## LXNP.400201.311 PP Q0bjec «Enumeration» BoardViewModel PawnType activePawn: BasePawnModel\* ActionButton blackPawns: QList<BasePawnModel\*> king ActionButton(QString) isEnPassantAvailable: bool buttonPressed(): void rook pawnViewModel: PawnViewMode hoverEnterEvent(QGraphicsSceneHoverEvent\*): void bishop whitePawns: QList<BasePawnModel\*> knight pawn hoverLeaveEvent(QGraphicsSceneHoverEvent\*): void whosTurn: PlayerType mousePressEvent(QGraphicsSceneMouseEvent\*): void winner: PlayerType\* setBackgroundColor(Qt::GlobalColor): void +type / BoardViewModel() didPromoteActivePawn(): bool didRemoveEnemyOnBoardPosition(BoardPosition): bool discardActivePawn(): void getActivePawn(): BasePawnModel\* -boardViewModel getBlackPawns(): QList<BasePawnModel\*> getBoardPositionForMousePosition(QPoint): BoardPosition getPawnOnBoardPosition(BoardPosition): BasePawnModel\* getWhitePawns(): QList<BasePawnModel\*> getWhosTurn(): PlayerType getWinner(): PlayerType\* initializePawns(): void initializePawnsForRow(int, PlayerType): void is King In Check (Player Type, bool, Board Position): boolsetActivePawnForField(PawnField\*): void setNewPositionForActivePawn(BoardPosition): void switchRound(): void validateAnotherPawnIntersection(BoardPosition, BasePawnModel\*, BoardPosition\*); bool validateKingsCheckForPawns(QList<BasePawnModel\*>, bool, BasePawnModel\*, BoardPosition): bool validatePawnMove(BoardPosition, BasePawnModel\*, BoardPosition\*): bool validatePawnPalcementForMousePosition(QPoint): bool -whosTurn -activePawn \ QGraphicsRectIte QGraphicsRectIter BasePawnModel BishopPawnModel PlayerType BoardField BoardFrameField didTakeFirstMove: bool BishopPawnModel(BoardPosition, PlayerType, PawnType, QString) BoardFrameField(QGraphicsItem\* defaultWidthHeight: int imagePath: QString validate Move (Board Position, Base Pawn Model\*, Board Position\*): boold a substitution of the properties of the propewhite owner: PlayerType position: BoardPosition BoardField(QColor, BoardPosition, QGraphicsItem type: PawnType getPosition(): BoardPosition BasePawnModel(BoardPosition, PlayerType, PawnType, QString) ~BasePawnModel() pawnWantsToMoveBvOneField(BoardPosition): bool validateDiagonalMove(BoardPosition, BasePawnModel\*): bool validate Move (BoardPosition, BasePawnModel\*, BoardPosition\*): boolvalidateVerticalOrHorizontalMove(BoardPosition, BasePawnModel\*); bool QGraphicsRectIte QObject Constants **BoardPosition** BoardView CongratulationsView defaultMargin: int defaultTextColor: QColor x: int checkWarningDescriptionTextItem: QGraphicsTextItem\* blackPlayerView: PlayerView\* check Warning Title Text Item: QGraphics Text Item \*+ CongratulationsView(PlayerType) board: BoardView\* fields: QList<BoardField\*> boardViewModel: BoardViewModel gameStarted: bool numberOfRowsColumns: int pawns: QList<PawnField\* scene: QGraphicsScene\* startXPosition: int whitePlayerView: PlayerView\* startYPosition: int displayMainMenu(): void BoardView() drawBoard(): void createFieldsColumn(int, int): void drawSettingsPanel(): void drawTitle(double, int): void draw(): void drawBoardFrame(): void drawUserPanel(): void drawBoardFrameAtPosition(QPoint, QRectF, QString): void drawViewForUser(PlayerType): PlayerView\* drawCheckWarningTextItems(): void ${\tt getCoordinatesForBoardPosition(BoardPosition): QPointF}$ handle Selecting Point For Active Pawn By Mouse (QPoint): void the property of the propertygetFields(): QList<BoardField\*> mouseMoveEvent(QMouseEvent\*): void getPawnAtBoardPosition(BoardPosition): PawnField\* mousePressEvent(QMouseEvent\*): void getPawnAtMousePosition(QPoint): PawnField\* moveActivePawnToSelectedPoint(QPoint): void initializePawnFields(QList<BasePawnModel\*>): void quitGame(): void moveActivePawnToMousePosition(QPoint, BasePawnModel\*): void releaseActivePawn(): void placeActivePawnAtBoardPosition(BasePawnModel\*, BoardPosition): void placeBoardFields(): void selectPawn(PawnField\*): void promotePawnAtBoardPosition(BoardPosition): void setCheckStateOnPlayerView(PlayerType, bool): void removePawnAtBoardPosition(BoardPosition): void showCongratulationsScreen(PlayerType): void setPawnMoveCheckWarning(bool): void startGame(): void QGraphicsRectite KnightPawnModel PawnPawnModel PawnViewModel PawnField KingPawnModel(BoardPosition, PlayerType, PawnType, QString) KnightPawnModel(BoardPosition, PlayerType, PawnType, QString) PawnPawnModel(BoardPosition, PlayerType, PawnType, QString) getImagePath(PawnType, PlayerType): QString validateMove(BoardPosition, BasePawnModel\*, BoardPosition\*): box image: QPixmap validateMove(BoardPosition, BasePawnModel\*, BoardPosition\*): boo validate Move (Board Position, Base Pawn Model \*, Board Position \*): board Position \*) : board PositiongetTypeForInitialPosition(BoardPosition): PawnTyp imageLabel: QLabel\* PawnViewModel() position: BoardPosition getPosition(): BoardPosition PawnField(BoardPosition, QString, QGraphicsIte setImage(QString): void setPosition(BoardPosition): void QGraphicsRectItem PlayerView + createTextItem(QString, int, QColor, QGraphicsItem\*): QGraphicsTextItem\* + setBackgroundColor(QColor, QAbstractGraphicsShapeItem\*): void QueenPawnModel(BoardPosition, PlayerType, PawnType, QString) + RookPawnModel(BoardPosition, PlayerType, PawnType, QString) checkTextItem: QGraphicsTextItem\* validateMove(BoardPosition, BasePawnModel\*, BoardPosition\*): bool validateMove(BoardPosition, BasePawnModel\*, BoardPosition\*): bool defaultWidthHeight: int -blackPlayerView PlayerView(QGraphicsItem\*) setActive(bool): void setIsInCheck(bool): void setPlayer(PlayerType): void ГУИР.400201.311 РР Лит. Масса Масштаб Изм Лист № документа Подпись Дата Разраб. Кузьмин Диаграмма классов Пров. Калютчик Лист Листов 1 ЭВМ, гр.250503