

```
class MersenneTwisterFast {
    serialVersionUID long
    N int
    M int
    MATRIX_A int
    LOWER_MASK int
    TEMPORING_MASK_B int
    TEMPORING_MASK_C int
    mt int[]
    mti int
    mag01 int[]
    _nextNextGaussian double
    _haveNextNextGaussian boolean
    MersenneTwisterFast()
    MersenneTwisterFast(long)
    MersenneTwisterFast(int[])
    clone() Object
    stateEquals(MersenneTwisterFast) boolean
    readState(DataInputStream) void
    writeState(DataOutputStream) void
    nextInt() int
    nextShort() short
    nextChar() char
    nextBoolean() boolean
    nextBoolean(double) boolean
    nextByte() byte
    nextBytes(byte[]) void
    nextLong() long
    nextLong(long) long
    nextDouble() double
    nextDouble(boolean, boolean) double
    clearGaussian() void
    nextGaussian() double
    nextFloat() float
    nextFloat(boolean, boolean) float
    nextInt(int) int
    nextInt(int, int) int
    nextDouble(double, double) double
    seed int[]
}
```

```
class CinemaTicket {
    CinemaTicket()
    CinemaTicket(Builder)
    movieName String
    price double
    seatLocation SeatLocation
    ticketId int
    cinemaId int
}

class Builder {
    build() T
}

class OnlineTicket {
    receiveSeatLocation(SeatLocation) void
    toString() String
    type OnlineTicketType
}

class StandardTicket {
    toString() String
}

class HotDealTicket {
    toString() String
}

class SurpriseTicket {
    toString() String
}
```

```
class Block {
    numberOfSeatsPerRow int
    Block(int)
    Block(Block, int)
    instantiateSeatSections(int) void
    hasSectionWithFreeSeats(SeatSection, int) boolean
    searchFreeSeatsFor(SeatSection, int) List<SeatLocation>
    checkSeatNumberArgument(int) void
    chooseNextFreeSection(int) List<SeatLocation>
    chooseRandomSeats(int) List<SeatLocation>
    getBlockId() int
    chooseBlock(int, Block) Block
    locateFreeSeatsIn(SeatSection, int) List<SeatLocation>
    getSectionSuccessor(SeatSection) SeatSection
    isFull(int) boolean
    full boolean
    95PercentFull boolean
    successor BlockLocation
    sections List<SeatSection>
}

class RightBlock {
    RightBlock()
    hasSectionWithFreeSeats(SeatSection, int) boolean
    chooseNextFreeSection(int) List<SeatLocation>
    isFull(int) boolean
}

class MiddleBlock {
    MiddleBlock(Block)
    hasSectionWithFreeSeats(SeatSection, int) boolean
    chooseNextFreeSection(int) List<SeatLocation>
    isFull(int) boolean
}

class LeftBlock {
    LeftBlock(Block)
    hasSectionWithFreeSeats(SeatSection, int) boolean
    chooseNextFreeSection(int) List<SeatLocation>
    isFull(int) boolean
}
```

```
class ClientVisitor {
    createTicketFor(RealClient) void
    createTicketFor(OnlineClient, OnlineTicketType) void
}

class AccessHandler {
    handleAccess() void
}

class OfficeCounter {
    ticketBuilder Builder
    clientCondition Predicate<ClientGroup>
    cinemaId int
    seatAdmissionObserver ICinemaObserver
    OfficeCounter(Block, Predicate<ClientGroup>, int)
    accept(ClientGroup) boolean
    offerSeats(ClientGroup) void
    createTicketFor(RealClient) void
    createTicketFor(OnlineClient, OnlineTicketType) void
    notifyCinemaFull() void
    equip(Builder) void
    equip(ICinemaObserver) void
    createTicket(SeatLocation, String, double) CinemaTicket
    terminal BoxOfficeTerminal
}

class OnlineAccessHandler {
    cinema Cinema
    onlineCounter OfficeCounter
    terminal BoxOfficeTerminal
    OnlineAccessHandler(Cinema)
    handleAccess() void
    chooseSeatsForType(ClientGroup, OnlineTicketType) void
    chooseSeatsStandard(ClientGroup) void
    chooseSeatsHotDeal(ClientGroup) void
    reserveAllSeats(ClientGroup, List<SeatLocations>) void
    reserveSeat(Client, SeatLocation) void
    reserveSeat(Client) void
    createOnlineVisitors() void
    randomAgeNearConfinement() int
    waitInterval(long) void
}

class RealAccessHandler {
    cinema Cinema
    ticketOffice BoxOffice
    RealAccessHandler(Cinema)
    handleAccess() void
    connect(OfficeCounter, ClientGroup) void
    createRealCinemaVisitors() ClientGroup
    randomAgeNearConfinement() int
    waitInterval(long) void
}
```

```
class Client {
    Client(String)
    Client(String, int)
    receiveTicket(CinemaTicket) void
    offerSeat(SeatLocation) void
    setOnline() void
    toString() String
    name String
    offeredSeatLocation SeatLocation
    onlineClient boolean
    ticket CinemaTicket
    age int
}

class OnlineClient {
    OnlineClient(String)
    OnlineClient(String, int)
    takeTicket(ClientVisitor, OnlineTicketType) void
    toString() String
}

class RealClient {
    RealClient(String)
    RealClient(String, int)
    takeTicket(ClientVisitor) void
    toString() String
}

class Cinema {
    screen Screen
    projector Projector
    Cinema()
    reportClientRejection(ClientGroup) void
    reportCinemaFull() void
    reportCinema95PercentFull() void
    closeTicketOffice() void
    isFull(Block, int) boolean
    is95PercentFull(Block) boolean
    startProjector() void
    rollOutScreen() void
    darkenRoom() void
    TicketOffice BoxOffice
    offerRejectingObserver ICinemaObserver
    full boolean
    leftBlock Block
    95PercentFull boolean
    seatAdmissionObserver ICinemaObserver
}
```

```
class ICinemaObserver {
    notifyCinemaFull() void
    notifyCinema95PercentFull() void
    notifyOfferRejection(ClientGroup) void
    wasTriggered() boolean
}

class OfferRejectionObserver {
    triggered boolean
    cinema Cinema
    OfferRejectionObserver(Cinema)
    notifyCinemaFull() void
    notifyCinema95PercentFull() void
    notifyOfferRejection(ClientGroup) void
    wasTriggered() boolean
    resetNumberOfRejectedOffers() void
    triggerUnsupportedOperationException(String) void
    numberOfRejectedOffers int
}

class SeatAdmissionObserver {
    cinema Cinema
    triggered boolean
    SeatAdmissionObserver(Cinema)
    notifyCinemaFull() void
    notifyCinema95PercentFull() void
    notifyOfferRejection(ClientGroup) void
    wasTriggered() boolean
    resetTriggered() void
    notifyOfferRejection(ClientGroup) void
}

class Configuration {
    instance
    seatNumberOuterBlocks int
    seatNumberInnerBlocks int
    maximumGroupSize int
    offerAcceptingProbability double
    iterationInterval long
    percentageOfFullCinema double
    numberOfOnlineBookings int
    mersenneTwister MersenneTwisterFast
    userDirectory String
    fileSeparator String
    lineSeparator String
    moduleName String
    logFilePath String
    buildLogFilePath() String
}

class ClientGroup {
    leftBlock Block
    ClientGroup()
    ClientGroup()
    generateRandomPreference() ClientSeatPreference
    acceptsOffer(int, Block) boolean
    leaveCinema() void
    getBlock(BlockLocation) Block
    addMember(Client) void
    checkMaximumMemberSize() boolean
    consistsOfOnlineClients() boolean
    toString() String
    preference ClientSeatPreference
    size int
    members List<Client>
}

class BoxOffice {
    BoxOffice(Block, int)
    guideToCounter(ClientGroup) void
    getTicketCounterFor(ClientGroup) ClientVisitor
    open() void
    close() void
    triggerIllegalStateException(ClientGroup) void
    registerSeatAdmissionObserver(ICinemaObserver) void
    normalCounter OfficeCounter
    onlineCounter OfficeCounter
    open boolean
    closed boolean
}

class Logger {
    instance
    dateFormat DateFormat
    writer BufferedWriter
    init() void
    write(String) void
    log(Object) void
    logError(String) void
    logLine() void
    newLine() void
    close() void
}
```

```
class ICinemaAccess {
    bookTicket() void
}

class ProxyAccess {
    onlineCounter OfficeCounter
    client OnlineClient
    type OnlineTicketType
    realAccess RealAccess
    ProxyAccess(OfficeCounter, OnlineClient, OnlineTicketType)
    bookTicket() void
}

class RealAccess {
    counter OfficeCounter
    client Client
    realAccess(OfficeCounter, Client)
    bookTicket() void
    onlineTicketType OnlineTicketType
}

class SeatLocation {
    reserve(Block, SeatSection, SeatRow, Seat) void
    toString() String
    locator String
    block Block
    section SeatSection
    row SeatRow
    seat Seat
}

class Seat {
    seatIdCounter int
    Seat()
    reserve() void
    free() void
    resetIdCounter() void
    seatId int
    reserved boolean
}

class BoxOfficeTerminal {
    BoxOfficeTerminal(Block)
    isPreferredSectionFree(ClientGroup) boolean
    chooseNextFreeSection(int) List<SeatLocation>
    fetchRandomSeats(int) List<SeatLocation>
    markOfferedSeats(ClientGroup) void
    leftBlock Block
}

class SeatRow {
    SeatRow(int, int)
    buildSeats(int, int) void
    checkNumberArgument(int, int) void
    rowId int
    seats List<Seat>
}

class SeatSection {
    SeatSection(String, int, int, int)
    buildRows(int, int, int) void
    orderedSeats List<Seat>
    sectionRows List<SeatRow>
    identifier String
}

class ClientSeatPreference {
    ClientSeatPreference(Block, SeatSection)
    toString() String
    block Block
    section SeatSection
}

class Application {
    init() void
    execute() void
    prepareShutdown() void
    main(String[]) void
}

class CinemaSimulator {
    realHandler AccessHandler
    onlineHandler AccessHandler
    startSimulation() void
}

class OnlineTicketType {
    standard
    hotDeal
    surprise
}

class BlockLocation {
    left
    middle
    right
}

class EnumUtility {
    randomEnumConstant(Class<E>) E
}

class Screen {
    displayMovie() void
}

class Projector {
    playFilm() void
}
```