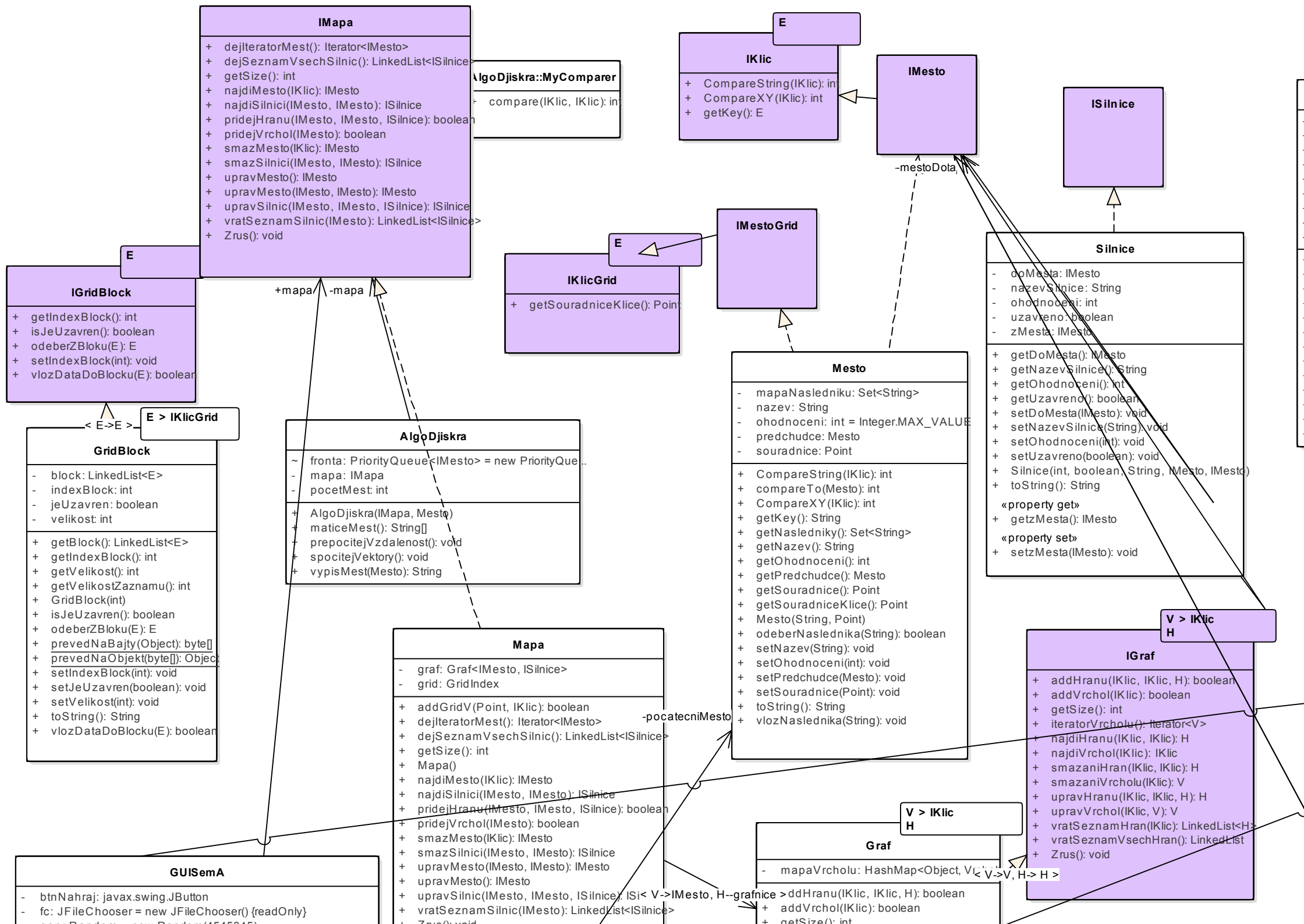
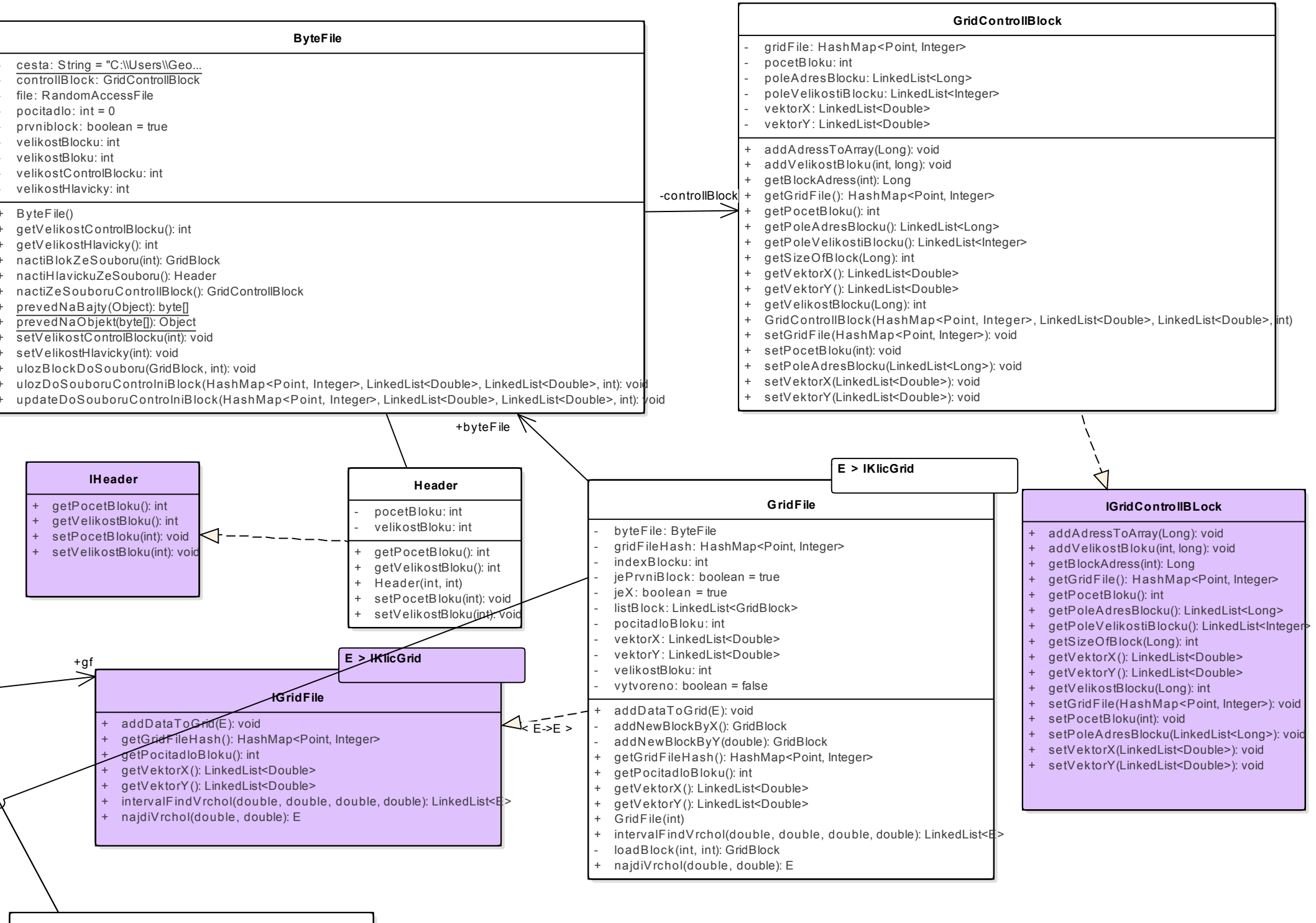
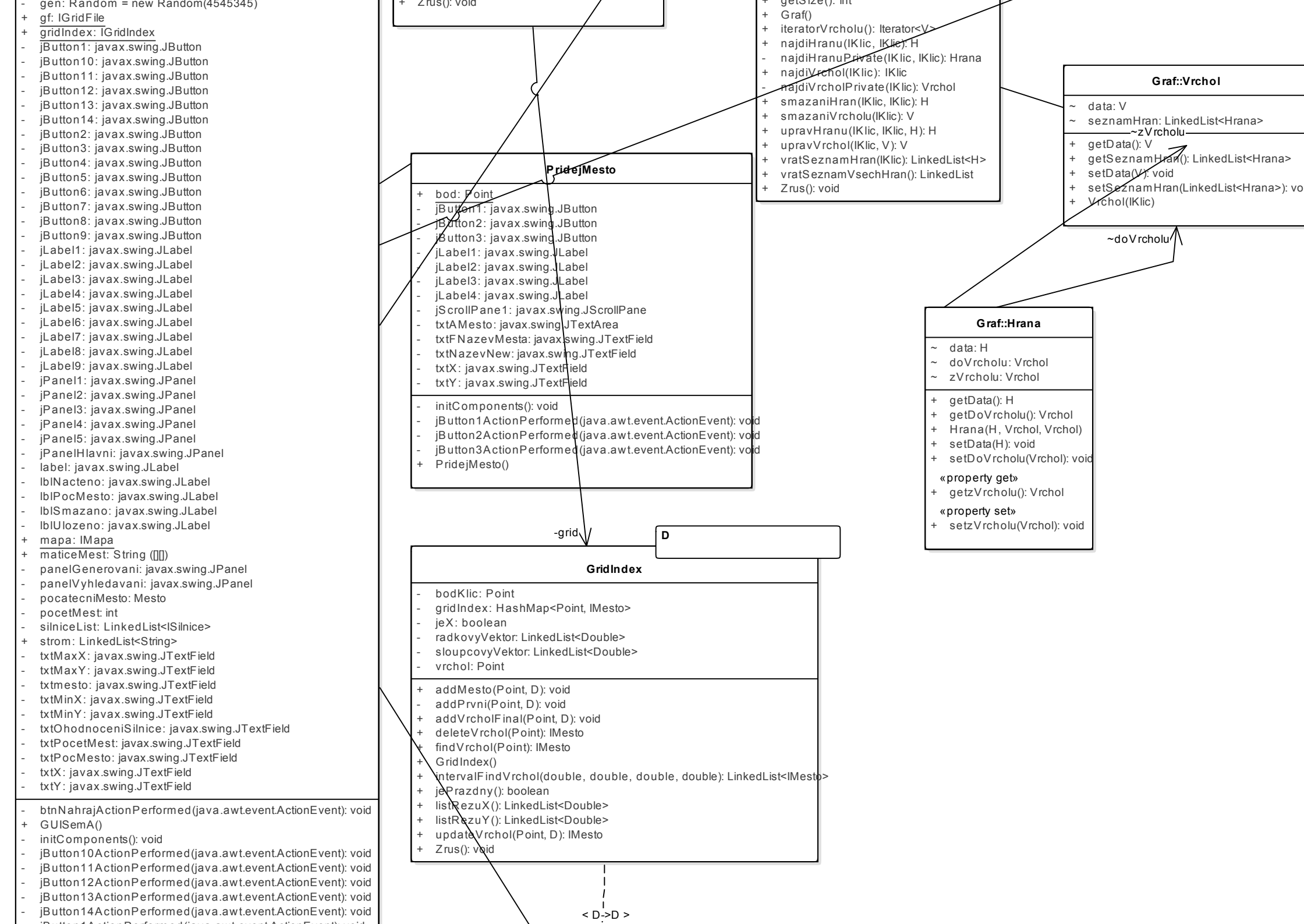
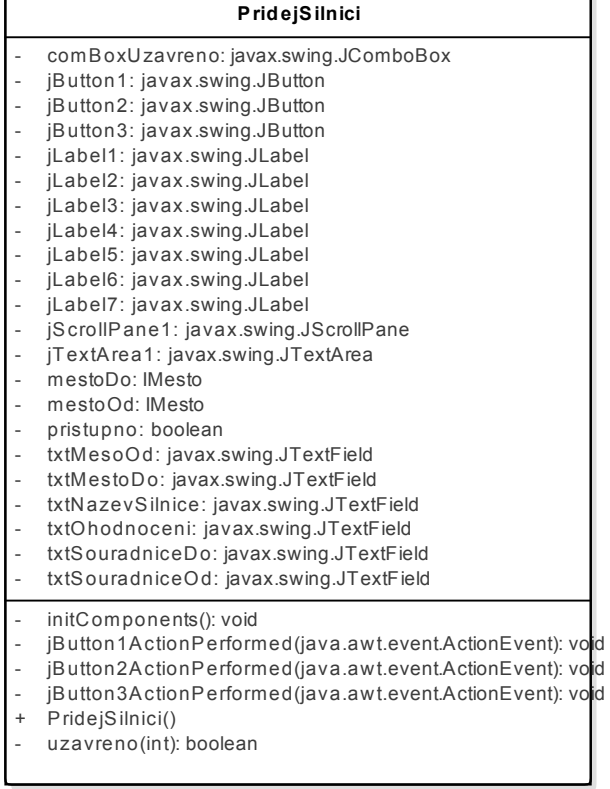
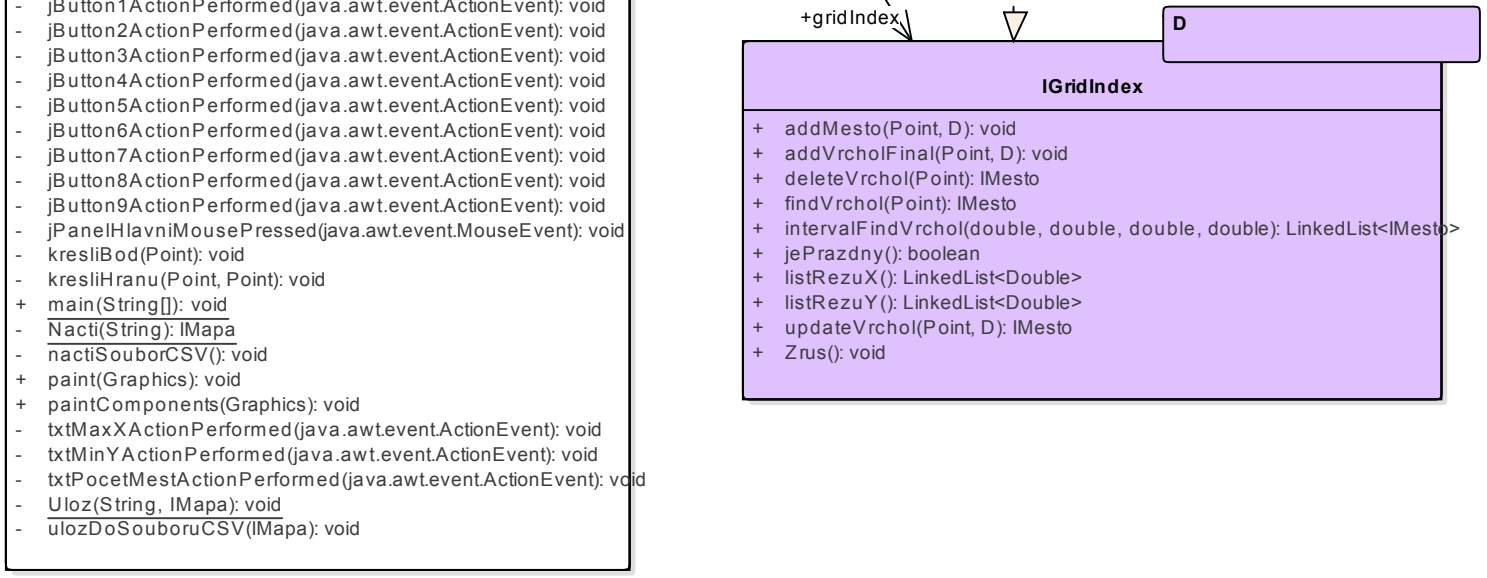

IMapa
+ dejIteratorMest(): Iterator<IMesto>
+ dejSeznamVsechSilnic(): LinkedList<ISilnice>
+ getSize(): int
+ najdiMesto(IKlic): IMesto
+ najdiSilnici(IMesto, IMesto): ISilnice
+ pridejHranu(IMesto, IMesto, ISilnice): boolean
+ pridejVrchol(IMesto): boolean
+ smazMesto(IKlic): IMesto
+ smazSilnici(IMesto, IMesto): ISilnice
+ upravMesto(): IMesto
+ upravMesto(IMesto, IMesto): IMesto
+ upravSilnic(IMesto, IMesto, ISilnice): ISilnice
+ vratSeznamSilnic(IMesto): LinkedList<ISilnice>
+ Zrus(): void
AlgoDjiskra::MyComparer
+ compare(IKlic, IKlic): int
E
IKlic
+ CompareString(IKlic): int
+ CompareXY(IKlic): int
+ getKey(): E
IMesto
ISilnice
-mestoDota
+mapa
-mapa
E
IGridBlock
+ getIndexBlock(): int
+ isJeUzavren(): boolean
+ odeberZBloku(E): E
+ setIndexBlock(int): void
+ vlozDataDoBlocku(E): boolean
E
IKlicGrid
+ getSouradniceKlice(): Point
IMestoGrid
Silnice
- doMesta: IMesto
- nazevSilnice: String
- ohodnoceni: int
- uzavreno: boolean
- zMesta: IMesto
+ getDoMesta(): IMesto
+ getNazevSilnice(): String
+ getOhodnoceni(): int
+ getUzavreno(): boolean
+ setDoMesta(IMesto): void
+ setNazevSilnice(String): void
+ setOhodnoceni(int): void
+ setUzavreno(boolean): void
+ Silnice(int, boolean, String, IMesto, IMesto)
+ toString(): String
«property get»
+ getzMesta(): IMesto
«property set»
+ setzMesta(IMesto): void
< E->E >
E > IKlicGrid
GridBlock
- block: LinkedList<E>
- indexBlock: int
- jeUzavren: boolean
- velikost: int
+ getBlock(): LinkedList<E>
+ getIndexBlock(): int
+ getVelikost(): int
+ getVelikostZaznamu(): int
+ GridBlock(int)
+ isJeUzavren(): boolean
+ odeberZBloku(E): E
+ prevedNaBajty(Object): byte[]
+ prevedNaObjekt(byte[]): Object
+ setIndexBlock(int): void
+ setJeUzavren(boolean): void
+ setVelikost(int): void
+ toString(): String
+ vlozDataDoBlocku(E): boolean
AlgoDjiskra
~ fronta: PriorityQueue<IMesto> = new PriorityQue...
- mapa: IMapa
- pocetMest: int
+ AlgoDjiskra(IMapa, Mesto)
+ maticeMest(): String[]
+ prepocitejVzdalenost(): void
+ spocitejVektory(): void
+ vypisMest(Mesto): String
Mesto
- mapaNaslednik: Set<String>
- nazev: String
- ohodnoceni: int = Integer.MAX_VALUE
- predchudce: Mesto
- souradnice: Point
+ CompareString(IKlic): int
+ compareTo(Mesto): int
+ CompareXY(IKlic): int
+ getKey(): String
+ getNasledniky(): Set<String>
+ getNazev(): String
+ getOhodnoceni(): int
+ getPredchudce(): Mesto
+ getSouradnice(): Point
+ getSouradniceKlice(): Point
+ Mesto(String, Point)
+ odeberNaslednika(String): boolean
+ setNazev(String): void
+ setOhodnoceni(int): void
+ setPredchudce(Mesto): void
+ setSouradnice(Point): void
+ toString(): String
+ vlozNaslednika(String): void
V > IKlic
H
IGraf
+ addHranu(IKlic, IKlic, H): boolean
+ addVrchol(IKlic): boolean
+ getSize(): int
+ iteratorVrcholu(): Iterator<V>
+ najdiHranu(IKlic, IKlic): H
+ najdiVrchol(IKlic): IKlic
+ smazaniHran(IKlic, IKlic): H
+ smazaniVrcholu(IKlic): V
+ upravHranu(IKlic, IKlic, H): H
+ upravVrchol(IKlic, V): V
+ vratSeznamHran(IKlic): LinkedList<H>
+ vratSeznamVsechHran(): LinkedList
+ Zrus(): void
-pocatecniMesto
Mapa
- graf: Graf<IMesto, ISilnice>
- grid: GridIndex
+ addGridV(Point, IKlic): boolean
+ dejIteratorMest(): Iterator<IMesto>
+ dejSeznamVsechSilnic(): LinkedList<ISilnice>
+ getSize(): int
+ Mapa()
+ najdiMesto(IKlic): IMesto
+ najdiSilnici(IMesto, IMesto): ISilnice
+ pridejHranu(IMesto, IMesto, ISilnice): boolean
+ pridejVrchol(IMesto): boolean
+ smazMesto(IKlic): IMesto
+ smazSilnici(IMesto, IMesto): ISilnice
+ upravMesto(IMesto, IMesto): IMesto
+ upravMesto(): IMesto
+ upravSilnic(IMesto, IMesto, ISilnice): ISi
+ vratSeznamSilnic(IMesto): LinkedList<ISilnice>
+ Zrus(): void
< V->IMesto, H--grafnice >
V > IKlic
H
Graf
- mapaVrcholu: HashMap<Object, V...
< V->V, H-> H >
+ addHranu(IKlic, IKlic, H): boolean
+ addVrchol(IKlic): boolean
+ getSize(): int
GUISemA
- btnNahraj: javax.swing.JButton
- fc: JFileChooser = new JFileChooser() {readOnly}


ByteFile
- cesta: String = "C:\\Users\\Geo...
- controllBlock: GridControllBlock
- file: RandomAccessFile
- pocitadlo: int = 0
- prvniblock: boolean = true
- velikostBlocku: int
- velikostBloku: int
- velikostControlBlocku: int
- velikostHlavicky: int
+ ByteFile()
+ getVelikostControlBlocku(): int
+ getVelikostHlavicky(): int
+ nactiBlokZeSouboru(int): GridBlock
+ nactiHlavickuZeSouboru(): Header
+ nactiZeSouboruControllBlock(): GridControllBlock
+ prevedNaBajty(Object): byte[]
+ prevedNaObjekt(byte[]): Object
+ setVelikostControlBlocku(int): void
+ setVelikostHlavicky(int): void
+ ulozBlockDoSouboru(GridBlock, int): void
+ ulozDoSouboruControlniBlock(HashMap<Point, Integer>, LinkedList<Double>, LinkedList<Double>, int): void
+ updateDoSouboruControlniBlock(HashMap<Point, Integer>, LinkedList<Double>, LinkedList<Double>, int): void
GridControllBlock
- gridFile: HashMap<Point, Integer>
- pocetBloku: int
- poleAdresBlocku: LinkedList<Long>
- poleVelikostiBlocku: LinkedList<Integer>
- vektorX: LinkedList<Double>
- vektorY: LinkedList<Double>
+ addAdressToArray(Long): void
+ addVelikostBloku(int, long): void
+ getBlockAdress(int): Long
+ getGridFile(): HashMap<Point, Integer>
+ getPocetBloku(): int
+ getPoleAdresBlocku(): LinkedList<Long>
+ getPoleVelikostiBlocku(): LinkedList<Integer>
+ getSizeOfBlock(Long): int
+ getVektorX(): LinkedList<Double>
+ getVektorY(): LinkedList<Double>
+ getVelikostBlocku(Long): int
+ GridControllBlock(HashMap<Point, Integer>, LinkedList<Double>, LinkedList<Double>, int)
+ setGridFile(HashMap<Point, Integer>): void
+ setPocetBloku(int): void
+ setPoleAdresBlocku(LinkedList<Long>): void
+ setVektorX(LinkedList<Double>): void
+ setVektorY(LinkedList<Double>): void
-controllBlock
+byteFile
IHeader
+ getPocetBloku(): int
+ getVelikostBloku(): int
+ setPocetBloku(int): void
+ setVelikostBloku(int): void
Header
- pocetBloku: int
- velikostBloku: int
+ getPocetBloku(): int
+ getVelikostBloku(): int
+ Header(int, int)
+ setPocetBloku(int): void
+ setVelikostBloku(int): void
E > IKlicGrid
GridFile
- byteFile: ByteFile
- gridFileHash: HashMap<Point, Integer>
- indexBlocku: int
- jePrvniBlock: boolean = true
- jeX: boolean = true
- listBlock: LinkedList<GridBlock>
- pocitadloBloku: int
- vektorX: LinkedList<Double>
- vektorY: LinkedList<Double>
- velikostBloku: int
- vytvoreno: boolean = false
+ addDataToGrid(E): void
- addNewBlockByX(): GridBlock
- addNewBlockByY(double): GridBlock
+ getGridFileHash(): HashMap<Point, Integer>
+ getPocitadloBloku(): int
+ getVektorX(): LinkedList<Double>
+ getVektorY(): LinkedList<Double>
+ GridFile(int)
+ intervalFindVrchol(double, double, double, double): LinkedList<E>
- loadBlock(int, int): GridBlock
+ najdiVrchol(double, double): E
IGridControllBLock
+ addAdressToArray(Long): void
+ addVelikostBloku(int, long): void
+ getBlockAdress(int): Long
+ getGridFile(): HashMap<Point, Integer>
+ getPocetBloku(): int
+ getPoleAdresBlocku(): LinkedList<Long>
+ getPoleVelikostiBlocku(): LinkedList<Integer>
+ getSizeOfBlock(Long): int
+ getVektorX(): LinkedList<Double>
+ getVektorY(): LinkedList<Double>
+ getVelikostBlocku(Long): int
+ setGridFile(HashMap<Point, Integer>): void
+ setPocetBloku(int): void
+ setPoleAdresBlocku(LinkedList<Long>): void
+ setVektorX(LinkedList<Double>): void
+ setVektorY(LinkedList<Double>): void
+gf
E > IKlicGrid
IGridFile
+ addDataToGrid(E): void
+ getGridFileHash(): HashMap<Point, Integer>
+ getPocitadloBloku(): int
+ getVektorX(): LinkedList<Double>
+ getVektorY(): LinkedList<Double>
+ intervalFindVrchol(double, double, double, double): LinkedList<E>
+ najdiVrchol(double, double): E
E->E >


- gen: Random = new Random(4545345)
+ gf: IGridFile
+ gridIndex: IGridIndex
- jButton1: javax.swing.JButton
- jButton10: javax.swing.JButton
- jButton11: javax.swing.JButton
- jButton12: javax.swing.JButton
- jButton13: javax.swing.JButton
- jButton14: javax.swing.JButton
- jButton2: javax.swing.JButton
- jButton3: javax.swing.JButton
- jButton4: javax.swing.JButton
- jButton5: javax.swing.JButton
- jButton6: javax.swing.JButton
- jButton7: javax.swing.JButton
- jButton8: javax.swing.JButton
- jButton9: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jLabel5: javax.swing.JLabel
- jLabel6: javax.swing.JLabel
- jLabel7: javax.swing.JLabel
- jLabel8: javax.swing.JLabel
- jLabel9: javax.swing.JLabel
- jPanel1: javax.swing.JPanel
- jPanel2: javax.swing.JPanel
- jPanel3: javax.swing.JPanel
- jPanel4: javax.swing.JPanel
- jPanel5: javax.swing.JPanel
- jPanelHlavni: javax.swing.JPanel
- label: javax.swing.JLabel
- lblNacteno: javax.swing.JLabel
- lblPocMesto: javax.swing.JLabel
- lblSmazano: javax.swing.JLabel
- lblUlozeno: javax.swing.JLabel
+ mapa: IMapa
+ maticeMest: String ([][])
- panelGenerovani: javax.swing.JPanel
- panelVyhledavani: javax.swing.JPanel
- pocatecniMesto: Mesto
- pocetMest: int
- silniceList: LinkedList<ISilnice>
+ strom: LinkedList<String>
- txtMaxX: javax.swing.JTextField
- txtMaxY: javax.swing.JTextField
- txtmesto: javax.swing.JTextField
- txtMinX: javax.swing.JTextField
- txtMinY: javax.swing.JTextField
- txtOhodnoceniSilnice: javax.swing.JTextField
- txtPocetMest: javax.swing.JTextField
- txtPocMesto: javax.swing.JTextField
- txtX: javax.swing.JTextField
- txtY: javax.swing.JTextField
- btnNahrajActionPerformed(java.awt.event.ActionEvent): void
+ GUISemA()
- initComponents(): void
- jButton10ActionPerformed(java.awt.event.ActionEvent): void
- jButton11ActionPerformed(java.awt.event.ActionEvent): void
- jButton12ActionPerformed(java.awt.event.ActionEvent): void
- jButton13ActionPerformed(java.awt.event.ActionEvent): void
- jButton14ActionPerformed(java.awt.event.ActionEvent): void
+ Zrus(): void
+ getSize(): int
+ Graf()
+ iteratorVrcholu(): Iterator<V>
+ najdiHranu(IKlic, IKlic): H
- najdiHranuPrivate(IKlic, IKlic): Hrana
+ najdiVrchol(IKlic): IKlic
- najdiVrcholPrivate(IKlic): Vrchol
+ smazaniHran(IKlic, IKlic): H
+ smazaniVrcholu(IKlic): V
+ upravHranu(IKlic, IKlic, H): H
+ upravVrchol(IKlic, V): V
+ vratSeznamHran(IKlic): LinkedList<H>
+ vratSeznamVsechHran(): LinkedList
+ Zrus(): void
Graf::Vrchol
~ data: V
~ seznamHran: LinkedList<Hrana>
~zVrcholu
+ getData(): V
+ getSeznamHran(): LinkedList<Hrana>
+ setData(V): void
+ setSeznamHran(LinkedList<Hrana>): vo
+ Vrchol(IKlic)
~doVrcholu
PridejMesto
+ bod: Point
- jButton1: javax.swing.JButton
- jButton2: javax.swing.JButton
- jButton3: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jScrollPane1: javax.swing.JScrollPane
- txtAMesto: javax.swing.JTextArea
- txtFNazevMesta: javax.swing.JTextField
- txtNazevNew: javax.swing.JTextField
- txtX: javax.swing.JTextField
- txtY: javax.swing.JTextField
- initComponents(): void
- jButton1ActionPerformed(java.awt.event.ActionEvent): void
- jButton2ActionPerformed(java.awt.event.ActionEvent): void
- jButton3ActionPerformed(java.awt.event.ActionEvent): void
+ PridejMesto()
Graf::Hrana
~ data: H
~ doVrcholu: Vrchol
~ zVrcholu: Vrchol
+ getData(): H
+ getDoVrcholu(): Vrchol
+ Hrana(H, Vrchol, Vrchol)
+ setData(H): void
+ setDoVrcholu(Vrchol): void
«property get»
+ getzVrcholu(): Vrchol
«property set»
+ setzVrcholu(Vrchol): void
-grid
D
GridIndex
- bodKlic: Point
- gridIndex: HashMap<Point, IMesto>
- jeX: boolean
- radkovyVektor: LinkedList<Double>
- sloupcovyVektor: LinkedList<Double>
- vrchol: Point
+ addMesto(Point, D): void
- addPrvni(Point, D): void
+ addVrcholFinal(Point, D): void
+ deleteVrchol(Point): IMesto
+ findVrchol(Point): IMesto
+ GridIndex()
+ intervalFindVrchol(double, double, double, double): LinkedList<IMesto>
+ jePrazdny(): boolean
+ listRezuX(): LinkedList<Double>
+ listRezuY(): LinkedList<Double>
+ updateVrchol(Point, D): IMesto
+ Zrus(): void
< D->D >


PridejSilnici
- comBoxUzavreno: javax.swing.JComboBox
- jButton1: javax.swing.JButton
- jButton2: javax.swing.JButton
- jButton3: javax.swing.JButton
- jLabel1: javax.swing.JLabel
- jLabel2: javax.swing.JLabel
- jLabel3: javax.swing.JLabel
- jLabel4: javax.swing.JLabel
- jLabel5: javax.swing.JLabel
- jLabel6: javax.swing.JLabel
- jLabel7: javax.swing.JLabel
- jScrollPane1: javax.swing.JScrollPane
- jTextArea1: javax.swing.JTextArea
- mestoDo: IMesto
- mestoOd: IMesto
- pristupno: boolean
- txtMesoOd: javax.swing.JTextField
- txtMestoDo: javax.swing.JTextField
- txtNazevSilnice: javax.swing.JTextField
- txtOhodnoceni: javax.swing.JTextField
- txtSouradniceDo: javax.swing.JTextField
- txtSouradniceOd: javax.swing.JTextField
- initComponents(): void
- jButton1ActionPerformed(java.awt.event.ActionEvent): void
- jButton2ActionPerformed(java.awt.event.ActionEvent): void
- jButton3ActionPerformed(java.awt.event.ActionEvent): void
+ PridejSilnici()
- uzavreno(int): boolean


- jButton1ActionPerformed(java.awt.event.ActionEvent): void
- jButton2ActionPerformed(java.awt.event.ActionEvent): void
- jButton3ActionPerformed(java.awt.event.ActionEvent): void
- jButton4ActionPerformed(java.awt.event.ActionEvent): void
- jButton5ActionPerformed(java.awt.event.ActionEvent): void
- jButton6ActionPerformed(java.awt.event.ActionEvent): void
- jButton7ActionPerformed(java.awt.event.ActionEvent): void
- jButton8ActionPerformed(java.awt.event.ActionEvent): void
- jButton9ActionPerformed(java.awt.event.ActionEvent): void
- jPanelHlavniMousePressed(java.awt.event.MouseEvent): void
- kresliBod(Point): void
- kresliHranu(Point, Point): void
+ main(String[]): void
- Nacti(String): IMapa
- nactiSouborCSV(): void
+ paint(Graphics): void
+ paintComponents(Graphics): void
- txtMaxXActionPerformed(java.awt.event.ActionEvent): void
- txtMinYActionPerformed(java.awt.event.ActionEvent): void
- txtPocetMestActionPerformed(java.awt.event.ActionEvent): void
- Uloz(String, IMapa): void
- ulozDoSouboruCSV(IMapa): void
+gridIndex
D
IGridIndex
+ addMesto(Point, D): void
+ addVrcholFinal(Point, D): void
+ deleteVrchol(Point): IMesto
+ findVrchol(Point): IMesto
+ intervalFindVrchol(double, double, double, double): LinkedList<IMesto>
+ jePrazdny(): boolean
+ listRezuX(): LinkedList<Double>
+ listRezuY(): LinkedList<Double>
+ updateVrchol(Point, D): IMesto
+ Zrus(): void