

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ

Н.Г. ЧЕРНЫШЕВСКОГО»

Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ
Декан факультета КНиИТ
С.В. Миронов

"13" сентября 2021 г.

Рабочая программа дисциплины
ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

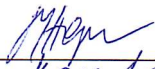

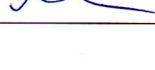
Направление подготовки бакалавриата
09.03.01 Информатика и вычислительная техника

Профиль подготовки бакалавриата
Вычислительные машины, комплексы, системы, сети

Квалификация выпускника
Бакалавр

Форма обучения
Очная

Саратов,
2021

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	Портенко М.С.		13.09.2021 г.
Председатель НМК	Кондратова Ю.Н.		13.09.2021 г.
Заведующий кафедрой	Огнева М.В.		13.09.2021 г.
Специалист Учебного управления			__._.2021 г.

1. Цели освоения дисциплины «Технологии программирования»

Целями освоения дисциплины «Технологии программирования» являются:

- получение базовых знаний о функциональных возможностях платформ, предоставляющих услуги организации этапов жизненного цикла программного обеспечения, трекинга задач, систем контроля версий программного обеспечения;
- формирование основных навыков работы в команде по проектированию и разработке программного обеспечения;
- владение применением стандартных паттернов проектирования;
- формирование умения поиска нестандартных схем взаимодействия модулей, компонентов и типов в системе;
- формирование умения проектирования и разработки программного обеспечения
- формирования навыка описания юзабилити-проблем, с которыми сталкивается пользователь.

2. Место дисциплины в структуре ООП

Данная учебная дисциплина Б1.О.18 «Технологии программирования» относится к части, формируемой участниками образовательных отношений, Блока 1 «Дисциплины (Модули)» учебного плана ООП и направлена на формирование у обучающихся профессиональных компетенций.

Для изучения дисциплины необходимы компетенции, сформированные в ходе изучения Б1.О.06 «Информационные технологии и программирование», Б1.О.09 «Математическая логика и теория алгоритмов», Б1.О.13 «Структуры данных и алгоритмы», Б1.О.19 «Базы данных».

Компетенции, сформированные при изучении данной дисциплины, используются при подготовке выпускной квалификационной работы, будут применяться при профессиональной деятельности.

3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора (индикаторов) достижения компетенции	Результаты обучения
ОПК – 8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения.	ОПК-8.1. Знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ОПК-8.2.	Знать: -виды жизненного цикла программного обеспечения, -технологии программирования, способствующие грамотному планированию и разработке программного обеспечения, -шаблоны проектирования, -платформы разработки, -системы контроля версий, -системы трекинга задач.

	<p>Уметь: применять навыки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-8.3.</p> <p>Владеть: навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>	<p>Уметь:</p> <ul style="list-style-type: none"> -определять схожесть и различия принципов языков программирования, -опираться на собственные знания в изучении новых языков программирования, -определять реляционные и/или нереляционные базы данных необходимы на проекте, -выявлять и анализировать характеристики облачных платформ; <p>Владеть</p> <ul style="list-style-type: none"> -навыками программирования на C# (Java), -навыком работы в команде по проектированию и разработке программного обеспечения.
<p>ПК – 5</p> <p>Проводить юзабилити-исследование программных продуктов и/или аппаратных средств.</p>	<p>ПК-5.1.</p> <p>Знать: основные направления развития аппаратно-программных средств</p> <p>ПК-5.2.</p> <p>Уметь: учитывать современный уровень развития информационных технологий, осуществлять выбор перспективных технологических и архитектурных решений в процессе разработки информационных систем</p> <p>ПК-5.3.</p> <p>Владеть: навыками анализа и оценки информационных ресурсов для применения эффективных аппаратных и программных информационных технологий</p>	<p>Знать</p> <p>Уметь:</p> <ul style="list-style-type: none"> -описывать требования к функциональности и интерфейсу продукта с точки зрения потребностей пользователей и особенностей их поведения, -выявлять структуру взаимодействия элементов системы, особенности возможного масштабирования и применять приемы проектирования и разработки программного обеспечения, учитывая преимущества, с учетом планирования возможных изменений; <p>Владеть:</p> <ul style="list-style-type: none"> - навыком описания юзабилити-проблем, с которыми сталкивается пользователь при выполнении задач в интерфейсе, - навыком анализа функциональности платформ и сред для определения необходимых инструментов для разработки информационных систем.

4. Структура и содержание дисциплины «Технологии программирования»

Общая трудоемкость дисциплины составляет 7 зачетных единиц 252 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			КС Р	Формы текущего контроля успеваемости (по неделям семестра) Формы промежуточной аттестации (по семестрам)
				Лек ции	Практические занятия			
					Обща я трудо емкос ть	Из них – практ ическ ая подго товка		
1	Сериализация	7	1-2	4	4	0	4	Практические задачи
2	Системы контроля версий	7	3	2	2	0	2	Практические задачи
3	Сравнение концепций реализации принципов объектно-ориентированного подхода в C# и Java	7	4-5	4	4	0	4	Выступления с результатом исследования
4	Шаблоны проектирования (23 шаблона GoF)	7	6-10	10	10	0	10	Практические задачи. Контрольная работа
5	Паттерны Grasp (General Responsibility Assignment Software Patterns)	7	11	2	2	0	2	Беседа по теме
6	Принципы SOLID (single responsibility, open-closed, Liskov substitution, interface segregation и dependency inversion)	7	12	2	2	0	2	Беседа по теме
7	Использование платформ для решения задач (Платформа Unity для C#)	7	13-17	10	10	0	34	Проектирование и реализация проекта в парах
	7 семестр		144	34	34	0	40	
	Промежуточная аттестация		36					Экзамен

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				КС Р	Формы текущего контроля успеваемости (<i>по неделям семестра</i>) Формы промежуточной аттестации (<i>по семестрам</i>)
				Лек ции	Практические занятия		-		
					Обща я трудо емкос ть	Из них – практ ическ ая подго товка			
8 семестр									
8	Системы управления совместной работой	8	1-2	4	4	-	4	Выполнение практического задания, обсуждение	
10	Нереляционные базы данных, сравнение с реляционными, платформы, возможности для студенческих проектов и промышленного использования	8	3-12	20	20	-	24	Контрольная работа Творческие задачи	
11	Исследование платформ и инструментов, связанных с исполнением творческих заданий и проектов	8	13-16	8	8	-	16	Выступления	
8 семестр			108	32	32	-	44	Зачет, контрольная работа	
Итого			252						

7 семестр

Сериализация

Хранение состояния объекта, обмен данными между компонентами системы, стандарты файлов: бинарный, XML, JSON.

Системы контроля версий

Управление изменениями проекта в индивидуальной и командной работе, начало работы, ежедневный цикл (обновление рабочей копии, модификация, фиксация изменений), команды, интеграция со средами разработки. На примере, возможно, GIT, SVN, Mercurial.

Сравнение концепций реализации принципов объектно-ориентированного подхода в C# и Java

Сравнение объектно-ориентированной модели C# и Java. Наследование, понятие интерфейса, классы object в обоих языках. Сравнение работы со строками в C# и Java. Сравнение иерархии коллекций в C# и Java. Сравнение Linq и Stream в C# и Java.

Шаблоны проектирования (23 шаблона GoF)

Набор типовых решений без привязки к конкретной задаче.

Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений.

Порождающие паттерны: Abstract factory (Абстрактная фабрика), Builder (Строитель), Factory method (Фабричный метод), Prototype (Прототип) и Singleton (Одиночка).

Структурные паттерны: Adapter (Адаптер), Bridge (Мост), Composite (Компоновщик), Decorator (Декоратор), Facade (Фасад), Flyweight (Приспособленец), Proxy (Заместитель).

Паттерны поведения Strategy (Стратегия), Chain of Responsibility (Цепочка обязанностей), Observer (Наблюдатель), Command (Команда), Iterator (Итератор), Mediator (Посредник), Memento (Хранитель), State (Состояние), Visitor (Посетитель), Template method (Шаблонный метод), Interpreter (Интерпретатор).

Паттерны GRASP

General Responsibility Assignment Software Patterns

Информационный эксперт (Information Expert), Создатель (Creator), Контроллер (Controller), Слабое зацепление (Low Coupling), Высокая связность (High Cohesion), Полиморфизм (Polymorphism), Чистая выдумка (Pure Fabrication), Перенаправление (Indirection), Устойчивость к изменениям (Protected Variations)

Принципы SOLID

(Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion)

Принципы: принцип единственности ответственности, принцип открытости-закрытости, подстановки Лисков, принцип разделения интерфейса, принцип инверсии зависимостей.

Использование платформ для решения задач

(Платформа Unity для C#)

Платформа, назначение, визуальные и анимационные эффекты, написание скриптов (на языке C#), работа с мобильными устройствами, 2D, 3D, обработка физических законов, совместимость с другими интерфейсами.

8 семестр

Системы управления совместной работой

Жизненный цикл разработки программного обеспечения. Системы решений для управления совместной работой. На примере, возможно, Jira, Штаб, Wrike, Битрикс 24.

Нереляционные базы данных

Нереляционные базы данных. Классификация. Примеры платформ. Возможности облачных платформ. Возможности платформ для студенческих проектов и промышленного использования. Сравнение с реляционными. (Возможные примеры: BASE, MongoDB, CouchDB, Cassandra, AWS DynamoDB, Azure Document DB, Yandex Managed Service for MongoDB, YandexDB из YandexCloud).

Исследование платформ и инструментов, связанных с исполнением творческих заданий и проектов

Выбранные проекты могут содержать необходимость применения определенных библиотек или API, студентам дается возможность рассказать о функциональности и способах интеграции того или иного инструмента в свой проект. Например, API карт, языковые словари, возможности библиотек кодирования, трансформации форматов, написание ботов для тех или иных соцсетей и так далее.

5. Образовательные технологии, применяемые при освоении дисциплины «Технологии программирования»

При преподавании дисциплины «Технологии программирования» предусматривается широкое использование в учебном процессе активных и интерактивных форм проведения занятий.

В рамках практических занятий по данной дисциплине используются кейс-задания, выполнение которых направлено на формирование таких профессиональных действий как выявление применимости паттерна программирования (иного инструмента и технологии программирования), нахождения компромиссов, определение назначения, устранения влияния заблуждений и критического отношения к источникам информации. Предусмотрено время на подготовку, проведение и обсуждение выступлений студентов по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской темы. В ходе работы используются тренажеры (например, sql-ex.ru по работе с запросами к реляционной базе данных). Предусмотрена творческая задача, над которой работа происходит в парах.

Для организации работы студентов в рамках данной дисциплины предусмотрена поддержка в рамках курса на платформе системы управления учебным процессом LMS Moodle (course.sgu.ru). Система позволяет выставлять задание, на которое студент может ответить в виде файла с

аннотацией, возможна организация коллективной работы через выкладывание материалов выступлений на форуме, организован выбор доступных тем для выступлений и т.д. Система также имеет функционал для людей с ограниченными возможностями здоровья.

При обучении лиц с ограниченными возможностями здоровья и инвалидов используются подходы, способствующие созданию безбарьерной среды: технологии дифференциации и индивидуализации обучения, увеличение времени на самостоятельное освоение материала.

Удельный вес занятий, проводимых в интерактивных формах, составляет 50% аудиторных занятий.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

В рамках самостоятельной работы студенты изучают материалы, разобранные на контактных занятиях, изучают литературу и другие форматы источников, выполняют домашние задания, готовят выступления, реализуют творческие задания, проекты, решают задачи на тренажерах с возможностью редакции кода. Подробнее – в фонде оценочных средств.

Тренажеры:

1. Тренировка запросов к реляционной базе данных sql-ex.ru
2. Тренировка с проверкой правильности кода по темам JSON
w3schools.com/js/js_json.asp
3. Тренировка запросов к реляционной базе данных MySQL
w3schools.com/mysql/default.asp
4. SQL Tutorial и тренировка запросов к реляционной базе данных
w3schools.com/sql/default.asp

Примерные темы студенческих выступлений и исследований

1. Сериализация в ... (новый язык программирования, отличный от рассматриваемого в рамках занятий)
2. Паттерн программирования ... (отличный от рассматриваемых в рамках занятий)
3. Работа с нереляционной базой данных MongoDB, интеграция с ... (возможные варианты языков программирования и платформ).
4. Уровень нетарифицируемого использования (free tier) для сервисов экосистемы бессерверных вычислений Yandex Cloud.
5. Перенос коллекций из MongoDB в Managed Service for MongoDB в экосистеме Yandex.Cloud.
6. Работа с системой контроля версий ... (отличной от разбираемой на занятиях)
7. Работа с системой управления проектами ... (отличной от разбираемой на занятиях)

Примеры заданий Контрольной работы 1. 8 семестр.

Контрольная проводится на тренажере sql-ex.ru, где есть возможность проверки правильности ответа. Контрольная проводится для осознания, готовы ли студенты к внедрению знаний дисциплины «Базы данных» в проекты дисциплины «Технологии программирования».

Схема БД состоит из четырех таблиц: Product(maker, model, type), PC(code, model, speed, ram, hd, cd, price), Laptop(code, model, speed, ram, hd, price, screen), Printer(code, model, color, type, price). Таблица Product представляет производителя (maker), номер модели (model) и тип ('PC' - ПК, 'Laptop' - ПК-блокнот или 'Printer' - принтер). Предполагается, что номера моделей в таблице Product уникальны для всех производителей и типов продуктов. В таблице PC для каждого ПК, однозначно определяемого уникальным кодом – code, указаны модель – model (внешний ключ к таблице Product), скорость - speed (процессора в мегагерцах), объем памяти - ram (в мегабайтах), размер диска - hd (в гигабайтах), скорость считывающего устройства - cd (например, '4x') и цена - price. Таблица Laptop аналогична таблице PC за исключением того, что вместо скорости CD содержит размер экрана -screen (в дюймах). В таблице Printer для каждой модели принтера указывается, является ли он цветным - color ('y', если цветной), тип принтера - type (лазерный – 'Laser', струйный – 'Jet' или матричный – 'Matrix') и цена - price.

Задача №1

Найдите номер модели, скорость и размер жесткого диска для всех ПК стоимостью менее 500 дол. Вывести: model, speed и hd

Ответ:

```
SELECT model, speed, hd  
FROM PC  
WHERE price<500
```

Задача №2

Найдите производителей принтеров. Вывести: maker

Ответ:

```
SELECT DISTINCT maker FROM Product WHERE type='Printer'
```

Задача №3

Найдите номер модели, объем памяти и размеры экранов ПК-блокнотов, цена которых превышает 1000 дол.

Ответ:

```
SELECT model, ram, screen FROM Laptop WHERE price>1000
```

Задача №4

Найдите все записи таблицы Printer для цветных принтеров.

Ответ:

```
SELECT * FROM Printer WHERE color= 'y'
```

Задача №5

Найдите номер модели, скорость и размер жесткого диска ПК, имеющих 12х или 24х CD и цену менее 600 дол.

Ответ:

```
SELECT model, speed, hd FROM PC
WHERE ((cd='12x' or cd='24x')
and price<600)
```

Задача №6

Для каждого производителя, выпускающего ПК-ноутбуки с объёмом жесткого диска не менее 10 Гбайт, найти скорости таких ПК-ноутбуки. Вывод: производитель, скорость.

Ответ:

```
SELECT DISTINCT p.make, l.speed
FROM laptop l
JOIN product p ON p.model = l.model
WHERE l.hd >= 10
```

Задача №7

Найдите номера моделей и цены всех имеющихся в продаже продуктов (любого типа) производителя В (латинская буква).

Ответ:

```
SELECT PC.model, price
FROM PC, Product
WHERE PC.model=Product.model and Product.make='B'
UNION SELECT Laptop.model, price
FROM Laptop, Product
WHERE Laptop.model=Product.model and Product.make='B'
UNION SELECT Printer.model, price
FROM Printer, Product
WHERE Printer.model=Product.model and Product.make='B'
```

Задача №8

Найдите производителя, выпускающего ПК, но не ПК-ноутбуки.

Ответ:

```
SELECT DISTINCT make
FROM product
WHERE type = 'pc'
EXCEPT
SELECT DISTINCT product.make
FROM product
WHERE type = 'laptop'
```

Задача №9

Найдите производителей ПК с процессором не менее 450 Мгц. Вывести: Make

Ответ:

```
SELECT DISTINCT product.make FROM pc
INNER JOIN product ON pc.model = product.model
WHERE pc.speed >= 450
```

Задача №10

Найдите модели принтеров, имеющих самую высокую цену. Вывести: model, price

Ответ:

```
SELECT model, price FROM printer  
WHERE price =(SELECT MAX(price) FROM printer )
```

Задача №11

Найдите среднюю скорость ПК.

Ответ:

```
SELECT AVG(speed) FROM pc
```

Задача №12

Найдите среднюю скорость ПК ноутбуков, цена которых превышает 1000 дол.

Ответ:

```
Select avg(speed) from Laptop  
where price>1000
```

Контрольные вопросы по освоению дисциплины «Технологии программирования»

1. Сериализация.
2. Сериализация бинарная. (Работа с классом BinaryFormatter C#)
3. Сериализация, абстракция. (Работа с классом SoapFormatter C#)
4. Сериализация XML формат. (Работа с классом XmlSerializer C#)
5. JSON сериализация.
6. Чтение UML диаграмм.
7. Система управления версиями программного обеспечения. Git. Варианты подключений и ПО. Основные команды.
8. Система управления версиями программного обеспечения. SVN. Варианты подключений и ПО. Основные команды.
9. Система управления версиями программного обеспечения Mercury. Варианты подключений и ПО. Основные команды.
10. Сравнение объектно-ориентированной модели C# и Java. Наследование, понятие интерфейса, классы object в обоих языках.
11. Сравнение работы со строками в C# и Java.
12. Сравнение иерархии коллекций в C# и Java.
13. Сравнение Linq и Stream в C# и Java.
14. Паттерны проектирования: назначение.
15. Паттерны проектирования: компромиссы.
16. Паттерны проектирования: заблуждения.
17. Паттерны проектирования: преимущества.
18. Паттерны проектирования: планирование будущих изменений.
19. Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Порождающий паттерн: Абстрактная фабрика (Abstract Factory).

- 20.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Порождающий паттерн: Строитель (Builder).
- 21.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Порождающий паттерн: Одиночка (Singleton).
- 22.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Порождающий паттерн: Фабричный метод (Factory method).
- 23.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Порождающий паттерн: Прототип (Prototype).
- 24.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Адаптер (Adapter или Wrapper – Обертка).
- 25.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Декоратор (Decorator).
- 26.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Фасад (Facade).
- 27.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Приспособленец (Flyweight).
- 28.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Заместитель (Proxy).
- 29.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Компоновщик (Composite).
- 30.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Структурный паттерн: Мост (Bridge).
- 31.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Стратегия (Strategy или Policy – Политика).
- 32.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Команда(Command).
- 33.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Итератор (Iterator).
- 34.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Цепочка обязанностей (Chain of responsibility).

- 35.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Посредник (Mediator).
- 36.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Хранитель (Memento).
- 37.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Наблюдатель (Observer).
- 38.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Состояние (State).
- 39.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Посетитель (Visitor).
- 40.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Шаблонный метод (Template method).
- 41.Схема, назначение, компромиссы, заблуждения, преимущества, планирование будущих изменений. Паттерн поведения: Интерпретатор (Interpreter).
- 42.Паттерны GRASP (General Responsibility Assignment Software Patterns): Информационный эксперт (Information Expert).
- 43.Паттерны GRASP (General Responsibility Assignment Software Patterns): Низкая связанность (Low Coupling).
- 44.Паттерны GRASP (General Responsibility Assignment Software Patterns): Высокое зацепление (High Cohesion).
- 45.Паттерны GRASP (General Responsibility Assignment Software Patterns): Устойчивый к изменениям (Protected Variations).
- 46.Паттерны GRASP (General Responsibility Assignment Software Patterns): Контроллер (Controller).
- 47.Паттерны GRASP (General Responsibility Assignment Software Patterns): Полиморфизм (Polymorphism).
- 48.Паттерны GRASP (General Responsibility Assignment Software Patterns): Чистая выдумка (Pure Fabrication).
- 49.Паттерны GRASP (General Responsibility Assignment Software Patterns): Посредник (Indirection).
- 50.Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип единственности ответственности.
- 51.Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип открытости-закрытости.
- 52.Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): подстановки Лисков.

53. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип разделения интерфейса.
54. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип инверсии зависимостей.
55. Платформа Unity, назначение.
56. Платформа Unity, физические процессы, возможности моделирования от платформы.
57. Платформа Unity, особенности приложений под мобильные устройства.
58. Платформа Unity, написание скриптов.
59. Платформа Unity, интеграция с другими инструментами (например, SQLite и т.д.).
60. Жизненный цикл разработки программного обеспечения.
61. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Jira
62. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Штаб.
63. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Wrike
64. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Битрикс.
65. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере И т.д.
66. Нереляционные базы данных. Классификация. Примеры
67. Нереляционные базы данных. Возможности облачных платформ.
68. Нереляционные базы данных. Возможности платформ для студенческих проектов и промышленного использования.
69. Нереляционные базы данных. Сравнение с реляционными.
70. Нереляционные базы данных. Особенности работы с примерами (Возможные примеры: BASE, MongoDB, CouchDB, Cassandra, AWS DynamoDB, Azure Document DB, Yandex Managed Service for MongoDB, YandexDB из YandexCloud).
71. Функциональные возможности появляющихся облачных платформ.

Экзаменационные вопросы 7 семестр

1. Бинарная сериализация.
2. JSON сериализация.
3. UML диаграммы – элементы, значение.
4. Система управления версиями программного обеспечения. Git.

5. Система управления версиями программного обеспечения. SVN.
6. Система управления версиями программного обеспечения Mercury.
7. Реализация множественного наследования в C# и Java. Проблемы множественного наследования.
8. Сравнение работы со строками в C# и Java.
9. Сравнение иерархии коллекций в C# и Java.
10. Сравнение Linq и Stream в C# и Java.
11. Паттерны проектирования: назначение. Примеры.
12. Паттерны проектирования: компромиссы. Примеры.
13. Паттерны проектирования: заблуждения. Примеры.
14. Паттерны проектирования: преимущества. Примеры.
15. Паттерны проектирования: планирование будущих изменений.
16. Порождающий паттерн: Абстрактная фабрика (Abstract Factory).
17. Порождающий паттерн: Строитель (Builder).
18. Порождающий паттерн: Одиночка (Singleton).
19. Порождающий паттерн: Фабричный метод (Factory method).
20. Порождающий паттерн: Прототип (Prototype).
21. Структурный паттерн: Адаптер (Adapter или Wrapper – Обертка).
22. Структурный паттерн: Декоратор (Decorator).
23. Структурный паттерн: Фасад (Facade).
24. Структурный паттерн: Приспособленец (Flyweight).
25. Структурный паттерн: Заместитель (Proxy).
26. Структурный паттерн: Компоновщик (Composite).
27. Структурный паттерн: Мост (Bridge).
28. Паттерн поведения: Стратегия (Strategy или Policy – Политика).
29. Паттерн поведения: Команда (Command).
30. Паттерн поведения: Итератор (Iterator).
31. Паттерн поведения: Цепочка обязанностей (Chain of responsibility).
32. Паттерн поведения: Посредник (Mediator).
33. Паттерн поведения: Хранитель (Memento).
34. Паттерн поведения: Наблюдатель (Observer).
35. Паттерн поведения: Состояние (State).
36. Паттерн поведения: Посетитель (Visitor).
37. Паттерн поведения: Шаблонный метод (Template method).
38. Паттерн поведения: Интерпретатор (Interpreter).
39. Паттерны GRASP (General Responsibility Assignment Software Patterns): Информационный эксперт (Information Expert).
40. Паттерны GRASP (General Responsibility Assignment Software Patterns): Низкая связанность (Low Coupling).
41. Паттерны GRASP (General Responsibility Assignment Software Patterns): Высокое зацепление (High Cohesion).
42. Паттерны GRASP (General Responsibility Assignment Software Patterns): Устойчивый к изменениям (Protected Variations).
43. Паттерны GRASP (General Responsibility Assignment Software Patterns): Контроллер (Controller).

44. Паттерны GRASP (General Responsibility Assignment Software Patterns): Полиморфизм (Polymorphism).
45. Паттерны GRASP (General Responsibility Assignment Software Patterns): Чистая выдумка (Pure Fabrication).
46. Паттерны GRASP (General Responsibility Assignment Software Patterns): Посредник (Inderection).
47. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип единственности ответственности.
48. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип открытости-закрытости.
49. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): подстановки Лисков.
50. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип разделения интерфейса.
51. Принципы SOLID (Single responsibility, open–closed, Liskov substitution, Interface segregation и Dependency inversion): принцип инверсии зависимостей.
52. Компоненты платформы Unity, назначение, особенности использования.
53. Интеграция платформы Unity с базами данных (например, SQLite).

Экзаменационные вопросы 8 семестр

1. Жизненный цикл разработки программного обеспечения.
2. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Jira
3. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Штаб.
4. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Wrike
5. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере Битрикс.
6. Системы решений для управления совместной работой. Функционал. Особенности платформы. Коммерческое и некоммерческое использование. На примере любой системы.
7. Нереляционные базы данных. Классификация. Примеры
8. Нереляционные базы данных. Возможности платформ для студенческих проектов и промышленного использования.
9. Нереляционные базы данных. Сравнение с реляционными.

10. Нереляционные базы данных. Особенности работы с примерами MongoDB
11. Нереляционные базы данных. Особенности работы с YandexDB из YandexCloud).
12. Нереляционные базы данных. Особенности работы с графовыми базами данных.

7. Данные для учета успеваемости студентов в БАРС

Таблица 1.1 Таблица максимальных баллов по видам учебной деятельности.

1	2	3	4	5	6	7	8	9
Семестр	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
7	20	30	0	20	0	0	30	100
8	20	30	0	20	0	0	30	100

Программа оценивания учебной деятельности студента

7 семестр

Лекции: Посещаемость, активность, выступления с сообщениями – 20 баллов.

Лабораторные занятия: Посещаемость, активность, своевременность решения задач и предоставления объяснений по ходу решения, подготовка выступлений – от 0 до 30 баллов.

Практические занятия: Не предусмотрены.

Самостоятельная работа: Выполнение домашних работ, реализация проекта, подготовка выступлений – от 0 до 20 баллов.

Автоматизированное тестирование: Не предусмотрено.

Другие виды учебной деятельности: Не предусмотрено.

Промежуточная аттестация: (экзамен) – от 0 до 30 баллов

При определении разброса баллов при аттестации преподаватель может воспользоваться следующим примером ранжирования:

- 25-30 баллов – ответ на «отлично»
- 20-24 баллов – ответ на «хорошо»
- 15-19 баллов – ответ на «удовлетворительно»
- 0-14 баллов – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 7 семестр по дисциплине «Технологии программирования» составляет 100 баллов.

Таблица 2.1 Таблица пересчета полученной студентом суммы баллов по дисциплине «Технологии программирования» в оценку (экзамен):

90-100 баллов	«отлично»
80-89 баллов	«хорошо»
65-79 баллов	«удовлетворительно»
0-64 баллов	«неудовлетворительно»

8 семестр

Лекции: Посещаемость, активность, выступления с сообщениями – 20 баллов.

Лабораторные занятия: Посещаемость, активность, своевременность решения задач и предоставления объяснений по ходу решения, подготовка выступлений – от 0 до 30 баллов.

Практические занятия: Не предусмотрены.

Самостоятельная работа: Выполнение домашних работ, реализация проекта, подготовка выступлений – от 0 до 20 баллов.

Автоматизированное тестирование: Не предусмотрено.

Другие виды учебной деятельности: Не предусмотрено.

Промежуточная аттестация: (экзамен) – от 0 до 30 баллов

При определении разброса баллов при аттестации преподаватель может воспользоваться следующим примером ранжирования:

- 25-30 баллов – ответ на «отлично»
- 20-24 баллов – ответ на «хорошо»
- 15-19 баллов – ответ на «удовлетворительно»
- 0-14 баллов – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 8 семестр по дисциплине «Технологии программирования» составляет 100 баллов.

Таблица 2.2 Таблица пересчета полученной студентом суммы баллов по дисциплине «Технологии программирования» в оценку (зачет):

75 баллов и более	«зачтено» (при недифференцированной оценке)
меньше 75 баллов	«не зачтено»

8. Учебно-методическое и информационное обеспечение дисциплины.

а) литература:

1. Гагарина, Л.Г. Введение в архитектуру программного обеспечения: учебное пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. – Москва: ФОРУМ: ИНФРА-М, 2020. – 320с. – (Высшее образование). – URL: <https://znanium.com/catalog/product/1046281>. – Режим доступа: по подписке. ✓
2. Варфоломеева А.О. Информационные системы предприятия: учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. – 2-е изд перераб. и доп. – Москва: ИНФРА – М, 2019. – 330 с. – (Высшее образование. Бакалавриат). ISBN 978-5-16-012274-8. – Текст: электронный. ✓
– URL: <https://znanium.com/catalog/product/1002067>. – Режим доступа: по подписке.
3. Цифровой бизнес: учебник / под науч. ред. О.В. Китовой. – Москва: ИНФРА-М, 2021. – 418с. – (Высшее образование. Магистратура). ISBN 978-5-16-013017-0. – Текст электронный. – URL: <https://znanium.com/catalog/product/1659834/> - Режим доступа по подписке ✓

б) программное обеспечение и Интернет-ресурсы

1. Среды разработки, под С# (и другие языки, если это цель исследования):
 - a. Visual Studio – студенческая подписка, или аналоги
 - b. Visual Studio Code
 - c. Intelij Idea – студенческая подписка, или аналоги
 - d. Eclipse + PyDev (open-source Linux, Windows для Java, с дополнительными модулями – и для Python)
 - e. Unity – платформа
2. Курсы и инструкции по разработке в Unity. Learn game development Unity | Courses & tutorials in game design, VR, AR, & Real-time 3D | Unity Learn <https://learn.unity.com/> ,
3. Платформа Unity. Unity Real-Time Development Platform | 3D, 2D VR & AR Engine <https://unity.com/>
4. Паттерны программирования на примере С# в Unity <https://www.habrador.com/tutorials/programming-patterns/2-flyweight-pattern/?msclkid=c1d86df0b61a11ec9c9941b3ce3fc454>
5. Паттерны программирования на примере С# <https://dofactory.com/net/flyweight-design-pattern?msclkid=c1d84c0db61a11ec94fab8d53c74d4f7>
6. Паттерны программирования на примере Java https://www.tutorialspoint.com/design_pattern/flyweight_pattern.htm#:~:text=Flyweight%20pattern%20is%20primarily%20used%20to%20reduce%20the,count%20thus%20improving%20the%20object%20structure%20of%20application.&msclkid=c1d4f2e3b61a11ec866d8f5187f20919

7. Stream in Java <https://www.geeksforgeeks.org/stream-in-java/?msclkid=3fde5943b62d11ec878c0045e1f844c4>
8. LINQ in C# <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/?msclkid=9a7dae1fb62d11ec963ec1b652b96793>
9. Список API сервисов Yandex.Cloud <https://cloud.yandex.ru/docs/overview/api>
10. Сопоставление Yandex. Cloud с аналогичными платформами <https://cloud.yandex.ru/docs/overview/platform-comparison/>
11. Система управления проектами Mercurial <https://www.mercurial-scm.org/?msclkid=2a7ca731b64411ec9fb0019ed3c7da87>
12. Система управления проектами Shtab <https://shtab.app/>
13. Система контроля версий git <https://git-scm.com/?msclkid=0052a91db64511ecafbbbbead308a46b>
14. Что такое git? <https://docs.microsoft.com/en-us/devops/develop/git/what-is-git?msclkid=00544a15b64511ec89d1800b57a59ba6>
15. Gitea – возможности собственного git сервиса. Официальный сайт <https://gitea.io/en-us/?msclkid=092e0f6fb64611ec860020b17b1b4f40>
Официальная работающая сущность <https://gitea.dev/?msclkid=092e6a8fb64611ec80820407a14fc9ba>

9. Материально-техническое обеспечение дисциплины

Для проведения лекционных занятий необходима лекционная аудитория, снабженная компьютером и проекционной системой.

Для проведения лабораторных занятий необходима аудитория, снабженная компьютерами с операционной системой Windows (Linux).

Программа составлена в соответствии с требованиями ФГОС ВО с учетом Примерной ООП ВО по направлению 09.03.01 – Информатика и вычислительная техника и профилю подготовки «Вычислительные машины, комплексы, системы и сети»

Автор
ст. преподаватель

М.С. Портенко

Программа одобрена на заседании кафедры информатики и программирования от 13 сентября 2021 года, протокол №2