**Вопрос 1** **Определение основной концепции ООП:**

Данные объекта не могут быть скрыты от остальной программы

В центре ООП находится понятие объекта

Сокрытие данных называется полиморфизмом.

Объект — это сущность, которой можно посылать сообщения и которая может на них реагировать

**Ответ:** В центре ООП находится понятие объекта   
Объект — это сущность, которой можно посылать сообщения и которая может на них реагировать

**Вопрос 2** **Определить основные области применения объектно ориентированного программирования:**

ООП не имеет смысла в использовании, так как его можно заменить структурным

Для задания определенного поведения объектам

Для создания своего типа данных

Для создания объектов по аналогии с объектами реального мира

**Ответ:** Для задания определенного поведения объектам   
Для создания своего типа данных   
Для создания объектов по аналогии с объектами реального мира

**Вопрос 3** **Дескрипторы это:**

Описание свойства

Модификаторы доступа к свойствам(скрытое или нет)

Свойства

Методы которые возвращают значения переменных

**Ответ:** Описание свойства

**Вопрос 4** **Выбрать правильные описания дескрипторов для некоторых свойств:**

writable - если установлено true, то свойство можно перезаписывать с помощью новых вызовов Object.defineProperty

writable - если установлен true, то значение свойства можно изменять

value — значение свойства

value - размер свойства в памяти

**Ответ:** writable - если установлен true, то значение свойства можно изменять   
value — значение свойства

**Вопрос 5** **Что означает данное выражение?**

**В классе задается поведение (функциональность) объекта.**

Класс будет иметь различное поведение в зависимости от заданного названия

Названия функций в классе будет определять поведение

Класс будет хранить поля или методы которые определяют поведение объекта

В классе будет храниться переменная(String) с описанием возможного поведения объекта

**Ответ:** Класс будет хранить поля или методы которые определяют поведение объекта

**Вопрос 6** **В чем достоинства ООП?**

Инкапсуляция информации защищает наиболее критичные данные от несанкционированного доступа

В организации информации при использовании данного подхода к программированию, разделения данных на «важное/неважное», «ключевое/подробности» и т.д.

Данные и операции вместе образуют определенную сущность и находятся в одной части программы

В возможности оперативно решить задачу, в которой отсутствует сложная иерархия

**Ответ:** Инкапсуляция информации защищает наиболее критичные данные от несанкционированного доступа   
В организации информации при использовании данного подхода к программированию, разделения данных на «важное/неважное», «ключевое/подробности» и т.д.   
Данные и операции вместе образуют определенную сущность и находятся в одной части программы

**Вопрос 7** **Возможно ли обращаться к методам чисел напрямую и что будет выведено:**

**alert( 12.34.toFixed(1) );**

Ни у переменной ни у числа такой метод вызвать нельзя

Да, метод округлит число до заданной точности

Нет, данный метод можно вызвать только у переменной которой присвоено число, у числа напрямую нельзя

Нет, toFixed свойство и должно вызываться без скобок()

**Ответ:** Да, метод округлит число до заданной точности

**Вопрос 8** **Что делает функция function?**

**function Animal(name) {**

**this.name = name;**

**this.canWalk = true;**

**}**

**var animal = new Animal("dog");**

Функция в данном случае не может использоваться как конструктор

Функция записывает данные в массив

Функция записывает переданные данные в переменные

Функция становится конструктором

**Ответ:** Функция записывает переданные данные в переменные   
Функция становится конструктором

**Вопрос 9** **Для какой из следующих задач нужно использовать дескрипторы?**

Для передачи свойства в функцию

Для сокрытия свойств

Для установления значения свойства

Для перечисления свойств в цикле

**Ответ:** Для установления значения свойства   
Для перечисления свойств в цикле

**Вопрос 10** **Получить доступ к изменению дескрипторов можно только используя функцию:**

Object.setProperty

Object.changeProperty

Object.defineProperty

Изменить дескрипторы нельзя

**Ответ:** Object.defineProperty

**Вопрос 11** **Указать верные утверждения о коде представленном ниже:**

**const fox = {};**

**Object.defineProperty(fox, 'name', {**

**value: 'Oliver',**

**writable: false,**

**configurable: true,**

**enumerable: true**

**});**

При изменении значения fox.name = 'John'; будет выведена ошибка доступа

Свойство можно будет перечислять в циклах

Свойство нельзя перечислять в цикле for .. in

Свойство можно будет перезаписать только через Object.defineProperty

При изменении значения fox.name = 'John' значение будет успешно изменено

**Ответ:** При изменении значения fox.name = 'John'; будет выведена ошибка доступа   
Свойство можно будет перечислять в циклах   
Свойство можно будет перезаписать только через Object.defineProperty

**Вопрос 12** **Как отработает данная часть скрипта:**

**alert( “hello.toUpperCase”);**

Будет выведено undefined

Код не отработает потому что упущены скобки

Будет выведено function toUpperCase() { ... }

Строка hello будет изменена в верхний регистр

**Ответ:** Строка hello будет изменена в верхний регистр

**Вопрос 13** **Все значения в JavaScript, за исключением null и undefined, содержат:**

Содержат только набор вспомогательных свойств

Содержат набор вспомогательных методов и свойств

Содержат только набор вспомогательных методов

Числовой индекс

**Ответ:** Содержат набор вспомогательных методов и свойств

**Вопрос 14** **Выбрать верные определения об основных принципах ООП:**

Абстракция - возможность одного интерфейса иметь различные реализация

Полиморфизм - выделение важных особенностей, характеристик объекта, главным образом отличающего его от других объектов

Полиморфизм - возможность одного интерфейса иметь различные реализация

Абстракция - выделение важных особенностей, характеристик объекта, главным образом отличающего его от других объектов

**Ответ:** Полиморфизм - возможность одного интерфейса иметь различные реализация   
Абстракция - выделение важных особенностей, характеристик объекта, главным образом отличающего его от других объектов

**Вопрос 15** **ООП возникло в результате развития идеологии:**

Логического программирования

Функционального программирования

Процедурного программирования

Императивного программирования

**Ответ:** Процедурного программирования

**Вопрос 16** **Выбрать верные определения об основных принципах ООП:**

Наследование - сокрытие реализации.

Инкапсуляция - сокрытие реализации.

Наследование возможность создавать объект на основании другого объекта.

Инкапсуляция - возможность создавать объект на основании другого объекта.

**Ответ:** Инкапсуляция - сокрытие реализации.   
Наследование возможность создавать объект на основании другого объекта.

**Вопрос 17** **Идентичны ли данные записи, и что делает данный код?**

**var MyTrack = new Lorry;**

**MyTrack["model"] = "6520"; и MyTrack.model = "6520";**

Правильно отработает только второй пример обращения через [] для записи в свойство

Оба примера являются неправильными

Оба примера идентично записывают в свойство объекта MyTrack значение 6520

Правильно отработает только первый пример обращения через точку для записи в свойство

**Ответ:** Оба примера являются неправильными

**Вопрос 18** **Имеют ли такие значения как null и undefined свойства или методы?**

Да

Да, но обратиться к ним напрямую нельзя

Нет, но их можно добавить

Нет

**Ответ:** Нет

**Вопрос 19** **Для чего нужны дескрипторы get и set?**

С их помощью можно запускать обозначенные функции при запросе к получению или записи свойства

Использование get и set не имеет смысла

Эти дескрипторы устанавливают область видимости для свойств

Они только предоставляют доступ к значениям свойств

**Ответ:** С их помощью можно запускать обозначенные функции при запросе к получению или записи свойства

**Вопрос 20** **Что будет выведено и чем является length?**

**alert( "Привет, мир!".length )**

12, length - метод

9, length - метод

9,length - свойство

12, length - свойство

**Ответ:** 12, length - свойство

**Вопрос 21** **Выбрать правильные описания дескрипторов для некоторых свойств:**

enumerable - если установлено true, то свойство можно перезаписывать с помощью новых вызовов Object.defineProperty

enumerable - если установлено true, то свойство будет перечисляться в цикле for .. in и при использовании метода Object.keys

configurable - если установлен true, то значение свойства можно изменять

configurable - если установлен true, то свойство можно перезаписывать с помощью новых вызовов Object.defineProperty

**Ответ:** enumerable - если установлено true, то свойство будет перечисляться в цикле for .. in и при использовании метода Object.keys   
configurable - если установлен true, то свойство можно перезаписывать с помощью новых вызовов Object.defineProperty

**Вопрос 22** **По умолчанию дескрипторы enumerable, configurable и writable установлены в значение:**

Null

true

false

undefined

**Ответ:** false

**Вопрос 23** **Выбрать основных участников идеи ООП, которые отсутствуют в процедурном подходе:**

Переменные

Функции

Классы

Объекты

Свойства классов

**Ответ:** Классы   
Объекты

**Вопрос 24** **Зачем нужен метод Object.defineProperties?**

Метод запрещает перезапись свойства

Такого метода не существует

Метод служит для определения значений нескольких свойств сразу

Метод является полным аналогом Object.defineProperty

**Ответ:** Метод служит для определения значений нескольких свойств сразу

**Вопрос 25** **Выберите правильные утверждения о:**

**hello.toUpperCase()**

Свойство переводит все символы в верхний регистр

Свойство заменяет первые буквы в каждом слове на большие

Метод переводит все символы в верхний регистр

toUpperCase - свойство и вызывать его как метод нельзя

**Ответ:** Метод переводит все символы в верхний регистр