КПИ. ИПСА 2018. ООП

Билет # 1

2. Что такое класс, что такое объект, в чем разница?

32. Что такое лямбда-выражение в Java.

24. Преобразование типов. Как перед приведением проверить реальный тип объекта?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 2

28. Composition-паттерны. Proxy, Facade, Decorator.

2. Что такое класс, что такое объект, в чем разница?

30. Override vs overload.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 3

3. Что такое свойства класса?

12. Cohesion vs coupling. Чем отличается, что лучше?

18. Модификатор final. Для чего используется?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 4

7. Что такое наследование, для чего мы его используем?

16. Mutable / Immutable объекты. Для чего используются?

29. Behavior-паттерны. Command, Observer, Iterator, Strategy.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 5

24. Преобразование типов. Как перед приведением проверить реальный тип объекта?

26. Что такое паттерн проектирования. Структура паттернов. Сколько всего паттернов?

1. Что такое ООП. Какие парадигмы знаете еще, чем ООП отличается от других, которые уже знаете и/или используете?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 6

5. Абстракция. Что это. Зачем?

20. Расширение / сужение области видимости.

18. Модификатор final. Для чего используется?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 7

12. Cohesion vs coupling. Чем отличается, что лучше?

18. Модификатор final. Для чего используется?

27. Builder-паттерны. Разница между Builder и Factory?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 8

8. Что такое полиморфизм, как он нам помогает?

23. Primary / secondary конструктор. Для чего используются. Преимущества?

24. Преобразование типов. Как перед приведением проверить реальный тип объекта?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 9

1. Что такое ООП. Какие парадигмы знаете еще, чем ООП отличается от других, которые уже знаете и/или используете?

31. Как «передать» поведение в виде параметра в метод/конструктор класса.

19. Модификатор static. Статическое поле. Статический метод. Нюансы доступа.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 10

23. Primary / secondary конструктор. Для чего используются. Преимущества?

28. Composition-паттерны. Proxy, Facade, Decorator.

9. Что такое декомпозиция, зачем мы ее делаем?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 11

21. Что такое конструктор класса. Конструктор по умолчанию. Класс без конструктора.

16. Mutable / Immutable объекты. Для чего используются?

14. Интерфейс. Что это такое, для чего используется?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 12

11. Что такое SOLID?

20. Расширение / сужение области видимости.

17. Модификаторы доступа: default / private / public / protected.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 13

27. Builder-паттерны. Разница между Builder и Factory?

11. Что такое SOLID?

22. Инстанциирование. Жизненный цикл объекта.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 14

12. Cohesion vs coupling. Чем отличается, что лучше?

13. Абстрактный класс, абстрактный метод, разница?

10. Область видимости поля / метода класса.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 15

5. Абстракция. Что это. Зачем?

8. Что такое полиморфизм, как он нам помогает?

6. Инкапсуляция. Когда используем?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 16

28. Composition-паттерны. Proxy, Facade, Decorator.

19. Модификатор static. Статическое поле. Статический метод. Нюансы доступа.

7. Что такое наследование, для чего мы его используем?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 17

15. Разница межу наследованием абстрактного класса и реализацией интерфейса.

32. Что такое лямбда-выражение в Java.

19. Модификатор static. Статическое поле. Статический метод. Нюансы доступа.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 18

3. Что такое свойства класса?

14. Интерфейс. Что это такое, для чего используется?

8. Что такое полиморфизм, как он нам помогает?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 19

29. Behavior-паттерны. Command, Observer, Iterator, Strategy.

22. Инстанциирование. Жизненный цикл объекта.

32. Что такое лямбда-выражение в Java.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 20

3. Что такое свойства класса?

17. Модификаторы доступа: default / private / public / protected.

25. Что такое множественное наследование. Как его реализовать если язык его не поддерживает?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 21

30. Override vs overload.

29. Behavior-паттерны. Command, Observer, Iterator, Strategy.

2. Что такое класс, что такое объект, в чем разница?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 22

6. Инкапсуляция. Когда используем?

25. Что такое множественное наследование. Как его реализовать если язык его не поддерживает?

11. Что такое SOLID?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 23

27. Builder-паттерны. Разница между Builder и Factory?

30. Override vs overload.

14. Интерфейс. Что это такое, для чего используется?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 24

9. Что такое декомпозиция, зачем мы ее делаем?

10. Область видимости поля / метода класса.

17. Модификаторы доступа: default / private / public / protected.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 25

20. Расширение / сужение области видимости.

26. Что такое паттерн проектирования. Структура паттернов. Сколько всего паттернов?

13. Абстрактный класс, абстрактный метод, разница?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 26

21. Что такое конструктор класса. Конструктор по умолчанию. Класс без конструктора.

4. Какие основные принципы / концепции ООП вы знаете?

15. Разница межу наследованием абстрактного класса и реализацией интерфейса.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 27

31. Как «передать» поведение в виде параметра в метод/конструктор класса.

31. Как «передать» поведение в виде параметра в метод/конструктор класса.

5. Абстракция. Что это. Зачем?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 28

23. Primary / secondary конструктор. Для чего используются. Преимущества?

4. Какие основные принципы / концепции ООП вы знаете?

15. Разница межу наследованием абстрактного класса и реализацией интерфейса.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 29

9. Что такое декомпозиция, зачем мы ее делаем?

4. Какие основные принципы / концепции ООП вы знаете?

16. Mutable / Immutable объекты. Для чего используются?

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 30

21. Что такое конструктор класса. Конструктор по умолчанию. Класс без конструктора.

25. Что такое множественное наследование. Как его реализовать если язык его не поддерживает?

10. Область видимости поля / метода класса.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.

КПИ. ИПСА 2018. ООП

Билет # 31

13. Абстрактный класс, абстрактный метод, разница?

26. Что такое паттерн проектирования. Структура паттернов. Сколько всего паттернов?

22. Инстанциирование. Жизненный цикл объекта.

• Ко всем вопросам необходимо привести примеры минимального кода, описывающего сущность вопроса.