

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ

ОП.11 Объектно-ориентированное программирование TypeScript

программы подготовки специалистов среднего звена

09.02.07 Информационные технологии и программирование

Форма обучения: *очная*

Владивосток 2023

СОДЕРЖАНИЕ

- 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ
УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1 Место дисциплины в структуре основной образовательной программы

Учебная дисциплина (ОП.11 Объектно-ориентированное программирование TypeScript) является частью общепрофессионального учебного цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по специальности/профессии 09.02.07 Информационные технологии и программирование.

1.2 Цель и планируемые результаты освоения дисциплины

По итогам освоения дисциплины, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

Код компетенции	Умения	Знания
ОК 01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Основные способы решения задач
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	Основные средства поиска, анализа и интерпретации информации для выполнения задач профессиональной деятельности

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Объем образовательной программы учебной дисциплины	55
в том числе:	
– теоретическое обучение	22
– практические занятия	24
– лабораторные занятия (если предусмотрено)	
– курсовая работа (проект) (если предусмотрено)	
– самостоятельная работа	5
– консультации	2
– промежуточная аттестация в форме экзамена	2

2.2 Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, самостоятельная работа обучающихся	Объем часов	Активные и интерактивные формы занятий
1	2	3	
Раздел 1	Введение объектно-ориентированное программирование. Теоретические основы объектно-ориентированного программирования	16	
1.1 Тема: Введение. Основные понятия. Настройка среды	<i>Содержание учебного материала</i> Структура и содержание дисциплины, ее связь с другими дисциплинами специальности. Обзор рекомендуемой литературы. Парадигма программирования. Методика разработки программ. Объект -структура. Системы объектно-ориентированного программирования. Операторы.	4	Знание
1.2 Тема: Классификация подвидов ООП	<i>Содержание учебного материала</i> Классификация методов: конструкторы, деструкторы, селекторы и модификаторы. Классы объектов: назначение и семантика. Класс как абстракция совокупности объектов. Факторы, обусловившие появление и содержание концепции ООП. Основные принципы, объектные языки и системы. Три основные концепции ООП: наследие, инкапсуляция и полиморфизм.	4	Знание

<p>1.3 Тема: Сложности определения и концепции</p>	<p><i>Содержание учебного материала</i> Факторы, обусловившие появление и содержание концепции ООП. Основные принципы, объектные языки и системы. Три основные концепции ООП: наследие, инкапсуляция и полиморфизм.</p>	<p>2</p>	<p>Знание</p>
<p>1.4 Тема: Наследование и инкапсуляция в ООП</p>	<p><i>Содержание учебного материала</i> Наследование и суперклассы. Простое наследование: Множественное наследование. Свойство языка программирования, Методы и данные реализации</p>	<p>2</p>	<p>Знание</p>
	<p><i>Практическая работа</i> Использование языков программирования. Построение логически правильные и эффективные программы</p>	<p>2</p>	<p>Процесс</p>
<p>1.5 Тема: Агрегирование и композиция</p>	<p><i>Содержание учебного материала</i> Методика создания нового класса. Агрегация (агрегирование по ссылке). Композиция (агрегирование по значению).</p>	<p>2</p>	<p>Знание</p>
<p>Раздел 2</p>	<p>Основы объектно-ориентированного программирования на языке TypeScript.</p>	<p>14</p>	

<p>2.1 Тема: Переменные и типы в TypeScript</p>	<p><i>Содержание учебного материала</i> Переменные и типы в TypeScript. Типы и как их использовать для описания сигнатур функций и методов. Тип объединения. Тип пересечения. Приведение типов или защитники типов. Псевдонимы типов.</p>	<p>2</p>	<p>Знания</p>
<p>2.2 Тема: Классы, типы данных TypeScript со свойствами</p>	<p><i>Содержание учебного материала</i> Классы. Наследование. Модификаторы доступа. Понимание private. Понимание protected. Свойства параметров. Аксессуары (геттеры/сеттеры) Статические свойства. Абстрактные классы. Дополнительные методы.</p>	<p>2</p>	<p>Знание</p>
	<p><i>Практическая работа</i> Запуск TypeScript. Изучение интерфейса программы. Типы данных в TypeScript. Конструктор</p>	<p>2</p>	<p>Процесс</p>
	<p><i>Практическая работа</i> Изучение основных элементов управления среды разработчика. Разработка проекта.</p>	<p>4</p>	<p>Процесс</p>
	<p><i>Практическая работа</i> Изучение принципов разработки программы, использующей операторы, указатели и массивы. Проектирование и формирование</p>	<p>4</p>	<p>Процесс</p>

	формы		
Раздел 3	Особенности реализации	14	
3.1 Тема: Методы и данные в ООП	<i>Содержание учебного материала</i> Атрибуты.переменные, конкретные характеристики объекта. Методы — функции, описание объекта или класса. методы. Контроль оступа. Методы доступа.Свойства объекта.	2	Знание
3.2 Тема: Объектно-ориентированный дизайн	<i>Содержание учебного материала</i> Объектно-ориентированный анализ. Реструктуризация данных класса.Реализация методов, Осуществление контроля. Реализация ассоциаций.	2	Знание
3.3 Тема: Основные принципы дизайна классов (S.O.L.I.D.)	<i>Содержание учебного материала</i> Основные принципы дизайна классов (S.O.L.I.D.) Принцип единственной ответственности(SRP.) Принцип открытости/закрытости (ОСР). Принцип подстановки Барбары Лисков (LSP). Принцип разделения интерфейса (ISP). Принцип инверсии зависимостей (DIP	2	Процесс

<p>3.4 Тема: Паттерны проектирования</p>	<p><i>Практическая работа</i> Отличие паттернов проектирования от схожих инструментов. Основные виды паттернов и выбор подходящего паттерна проектирования</p>	<p>4</p>	<p>Процесс</p>
<p>3.5 Реализация интерфейсов</p>	<p><i>Содержание учебного материала</i> Интерфейсы классов, объектов. <i>Практическая работа</i> Расширение интерфейсов. Имплементация интерфейса. Интерфейсы массивов</p>	<p>4</p>	<p>Артефакт</p>

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрено наличие следующих специальных помещений:

Лаборатория: Программного обеспечения компьютерных сетей, программирования и баз данных. Рабочее место для преподавателя, оборудованное ПК - 1шт. Компьютеризированных посадочных мест с выходом в интернет – 25 штук ЖК панель 1 шт. Маркерная доска 1 шт. Лицензионное программное обеспечение общего и профессионального назначения

3.2 Информационное обеспечение реализации программы

Для реализации программы учебной дисциплины библиотечный фонд ВГУЭС укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Основная литература

1. Черный Борис – Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений — Санкт-Петербург: Питер, 2021. — 352 р.

Дополнительная литература

2. Файн Яков, Моисеев Антон TypeScript быстро — Санкт-Петербург: Питер, 2021. — 528 р.

Электронные ресурсы

1. Typescript: The starting point learning typescript - Режим доступа:
<https://www.typescriptlang.org/docs/>

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Методы оценки
<u>Перечень знаний, осваиваемых в рамках дисциплины:</u> <ul style="list-style-type: none">• основные концепции ООП;• принципы построения классов и объектов;• виртуальные методы и классы, абстрактные классы• принципы и виды наследования классов, шаблоны классов• обработку исключительных ситуаций, методику ООП и проектирования..	90-100 баллов «Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. 70-89 баллов «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. 50-69 баллов «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий	Оценка знания теоретическими сведениями дисциплины
<u>Перечень умений, осваиваемых в рамках дисциплины:</u> <ul style="list-style-type: none">• создавать объектно-ориентированные		Оценка выполнения X практических заданий

<p>программы с применением классов на языке TypeScript</p> <ul style="list-style-type: none"> • создавать иерархию классов • использовать полиморфизм, проектировать с учетом множественного наследования • создавать шаблоны функции и классов • использовать механизм обработки исключений • использовать библиотеку потоковых классов • производить отладку программного кода в среде разработчика 	<p><i>содержат ошибки.</i></p> <p><i>0-49 баллов «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</i></p>	
---	---	--

Для оценки достижения запланированных результатов обучения по дисциплине разработаны контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации, которые прилагаются к рабочей программе дисциплины.

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА
для проведения текущего контроля и промежуточной аттестации
по учебной дисциплине

ОП.11 Объектно-ориентированное программирование TypeScript

программы подготовки специалистов среднего звена
09.02.07 Информационные технологии и программирование

Форма обучения: *очная*

Владивосток 2023

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации по учебной дисциплине *ОП.11 «Объектно-ориентированное программирование TypeScript»* разработаны в соответствии с требованиями ФГОС СПО по специальности / профессии *09.02.07 Информационные технологии и программирование*, утвержденного приказом Минобрнауки России от *09.12.2016 N 1547 (ред. от 17.12.2020)* примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик(и): *Стриж Е.В., преподаватель*

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 4 от «22» мая 2023 г.

Председатель ЦМК _____ *И.О. Фамилия*


подпись

1 Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины *ОП.11 «Объектно-ориентированное программирование TypeScript»*.

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме экзамена (с использованием оценочного средства – *итоговая работа*)

2 Планируемые результаты обучения по дисциплине, обеспечивающие результаты освоения образовательной программы

Код ОК, ПК ¹	Код результата обучения ¹	Наименование результата обучения
ОК	01.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
	02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

3 Соответствие оценочных средств контролируемым результатам обучения

3.1 Средства, применяемые для оценки уровня практической подготовки

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель ² овладения результатами обучения	Наименование оценочного средства и представление его в КОС ³	
			Текущий контроль ⁴	Промежуточная аттестация ⁴
Раздел 1 Введение объектно-ориентированное программирование.				
Теоретические основы объектно-ориентированного программирования				
1.1 Тема: Введение. Основные понятия. Настройка среды	01.	Способность <i>рассказать, что такое ООП</i>	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
	01.	Способность рассказать, как настраивать среду		
1.2 Тема: Классификация подвидов ООП	01.	Способность классифицировать ООП	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
1.3 Тема: Сложности определения и концепции	01.	Способность разбираться в концепциях ООП	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
1.4 Тема: Наследование и инкапсуляция в ООП	01.	Знание наследования и инкапсуляции в ООП	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
1.5 Тема: Агрегирование и	01.	Знание агрегации и композиции, навыки работы с коллекциями объектов	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>

композиция				
Раздел 2 Основы объектно-ориентированного программирования на языке TypeScript.				
2.1 Тема: Переменные и типы в TypeScript	01.	Способность работать с переменными и типами в TypeScript	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
2.2 Тема: Классы, типы данных TypeScript со свойствами	01.	Способность работать с классами в TypeScript	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
Раздел 3 Особенности реализации				
3.1 Тема: Методы и данные в ООП	02.	Умение работать с методами и данными в ООП	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
3.2 Тема: Объектно-ориентированный дизайн	02.	Способность понимать структуру ООП задач	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
3.3 Тема: Основные принципы дизайна классов (S.O.L.I.D.)	02.	Знание основных принципов дизайна классов	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
3.4 Тема: Паттерны проектирования	02.	Понимание отличия паттернов проектирования от схожих инструментов, умение выбрать подходящий паттерн проектирования	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>
3.5 Тема: Реализация интерфейсов	02.	Умение работать с интерфейсами классов, объектов	<i>Практическая работа</i>	<i>Итоговая практическая работа</i>

4 Описание процедуры оценивания

Результаты обучения по дисциплине, уровень сформированности компетенций оцениваются по *по бальной системе*. Максимальная сумма баллов по дисциплине равна 100 баллам. При наборе до 49 баллов, студенту выставляется оценка «Неудовлетворительно», от 50 до 69 баллов - оценка «Удовлетворительно», от 70 до 89 баллов – оценка «Хорошо», от 90 до 100 баллов – оценка «Отлично».

Текущая аттестация по дисциплине проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

Критерии оценивания устного ответа

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания практических работ

Каждая практическая работа состоит из нескольких заданий, каждое из заданий оценивается в 2 балла. Выставляется 2 балла за полностью рабочее практическое задание и выполнение всех условий, описанных в задании.

Критерии выставления оценки студенту на экзамене

Итоговая оценка студента состоит из 100-бальной шкалы. При наборе до 49 баллов, студенту выставляется оценка «Неудовлетворительно», от 50 до 69 баллов - оценка «Удовлетворительно», от 70 до 89 баллов – оценка «Хорошо», от 90 до 100 баллов – оценка «Отлично».

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности,

	затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

6. Примеры оценочных средств для проведения промежуточной аттестации

6.1 Практические работы по дисциплине:

1. Напишите прототип программы

Напишите **прототип** программы **последовательно** (расширяя программу) добавляя необходимый функционал:

1. чтение с клавиатуры и вывод в терминал
 2. чтение из файла и вывод в терминал
 3. чтение с клавиатуры и отправка данных по сети
 4. чтение из файла и отправка данных по сети
 5. чтение с клавиатуры и запись в файл
 6. получение данных из сети и вывод в терминал
 7. получение данных из сети и запись в файл
- #### 2. Объявите класс Point3D (КТ: 2 балла)

Объявите класс Point3D для хранения координат трехмерного пространства.

Создайте три таких объекта, запишите в них различные координаты и выведите их поля в консоль.

3. Объявите класс Rect

Объявите класс Rect для представления прямоугольника, в котором хранятся две координаты: верхнего левого и правого нижнего углов.

Реализуйте возможность создать экземпляр класса без аргументов (в этом случае все координаты инициализируются значением 0) и с четырьмя аргументами (координата левого верхнего угла, ширина и высота)

Создайте несколько экземпляров с вызовом разных конструкторов и выводом значений полей в консоль.

4. Объявите класс Triangle

Объявите класс Triangle, хранящий три координаты вершин.

Координаты представьте в виде ссылок на класс Point, который рассмотрен на этом занятии.

Реализуйте возможность создать экземпляр класса без аргументов и с шестью аргументами (по два на каждую координату).

Создайте два объекта и выведите координаты вершин по каждому объекту в консоль.

5. Объявите класс Line

Объявите класс Line для представления линии на плоскости, хранящий две координаты: начало и конец линии.

Создайте два объекта этого класса и определите (напишите для этого функцию): пересекаются ли эти две линии.

6. Объявите класс Person

Объявите класс Person для описания сотрудника с полями: Ф.И.О., возраст, вес, номер разряда (целое число от 1 до 5).

Пропишите конструктор(ы), сеттер(ы) и геттер(ы) для записи значений по сотруднику и считывания данных.

Обеспечьте корректность представления данных:

- возраст и вес – положительные числа в допустимых пределах;
- разряд в диапазоне [1; 5];

- в Ф.И.О. могут использоваться только буквенные символы, пробел и дефис.

Создайте несколько таких объектов и убедитесь в их корректной работе.

7. Объявите классы для описания мебели

Объявите классы для описания мебели: стулья, шкафы, полки, столы.

У этих классов имеются общие поля: название, габариты, цена.

И уникальные для каждого объекта:

- для стула: число ножек, высота ножек, наличие спинки;
- для шкафов: материал ручек, число створок и шкафчиков;
- для полок: число сегментов и размер каждого сегмента;
- для столов: число ножек и площадь столешницы.

Подумайте, как описать эти объекты. Создайте их и выведите значение полей в консоль.

8. Требуется создать описание окна приложения для разных устройств

Требуется создать описание окна приложения для разных устройств: смартфонов, планшетов и настольных компьютеров (десктопов).

Окно имеет общие параметры: заголовок, шрифт, наличие/отсутствие рамки.

И уникальные для каждого устройства:

- для смартфонов: ничего (все берется из базового класса);
- для планшетов: положение и размер окна;
- для десктопов: положение и размер окна, возможность менять размеры, полноэкранный режим.

Подумайте, как описать эти классы. Создайте экземпляры классов для каждого устройства и выведите значение полей в консоль.

9. Объявите базовый класс Stationery

Объявите базовый класс Stationery для описания канцтоваров с полями:

- цена
- наличие на складе
- идентификатор

В этом классе реализуйте два конструктора: без аргументов и с тремя аргументами для инициализации его полей.

Затем, объявите производные от него классы для описания: ручек, карандашей, тетрадей.

В дочерних классах реализуйте несколько конструкторов с вызовом конструктора базового класса без аргументов и с аргументами.

Также в производных классах придумайте разные поля (разные у разных классов) и пропишите методы в соответствующих классах:

- displayPen() – для отображения данных по ручкам;
- displayPencil() – для отображения данных по карандашам;
- displayNotebook() – для отображения данных по тетрадям.

Создайте объекты дочерних классов и выведите информацию по ним в консоль.

10. Опишите базовый класс Graph

Опишите базовый класс Graph для представления графиков с полями:

- массив из N значений (значения графика),
- название

Опишите производные от него классы: LineGraph (для линейного графика), Chart (для круговой диаграммы), Bar (для столбикового графика).

В дочерних классах следует реализовать перегрузку метода draw() базового класса Graph для рисования графика в соответствующем виде (рисование – это вывод в консоль сообщения, что рисуется такой-то график с такими-то значениями).

Создайте несколько экземпляров дочерних классов со ссылками на них типа Graph. Через эти ссылки вызвать метод draw() и убедиться в работоспособности механизма динамической диспетчеризации (вызовов методов из дочерних классов).

11. Создайте базовый класс Nota

Создайте базовый класс Nota для описания музыкальных нот с полями:

- идентификатор
- название
- длительность
- наличие диеза
- наличие бемоля

Все поля должны быть закрыты и не наследоваться в дочерних классах. Работа с ними должна осуществляться через сеттеры и геттеры.

Добавьте определения дочерних классов для нот: до, ре, ми, фа, соль, ля, си.

Создайте их экземпляры и выведите информацию по ним в консоль.

12. Создайте класс User и его наследника класс SuperUser

Создайте класс User и его наследника класс SuperUser, которые описывают пользователя и супер-пользователя.

В классе User необходимо описать:

- конструктор, который принимает в качестве параметров значения для атрибутов name, login и password
- свойства (геттеры и сеттеры) для изменения и получения значений атрибутов
- метод showInfo, который печатает в произвольном формате значения атрибутов name и login
- статическое поле класса count для хранения количества созданных экземпляров класса User

Необходимые условия, которые надо учесть:

- атрибут name доступен и для чтения, и для изменения
- атрибут login доступен только для чтения
- атрибут password доступен только для изменения

В классе SuperUser необходимо описать:

- конструктор, который принимает в качестве параметров значения для атрибутов name, login, password и role
- свойство для изменения и получения значения атрибута role
- метод showInfo, который печатает в произвольном формате значения атрибутов name, login и role
- статическое поле класса count для хранения количества созданных экземпляров класса SuperUser

13. Опишите абстрактный класс Geometry

Опишите абстрактный класс Geometry для представления геометрических фигур с полями: width, color для определения толщины и цвета линии, а также с абстрактным методом draw() для рисования конкретного графического примитива.

Затем, опишите дочерние классы Line, Rect, Ellipse для представления линий, прямоугольников и эллипсов.

Определите в них поля для хранения координат этих фигур и метод draw() для их рисования.

Создайте обобщенные ссылки Geometry на объекты дочерних классов и вызовите у них метод draw().

14. Опишите абстрактный класс Recipes

Опишите абстрактный класс Recipes (рецепты)

с полями:

- название,
- тип // вегетарианский или обычный

и абстрактными методами:

- showIngredients // показать ингредиенты
- showRecipe // показать рецепт

Опишите несколько дочерних классов: Salad (для салатов), Pizza (для пицц), Porridge (для каш).

В каждом дочернем классе определите поле для списка ингредиентов (в виде строки) и описания самого рецепта (в виде строки). А также реализуйте абстрактные методы базового класса Recipes.

Создайте несколько экземпляров дочерних классов и через общий интерфейс (в виде ссылок типа Recipes) вызовите методы showRecipe и showIngredients.

15. Реализуйте интерфейс PersonInterface

Реализуйте интерфейс PersonInterface для единой работы с БД сотрудников. В этом интерфейсе объявите абстрактные методы:

- getInfo() – для получения общей информации о сотруднике;
- getStatus() – для получения информации о должности;
- getFIO() – для получения ФИО сотрудника.

Объявите дочерние классы: Supervisers (для руководителей), Jobs (для рядовых сотрудников), Clients (для клиентов).

В этих классах надо хранить информацию: ФИО, должность, год рождения, подразделение (если есть), телефон, адрес.

Реализуйте интерфейс PersonInterface с определением необходимых методов.

Создайте несколько экземпляров классов Supervisers, Jobs и Clients, используя обобщенные ссылки типа PersonInterface.

Вызовите для этих объектов методы интерфейса и убедитесь в их корректной работе.

16. Добавление класса Persons

Используя интерфейс и классы из предыдущего задания, добавьте к классам базовый класс Persons для хранения общих полей: ФИО, год рождения, адрес.

Кроме того, добавьте две статические переменные: count и count_clients для подсчета числа сотрудников (классы Supervisers и Jobs) и клиентов организации (класс Clients).

Создайте несколько объектов, используя ссылки обобщенного типа PersonInterface.

Выведите информацию по объектам, а также число сотрудников и клиентов.

17. Преобразуйте процедурный код в объектно-ориентированный

В приложенном файле plotter.ts код, который эмулирует работу плоттера.

Плоттер - это как принтер, только для черчения.

Виртуальный плоттер поддерживает пять команд:

1. Переместить каретку на некоторое расстояние в текущем направлении.
2. Повернуть на определенное количество градусов по часовой стрелке или против часовой стрелки.
3. Опустить или поднять каретку. Когда каретка опущена, плоттер при перемещении рисует линию.
4. Установить цвет линии (один из черного, красного или зелёного).
5. Установить начальную позицию каретки.

Код в рабочем состоянии (можете запустить его) и хорошо откомментирован.

Код написан в классическом процедурном стиле.

Ваша задача разобраться в коде и переписать его в объектно-ориентированном стиле.

Вы должны реализовать как минимум два класса:

- `Plotter` - основной класс
- `LogToConsole` - класс, который реализует интерфейс `Logger`, для логирования действий плоттера. `LogToConsole` должен выводить информацию в консоль.