) программ
 - то что обеспечивает работу программ
3. Редактор кода это:
 - программа, работающая как текстовый редактор, только более подходящий для написания кода.
 - автоматизированное средство поиска ошибок
 - сотрудник, который проверяет за всеми код
4. Базовое (системное) ПО
 - программы, предназначенные для разработки программного обеспечения
 - Организует процесс обработки информации в компьютере и обеспечивает нормальную рабочую среду для прикладных программ.
5. Операционная система (ОС)
 - программы, предназначенные для разработки программного обеспечения
 - комплекс специальных программных средств, предназначенных для управления загрузкой компьютера, запуском и выполнением других пользовательских программ, а также для планирования и управления вычислительными ресурсами ПК.
6. Инструментальное ПО системные программы это
 - программы, предназначенные для разработки программного обеспечения

- программы, обеспечивающие процесс обработки информации в компьютере и обеспечивает нормальную рабочую среду для прикладных программ.
7. Интегрированная среда разработки(integrated development environment - IDE)
набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.
среда, поставляемая с операционной системой и предназначенная для разработки специфического для этой ОС программного обеспечения
 8. Единая интерактивная оболочка IDE обеспечивает
вызов всех других компонент, не выходя из среды, с широким использованием функциональных клавиш полную интеграцию с инструментами операционной системы
 9. Современные текстовые редакторы в интегрированных средах обеспечивают
режим автоматического завершения кода (code completion), который в них включен по умолчанию и в котором редактор среды подсказывает разработчику кода возможные и синтаксически правильные его продолжения
режим одновременной отладки кода по мере написания программ
 10. Система поддержки сборки (build) IDE обеспечивает
Компиляции проектов из исходных кодов, включающая компилятор с исходного реализуемого языка автоматическую рассылку кода всем участникам, помеченным как разработчики в системе управления проектом
 11. Отладчик (debugger) необходим для
отладки автоматических процессов в среде разработки
отладки программ в среде с помощью типичного набора команд
 12. Профилировщик (profiler) это
инструмент для накопления и анализа статистических данных, полученных в результате исполнения программы под управлением интегрированной среды: число вызовов процедур (методов), объем памяти, используемой при выполнении программы, и т.д.
набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.
 13. Рефакторинг (refactoring) это
инструментарий систематических групповых модификаций программ в среде, без принципиальных изменений их функциональности, с целью улучшения кода.
инструмент для накопления и анализа статистических данных, полученных в результате исполнения программы под управлением интегрированной среды: число вызовов процедур (методов), объем памяти, используемой при выполнении программы, и т.д.
 14. Генератор тестов (unit test generator) это
Инструмент для генерации типовых тестов для тестирования модулей (units) - методов или процедур - с различными возможными сочетаниями значений аргументов.
инструментарий систематических групповых модификаций тестов программ в среде, без принципиальных изменений их функциональности, с целью улучшения кода.
 15. Система управления версиями исходных кодов (source code control system)
обеспечивает поддержку управления версиями файлов исходных кодов проектов в среде при сопровождении программ
обеспечивает поддержку тестирования и выборки из файлов исходных кодов проектов в среде при сопровождении программ
 16. поддержка управления версиями файлов исходных кодов проектов в среде при сопровождении программ обеспечивает
его семантическую корректность, в частности отсутствие некоторых видов ошибок, обнаруживаемых обычно при исполнении, например, недостижимые условия;
отсутствие необходимых проверок и полномочий безопасности и др.
обеспечивает генерацию типовых тестов для тестирования модулей (units) - методов или

процедур - с различными возможными сочетаниями значений аргументов на стороне внешних источников.

17. Инструменты "запутывания" кода (obfuscation)

выполняют замену имен элементов кода - классов, методов, полей и т.д. на непонятные, "случайные", "запутанные" имена, с целью затруднения изучения декомпилированного бинарного кода, для защиты от "взлома" кода злоумышленниками, которые хотят несанкционированным образом присвоить себе новые идеи, содержащиеся в коде, либо изучить его со злостными целями организации атак

представляют собой набор инструментов для разработки и отладки программ, имеющих общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.

18. Поддержка создания программных проектов (projects) и решений (solutions) на основе типовых шаблонов кода (code patterns) позволяет

автоматизировать создание различного рода проектов, предоставляя шаблоны исходного кода и генерируя автоматически необходимые для проекта конфигурационные файлы.

обеспечить поддержку управления версиями файлов исходных кодов проектов в среде при сопровождении программ

19. Microsoft Visual Studio поддерживает языки и технологии

Ajax, ASP.NET, DHTML, JavaScript, JScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML R, Maple, MathR

20. NetBeans бесплатная среда разработки с открытым исходным кодом:

в основном используется для разработки Java приложений

в основном используется для разработки приложений на C++

1	1
2	2
3	1
4	2
5	2
6	1
7	1
8	1
9	1
1	1
0	
1	2
1	
1	1

2	
1	1
3	
1	1
4	
1	1
5	
1	1
6	
1	1
7	
1	1
8	
1	1
9	
2	1
0	

5.3.5 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-5

1. Логическая разметка HTML-документа показывает
Как оформлять тот или иной раздел документа
какую роль играет данный участок документа в его общей структуре
2. Тег HTML это
элемент разметки вида `<Имя_тега>`, который предназначен для указания браузеру каким образом отобразить элемент на странице
метка, которая помечает, к какому смысловому разделу относится блок контента
3. Закрывающий тег используется
когда нужно показать, что контейнер относится к закрытому типу
в случае, если элемент имеет содержимое или является тегом-контейнером, то есть тегом, не содержащим значения, а лишь другие теги
4. В общем случае элемент HTML это
фиксированное закрытое поле, позволяющее добавить контент на веб-страницу
конструкция вида `<Имя_тега> Содержимое элемента </Имя_тега>`
5. Атрибуты HTML необходимы для
 задания тегу признаков принадлежности элементу
сообщения браузеру дополнительной информации о выводе содержимого элемента.

6. Комментарии в HTML имеют вид

```
<!-- это комментарий -->
/*комментарий*/
```

7. Каскадные таблицы стилей представляют собой

простую технологию присоединения стилей к документам HTML

возможность передавать настройки страницы тем документам, на которые с нее ведут гиперссылки

8. Таблица стилей – это

шаблон, который управляет форматированием тегов в документе. Она представляет собой набор правил отображения.

элемент разметки вида <Имя_тега>, который предназначен для указания браузеру каким образом отобразить элемент на странице

9. Любое правило каскадных таблиц стилей состоит из двух частей:

селектора и определения

селектора и элемента

10. Единица em - это

изменяемое значение, которое зависит от размера шрифта текущего элемента

не изменяемое значение, которое зависит от размера шрифта браузера, установленного по умолчанию

11. CSS Flexbox предназначен для

создания гибких макетов

создания сетки

12. С помощью CSS Flexbox можно

очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров.

очень просто и гибко расставить элементы в контейнере, без распределения доступного пространства между ними, но можно выровнять их тем или иным способом даже если они не имеют конкретных размеров.

13. W3C описывает модуль CSS Grid Layout как

технологию создания гибких макетов

систему двумерного макета, оптимизированного для дизайна пользовательского интерфейса

14. HTML код для создания картинки выглядит следующим образом:

```
<IMG src="1.gif">
```

```
<IMG href="1.gif">
```

15. Строка объявления типа документа HTML 5 выглядит следующим образом:

```
<!DOCTYPE HTML5>
```

```
<!DOCTYPE HTML>
```

16. Установка кодировки utf-8 документа HTML выглядит следующим образом:

```
<META charset="utf-8">
```

```
<META content="charset" value="utf-8">
```

17. Код гиперссылки HTML выглядит следующим образом:

```
<A src="http://www.w3.org/"> Web-сайте W3C</A>
```

```
<A href="http://www.w3.org/"> Web-сайте W3C</A>
```

18. Код контейнера меню HTML5 выглядит следующим образом:

```
<nav>меню</nav>
```

```
<menu> Меню</menu>
```

19. Род заголовка первого уровня HTML выглядит следующим образом:

`<h1>Header</h1>`

`<header1>Header</header1>`

20. Код абзаца HTML выглядит следующим образом:

`<par>paragraph</par>`

`<p>paragraph</p>`

1	2
2	1
3	2
4	2
5	2
6	1
7	1
8	1
9	1
1	1
0	
1	1
1	
1	1
2	
1	2
3	
1	1
4	
1	2
5	
1	1
6	

1	2
7	
1	1
8	
1	1
9	
2	2
0	

5.3.6 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ОПК-7

- В основе методологии Waterfall лежит

логическая последовательность шагов, которые должны быть предприняты на протяжении жизненного цикла разработки ПО. Каждый этап согласовывается компетентными сотрудниками, документируется и передаётся дальше.

логическая последовательность шагов, которые должны быть предприняты на протяжении жизненного цикла разработки ПО. Каждый этап завершается и возврат к его выполнению противоречит методологии.
- Главным минусом методологии Waterfall является

неадаптивная структура ПО

неудобство согласования
- Методология RUP – Rational Unified Process подразумевает

идентификацию и устранение рисков только на последнем этапе разработки

раннюю идентификацию и непрерывное (до окончания проекта) устранение основных рисков
- Методология Agile это

метод гибкой разработки программного обеспечения, включающий постоянный возврат на предыдущие стадии разработки

метод гибкой разработки программного обеспечения, предполагающий большое количество итераций
- Четыре центральных идеи Agile Manifesto включают:

люди и взаимодействие важнее, чем процессы и инструменты, работающее ПО важнее, чем исчерпывающая документация, сотрудничество с заказчиком важнее, чем согласование условий контракта, готовность к изменениям важнее, чем следование первоначальному плану.

люди и взаимодействие менее важны, чем процессы и инструменты, работающее ПО обязательно должно сопровождаться исчерпывающей документацией, сотрудничество с заказчиком важнее, чем согласование условий контракта, готовность к изменениям важнее, чем следование первоначальному плану.
- Главным преимуществом Agile по скорости разработки ПО является следующее:

программный продукт готов к использованию на самых ранних этапах его разработки, пусть и не с полной функциональностью.

программный продукт готов к использованию только тогда, когда реализован полный функционал.

7. Методология XP (eXtremal Programming) дает возможность зафиксировать все требования на первом этапе и больше к ним не возвращаться
вести разработку в условиях постоянно меняющихся требований
8. Scrum это гибкий метод управления проектами, целью которого является быстрое и полное удовлетворение требований заказчика
гибкий метод управления проектами, целью которого является повышение производительности труда в командах, ранее парализованных более тяжелыми методологическими процессами.
9. Спринт в Scrum это **короткая итерация, строго ограниченная по времени (обычно 2-4 недели).**
произвольной длины итерация, длительность которой зависит от поставленной задачи
10. RAD это методология, которая во главу угла ставит скорость и удобство разработки. Одно из главных условий — использование систем документирования разработки.
методология, которая во главу угла ставит скорость и удобство разработки. Одно из главных условий — использование языка быстрой разработки.
11. Joint Application Development (JAD) - это методология, нацеленная на максимальную занятость в разработке каждого исполнителя.
методология, нацеленная на максимальную занятость в разработке конечного пользователя.
12. Спиральная модель разработки ПО это организация жизненного цикла ПО, которая не фокусируется на раннем выявлении и уменьшении проектных рисков.
организация жизненного цикла ПО, которая фокусируется на раннем выявлении и уменьшении проектных рисков.
13. К преимуществам методологии Waterfall относятся гибкое отслеживание рисков и возможность изменения сроков разработки
устойчивость к изменению кадрового состава за счет подробного документирования, дисциплина, гибкость на ранних этапах, ориентация на сроки и финансы.
14. Код CSS для установки картинки фона выглядит следующим образом:
`body { background: url("pendant.png");}`
`body { backgroundImage: url("pendant.png");}`
15. Код CSS для установки картинки для маркера списка выглядит следующим образом:
`ul { list-style-image: url("http://png.com/ellipse.png") }`
`ul { listStyleImage: url("http://png.com/ellipse.png") }`
16. Код класса big для установки размера шрифта в два раза больше шрифта по умолчанию выглядит следующим образом:
`.big{ font-size: 2em;}`
`.big{ font-size: 2rem;}`
17. Правило CSS, устанавливающее рамки таблицы разных цветов, толщины 10px с эффектом вдавливания выглядит следующим образом:
`table{`
`border-width: 10px;`
`border-style: solid;`
`border-color: (aqua red navajowhite maroon);`
`}`
`table{`
`border-width: 10px;`

```
border-style: inset;
border-color: aqua red navajowhite maroon;
}
```

18. Правило CSS для элемента `header`, устанавливающее flex разметку с отступом между блоками прижатыми к краям контейнера выглядит следующим образом:

```
header{
  display: flex;
  justify-content: space-between;
}
header{
  display: flexbox;
  justify-content: spaceBetween;
}
```

19. Правило CSS для элемента контейнера устанавливающее flex разметку с элементами, прижатыми по вертикали к верхней части контейнера выглядит следующим образом:

```
.container-content{
  display: flex;
  align-items: flex-start;
}
.container-content{
  display: flexBox;
  align-items: flexStart;
}
```

20. Правило CSS для элемента контейнера устанавливающее главной осью вертикальную выглядит следующим образом:

```
.container-content{
  display: flexbox;
  flex-direction: column;
}
.container-content{
  display: flex;
  flex-direction: column
}
```

1	1
2	1
3	2
4	2
5	1
6	1

7	1
8	2
9	1
1	2
0	
1	1
1	
1	2
2	
1	2
3	
1	1
4	
1	1
5	
1	2
6	
1	2
7	
1	1
8	
1	1
9	
2	1
0	

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
превосходно	20 из 20

Оценка	Критерии оценивания
отлично	20 из 20
очень хорошо	18-19
хорошо	16-17
удовлетворительно	12-15
неудовлетворительно	менее 12
плохо	менее 12

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Полуэктова Н. Р. Разработка веб-приложений : учебное пособие / Н. Р. Полуэктова. - Москва : Юрайт, 2022. - 204 с. - (Высшее образование). - URL: <https://urait.ru/bcode/496682> (дата обращения: 14.08.2022). - ISBN 978-5-534-13715-6 : 699.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=818653&idb=0>.
2. Чернышев С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие / С. А. Чернышев. - Москва : Юрайт, 2023. - 176 с. - (Высшее образование). - ISBN 978-5-534-14383-6. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=839657&idb=0>.

Дополнительная литература:

1. Разработка одностраничных веб-приложений / Майкл С. Миковски, Джош К. Пауэлл. - Москва : ДМК-пресс, 2014., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=647188&idb=0>.
2. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных WEB-сайтов. - СПб. : БХВ-Петербург, 2011. - 414 с. - (Профессиональное программирование). - ISBN 978-5-9775-0596-3 : 189.00., 2 экз.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

Справочник по HTML [Электронный ресурс]. - <http://htmlbook.ru/html>
Справочник CSS [Электронный ресурс]. - <http://html5book.ru/css>
HTML5BOOK.RU – HTML, CSS, JavaScript и jQuery [Электронный ресурс]. - <https://html5book.ru/Cascading Style Sheets, level 1> <http://www.w3.org/TR/CSS1/>
HTML 4.01 Specification <http://www.w3.org/TR/html401/>

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 09.04.03 - Прикладная информатика.

Автор(ы): Беспалько Анна Андреевна, кандидат педагогических наук, доцент.

Заведующий кафедрой: Трифонов Юрий Васильевич, доктор экономических наук.

Программа одобрена на заседании методической комиссии от 12.12.2023, протокол № 6.