

Практическое занятие № 12

Тема: Интерфейсы и потоки выполнения (Threads) в ЯП Java, Kotlin и JavaScript. Главный и дочерние потоки. Сравнение возможностей языков в области интерфейсов и потоков выполнения, достоинства и недостатки.

Литература:

1. Монахов В.В. Язык программирования Java. СПб.: БХВ-Петербург, 2011. 704 с.
2. Шилдт Г. Java: руководство для начинающих. М.: Вильямс, 2012. 624 с.
3. Гонсалвес Э. Изучаем Java EE 7. СПб.: Питер, 2014. 640 с.
4. Скин Дж., Гринхол Д. Kotlin. Программирование для профессионалов. СПб.: Питер, 2020. – 464 с.: (Серия «Для профессионалов»).
5. Хорстман К.С. Современный JavaScript для нетерпеливых. Addition Wesley. Перевод на русский. ДМК Пресс. Москва, 2021. 288 с.

Самостоятельный расширенный поиск по теме практической работы.

Вопросы

1. Проблемы множественного наследования классов в языках программирования Java, Kotlin и JavaScript. *Ромбовидное* наследование. Примеры и скриншоты.
2. Множественное наследование, с помощью интерфейсов в Java, Kotlin и JavaScript. Применение интерфейсов для написания полиморфного кода. Примеры и скриншоты.
3. Интерфейсы как разновидность абстрактных классов в Java, Kotlin и JavaScript. Вызов с помощью интерфейса продекларированных методов. Отличие интерфейсов от классов. Примеры их реализации, скриншоты в Java, Kotlin и JavaScript.
4. Проблемы наследования интерфейсов в Java, Kotlin и JavaScript. Совпадающие сигнатуры, как устранить проблему. Примеры работы с интерфейсами, скриншоты.
5. Композиция, как альтернатива множественному наследованию в языках ООП Java, Kotlin и JavaScript. Примеры реализации, и скриншоты результатов.
6. Достоинства и недостатки применения интерфейсов и композиций в Java, Kotlin и JavaScript. Примеры и скриншоты, их сравнение, выявление проблем.
7. *Потоки выполнения* в языках программирования Java, Kotlin и JavaScript. *Главный и дочерние* потоки. Класс *Thread*. Преимущества и недостатки. Примеры и скриншоты.
8. *Последовательная и параллельная* парадигмы выполнения программ в языках программирования Java, Kotlin и JavaScript. Примеры и скриншоты.
9. *Синхронизация потоков* по ресурсам и событиям в языках ООП Java, Kotlin и JavaScript. Оператор *Synchronize*, операции *Wait*, *Notify*. Примеры и скриншоты.
10. *Многопоточная среда* и способы её создания в языках Java, Kotlin и JavaScript. Примеры её реализации, скриншоты. Достоинства и недостатки.
11. *Синхронизация*, на базе интерфейса *Lock* в Java, Kotlin и JavaScript. Функции *Void Lock()*, *Void Unlock()* и *Boolean tryLock()*. Примеры синхронизации, скриншоты.
12. Достоинства и недостатки *синхронизации потоков* в Java, Kotlin и JavaScript. Сравнение скриншотов. Проблемы, преимущества и недостатки
13. Самостоятельное изучение самой свежей научной литературы (научных статей, монографий, учебников 2021 – 2025 годов издания) по теме занятия.