1-й вопрос билета

1 Змінні, типи даних, оператори та вирази.

1 Класси.

1 Успадкування.

1 Поліморфізм. Абстрактні класи і інтерфейси.

1 Вкладенні та внутрішні класи.

1 Архітектура Java Collections Framework. Основні особливості основних колекцій.

2-й вопрос билета

2 Потоки введення-виведення в Java.

2 Файлове введення-виведення в Java.

2 Багатопотоковість в Java.

2 Обробка подій в програмах з GUI

2 Розробка GUI-застосунків за допомогою бібліотеки SWING

2 Моделі життєвого циклу ПЗ. Водопадна та інкрементна моделі

# Теория

## 1 Переменные, типы данных, операторы и выражения

Объясните человеку, который впервые видит Джаву, основы программирования на ней. Но — не затрагивая ООП. Операторы управления выполнением программы (if, for etc) также можно пропустить. При этом все же следует упомянуть, что переменные могут быть локальными, статическими и нестатическими полями класса, параметрами функций; что типы данных существуют простые и ссылочные; что у операторов есть приоритеты и при их выполннии выполняется контроль и преобразование типов (в т.ч. ссылок на классы). Для полноты картины можно рассказать про элементарный консольный ввод-вывод, но это не требуется.

## 2 Классы

Все про разработку собственных классов и пользование ими. Но — не затрагивая наследование и все, что после него. Не забудьте про this, статические поля и методы, уровни доступа. Объясните, зачем это все вообще придумали (инкапсуляция).

## 3 Наследование

Все про наследование. Зачем его придумали, как им пользоваться, какова структура объекта класса-наследника, как происходит его конструирование, зачем нужно слово super. Упомяните про наследование интерфейса и реализации, композицию и делегирование; можно упомянуть про полиморфизм, но не подробно. Отдельно интересует вопрос “совместимости” ссылок на класс-предок и класс-потомок, а также какой из них “шире” с теоретико-множественной точки зрения.

## 4 Полиморфизм. Абстрактные классы и интерфейсы

Объясните, какая польза от полиморфизма. Какую проблему он решает? Приведите примеры. Не забудьте изложить проблему множественного наследования. Абстрактные классы и интерфейсы само собой.

## 5 Вложенные и внутренние классы

“Статические” и “нестатические”.  Локальные. Безымянные. Доступ к “внешним” сущностям (this’ам и локальным переменным). Типичные примеры применения.

## 6 Архитектура Java Collections Framework. Основные особенности основных коллекций

Проще всего здесь нарисовать диаграмму классов с интерфейсами и основными методами. Про каждый класс объяснить, что он умеет такого, чего не умеют другие. Не забудьте повторить, какова асимптотическая сложность различных операций, а также особенности работы с Tree/Hash/Map/Set.

## 7 Потоки ввода-вывода в Java

Байты, символы, кодировки, бинарные и текстовые данные, сериализация. Иерархия классов и основные методы в них. Простые иллюстративные примеры. Про конкретные классы для работы с файлами ничего рассказывать не надо.

## 8 Файловый ввод-вывод

Основные способы писать/читать текстовые и бинарные файлы (потоковый в/в и random access). Примеры кода!

## 9 Многопоточность в Java

Что такое многопоточность и зачем она нужна. Как она работает на уровне ОС и “железа”. Два способа создать поток. Какой из них лучше? Механизмы синхронизации и взаимного исключения (мониторы). Проблемы, которые они решают.

## 10 Обработка событий в программах с GUI

Модель делегирования событий, источники и слушатели. Основные классы событий. Регистрация слушателей, интерфейсы слушателей. Адаптеры.

## 11 Разработка GUI-приложений с помощью библиотеки SWING

Иерархия классов-компонентов SWING (и их “характерные” методы), построение иерархии компонентов на форме с помощью контейнеров, стандартные события и их обработка, layout’ы. Вопрос может показаться объемным, но это не так: здесь не надо перечислять все классы, все типы событий, все layout manager’ы и т.п., — достаточно выделить их основные группы и описать наиболее характерных их представителей.

## 12 Модели жизненного цикла ПО. Водопадная и инкрементная модели

Картинки и описание основных этапов. Область применения, достоинства и недостатки.

# Практика

Потренируйтесь, пожалуйста, писать **на бумаге** и **без подсказок** несложные программы, использующие следующее:

1. Несложные классы.
2. Массивы и основные стандартные коллекции, в которых хранятся объекты классов.
3. Работу с текстовыми и бинарными файлами, сериализацию.десериализацию.

3 \*Ви розробляєте програму для бібліотекаря. Програма допомагає йому вносити книги до каталогу і вести облік виданим книгам. При видачі книги вказується ПІБ читача. Книги можна повертати.

1. Розробіть клас, який зберігає в собі поточний стан всіх книг, а також дозволяє додати книгу до списку, видати читачеві і повернути, видалити по ID, подивитись список книг (Атрибути книги, ПІБ читача або “на полиці”). 15 балов

2. Додайте до п.1 методи збереження і загрузки всіх даних в/із текстового файла. В main’і продемонструйте, як ці методи протестувати. (5 балов)

\*Примітки:

1) Код методів get\*/set\* і “додаткових” конструкторів не пишіть!

2) Очікується два класси: один інкапсулює деяк сутність, другий — набір цих сутностей.

3) Класи із п.3.1 нічого не повинні робить з клавіатурою. Всі операції проводяться шляхом визову методів!