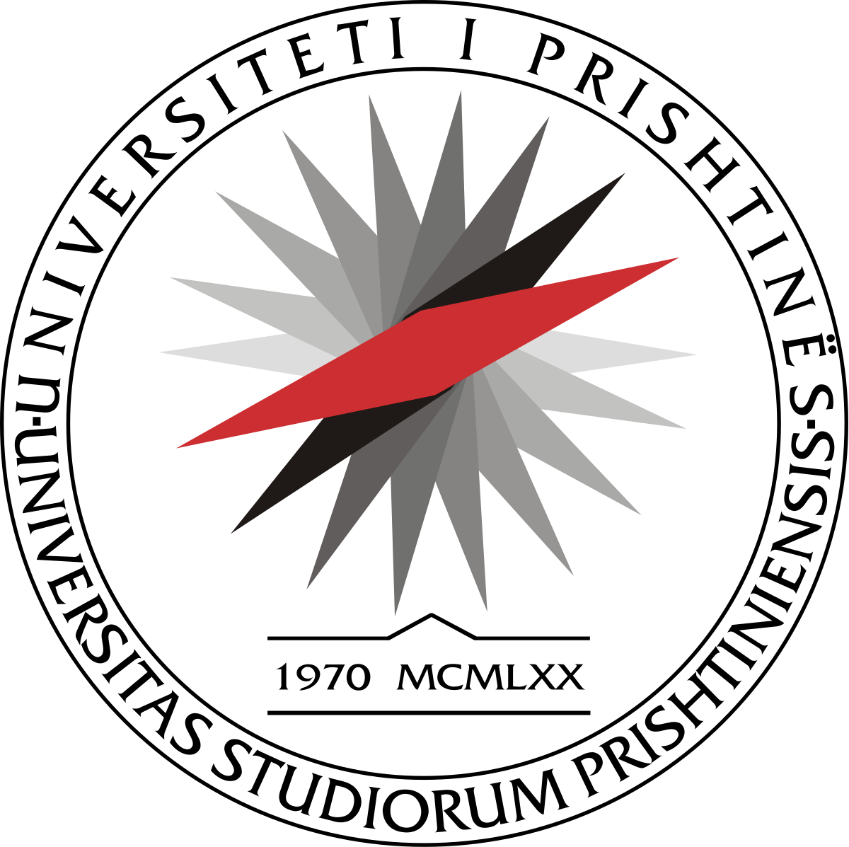
UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”

FAKULTETI I SHKENCAVE MATEMATIKE-NATYRORE

Departamenti: Matematikë | Programi: Shkencë kompjuterike



Detyra 4

Lënda: Siguria e të dhënave

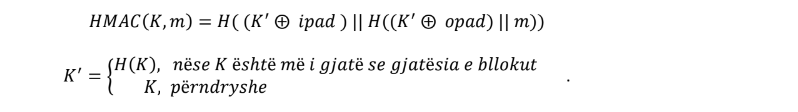
Punoi: Alketa Hajdari dhe Elda Meshi

Asistent: Besnik Duriqi

# **Paraqitja e problemit**

Implementoni në Java përkuzimin e dhënë të Hash Message Authentication Code (HMAC) i

cili mund të quhet HMAC\_Simplified\_SHA-1 si version i thjeshtuar HMAC\_SHA-1. (2 pikë)

. 

Ku:

– është Simplified SHA-1

– çelësi sekret

– mesazhi

– gjatësia e bllokut në Byte te (Simplified SHA-1)

– vlera 0x36 përsëritet B herë

– vlera 0x5C përsëritet B here

# **Ideja**

Ideja prapa HMAC (Kodi i Autentifikimit të Mesazhit Hash) është të sigurojë një mënyrë të sigurt për të verifikuar integritetin dhe vërtetësinë e një mesazhi duke përdorur një funksion hash kriptografik dhe një çelës sekret.

HMAC\_Simplified\_SHA-1 është një version i thjeshtuar i HMAC\_SHA-1 që përdor algoritmin e thjeshtuar SHA-1 si funksionin bazë hash. Ai përfshin hapat e mëposhtëm:

* Përgatitni çelësin sekret:

Nëse çelësi sekret është më i gjatë se gjatësia e bllokut (në bajt) të SHA-1 të thjeshtuar, hash çelësin duke përdorur SHA-1 të thjeshtuar për të marrë një çelës me gjatësi fikse.

Nëse çelësi sekret është më i shkurtër se gjatësia e bllokut, mbulojeni atë me zero që të përputhet me gjatësinë e bllokut.

* Krijoni çelësin e brendshëm:

XOR çdo bajt të çelësit sekret të përgatitur me vlerë 0x36.

Përsëriteni operacionin XOR për gjatësinë e bllokut.

* Krijoni çelësin e jashtëm:

XOR çdo bajt të çelësit sekret të përgatitur me vlerën 0x5C.

Përsëriteni operacionin XOR për gjatësinë e bllokut.

* Llogaritni hash-in e brendshëm:

Lidhni çelësin e brendshëm me mesazhin.

Llogaritni hash-in e vlerës së bashkuar duke përdorur algoritmin e thjeshtuar SHA-1.

* Llogaritni hash-in e jashtëm:

Lidhni çelësin e jashtëm me hash-in e brendshëm (të marrë nga hapi 4).

Llogaritni hash-in e vlerës së bashkuar duke përdorur algoritmin e thjeshtuar SHA-1.

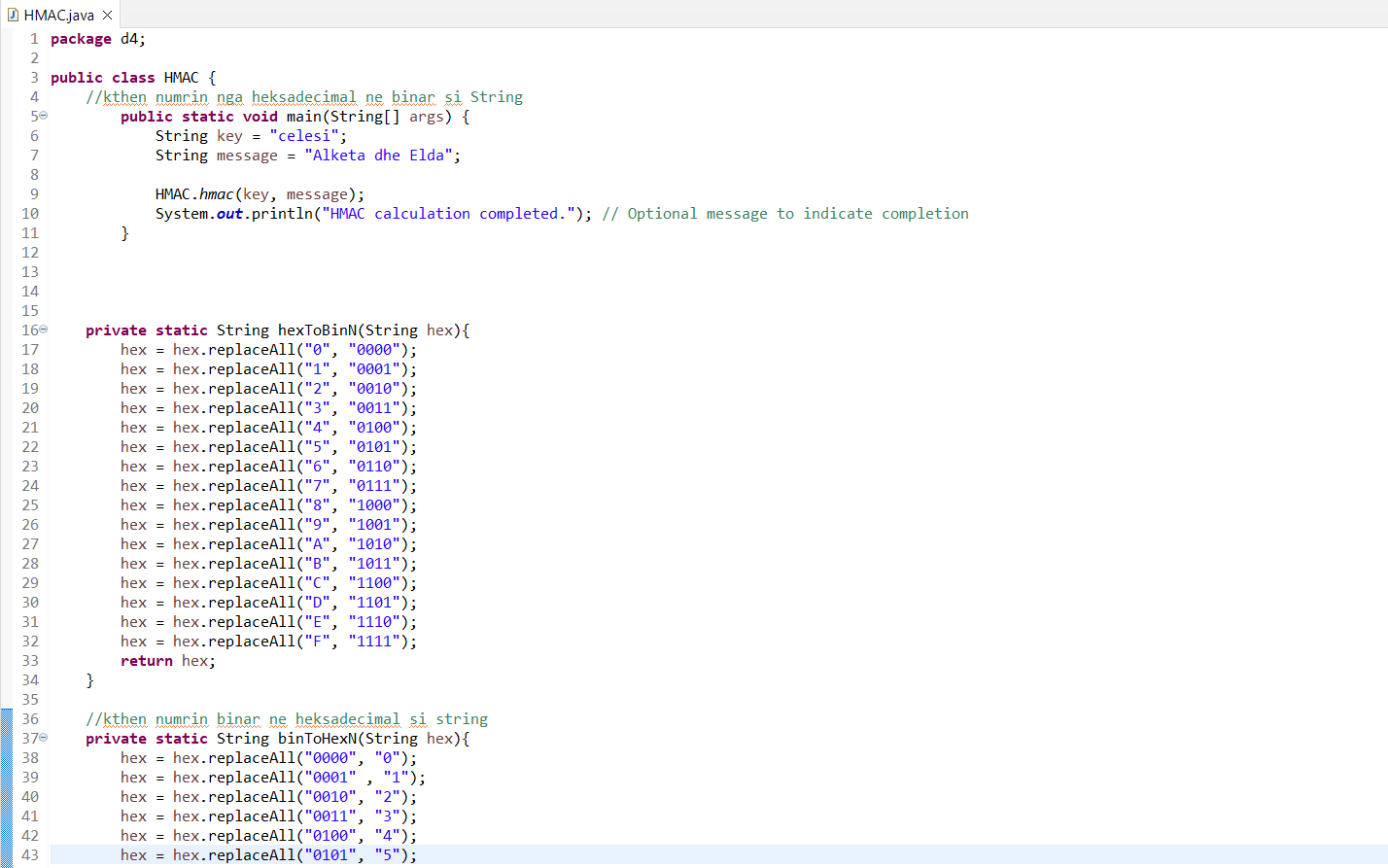
Hash-i që rezulton nga hapi 5 është vlera e koduar HMAC\_Simplified\_SHA-1 e mesazhit me çelësin e dhënë sekret.

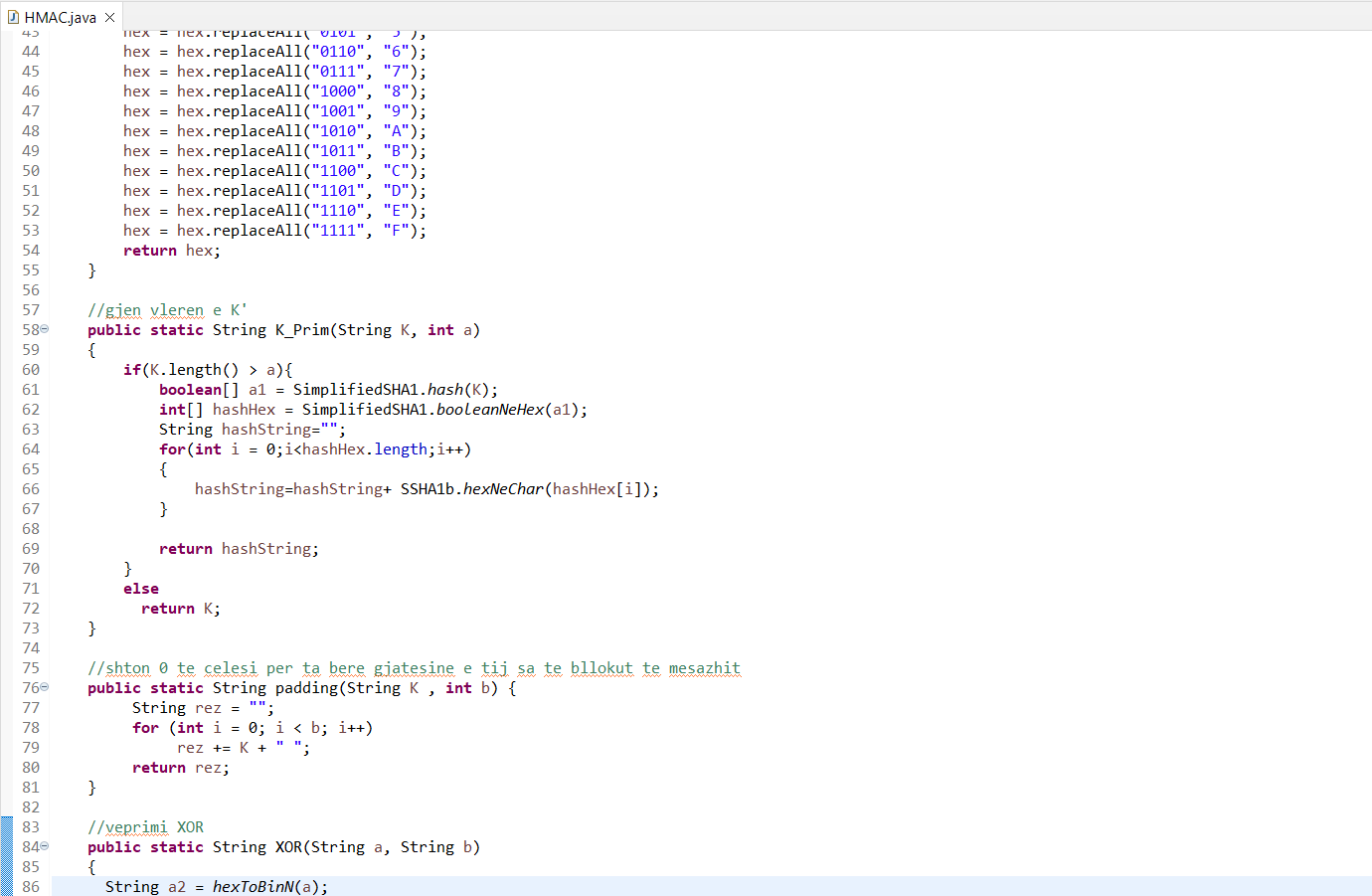
# **Dokumetimi teknik i kodit:**

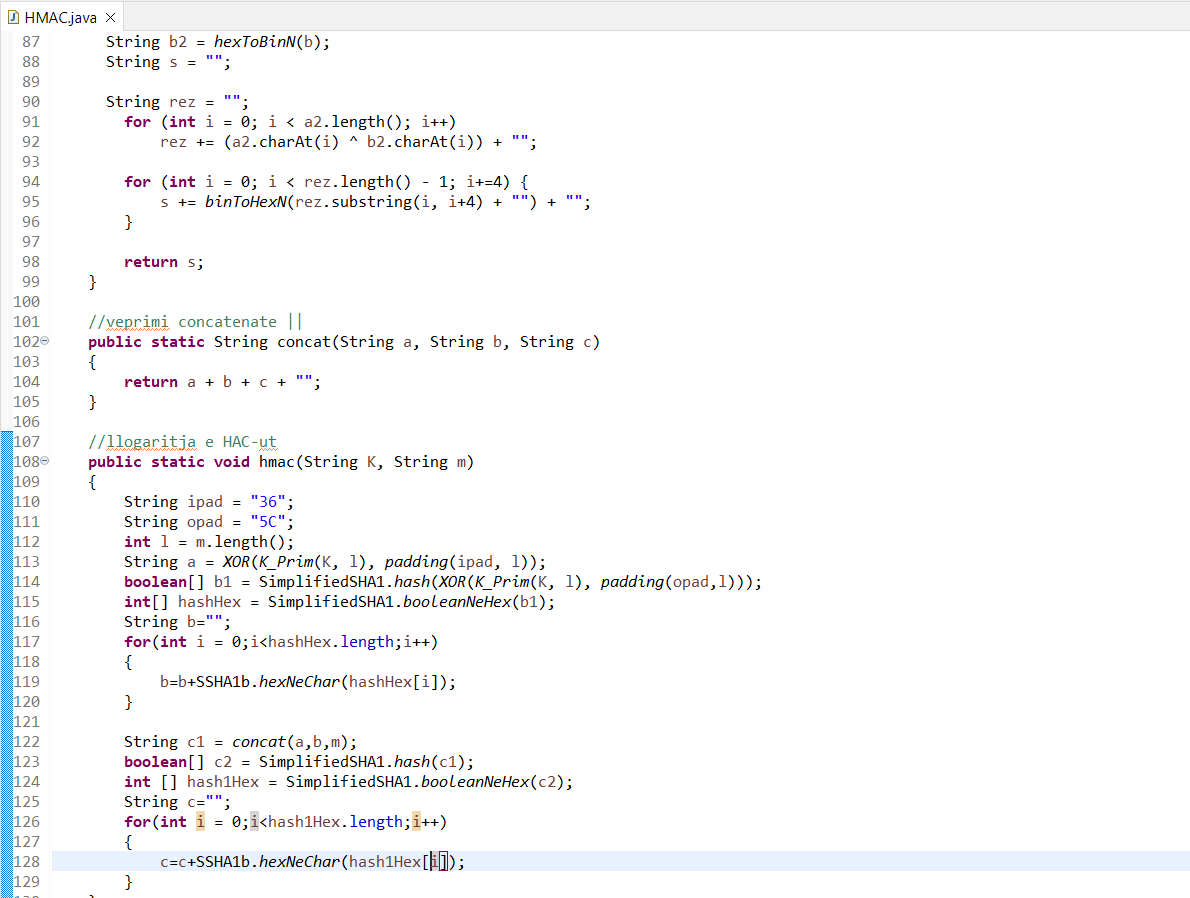
Kodi përbëhet nga tri klasa: HMAC, SimplifiedSHA1 dhe SHHA1b

Klasa SimplifiedSHA1 ofron metoda ndihmëse për kryerjen e operacioneve SHA-1 (Secure Hash Algorithm 1). Ai përfshin metoda për operacione bit (XOR, AND, negacion), konvertim ndërmjet heksadecimal dhe binare, dhe metoda të tjera të dobishme. Klasa HMAC përmban logjikën kryesore për llogaritjen e HMAC. Ai përfshin metoda për konvertimin heksadecimal në binar dhe anasjelltas, mbushjen e çelësit, kryerjen e operacioneve XOR dhe lidhjes dhe llogaritjen e vlerës HMAC. Për të përdorur kodin, mund të krijoni një shembull të klasës HMAC dhe të thirrni

metodën hmac, duke kaluar çelësin dhe mesazhin si parametra. Metoda hmac do të llogarisë vlerën HMAC dhe do ta printojë atë.





Kodi i ofruar është një implementim Java i algoritmit HMAC (Hash-based Message Authentication Code). Le të kalojmë kodin hap pas hapi për të kuptuar funksionalitetin e tij:

Kodi fillon me deklaratën e klasës HMAC dhe metodën main.

Metoda main inicializon variablat kyç dhe mesazh me vlera mostër.

Metoda hmac thirret nga metoda main , duke kaluar çelësin dhe mesazhin.

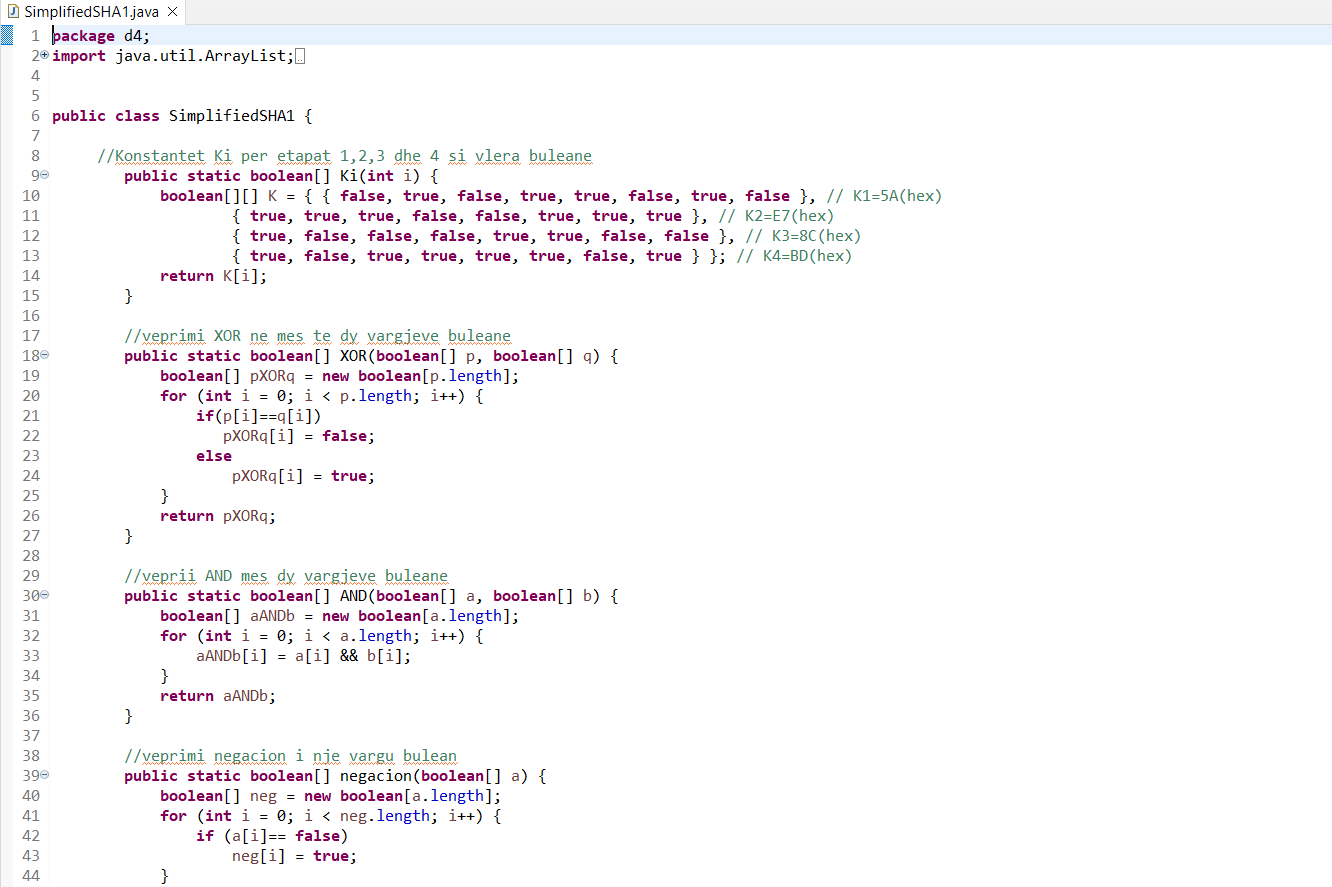
Metoda hmac kryen llogaritjen e HMAC.

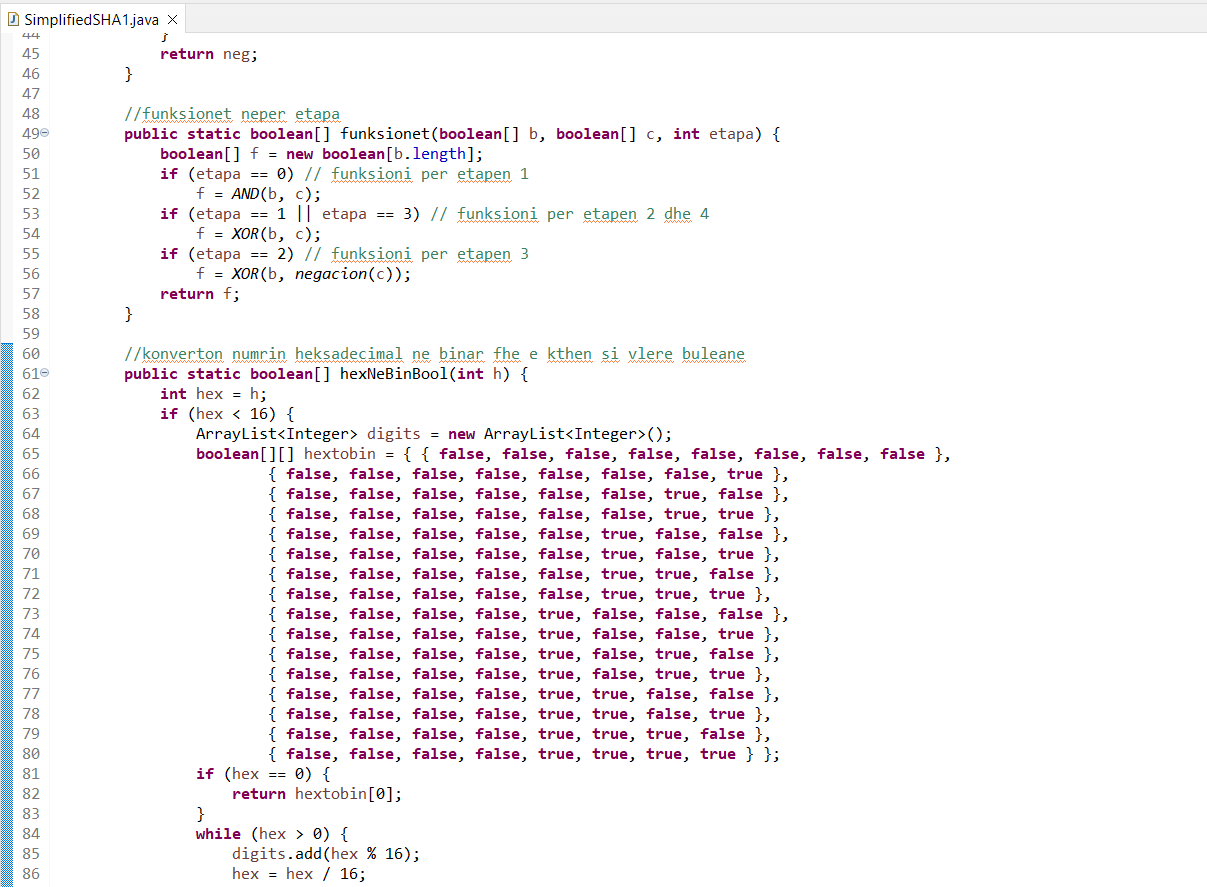
Metoda hmac fillimisht llogarit vlerën e K' (K prime) duke thirrur metodën K\_Prim. Nëse gjatësia e çelësit (K) është më e madhe se a, ai llogarit hash SHA-1 të K duke përdorur klasën SimplifiedSHA1 (e cila nuk është dhënë në copëzën e kodit). Përndryshe, ai e kthen çelësin siç është.

Metoda padding quhet për të shtuar mbushje në ipad (jastëk i brendshëm) duke përdorur gjatësinë e mesazhit (m).

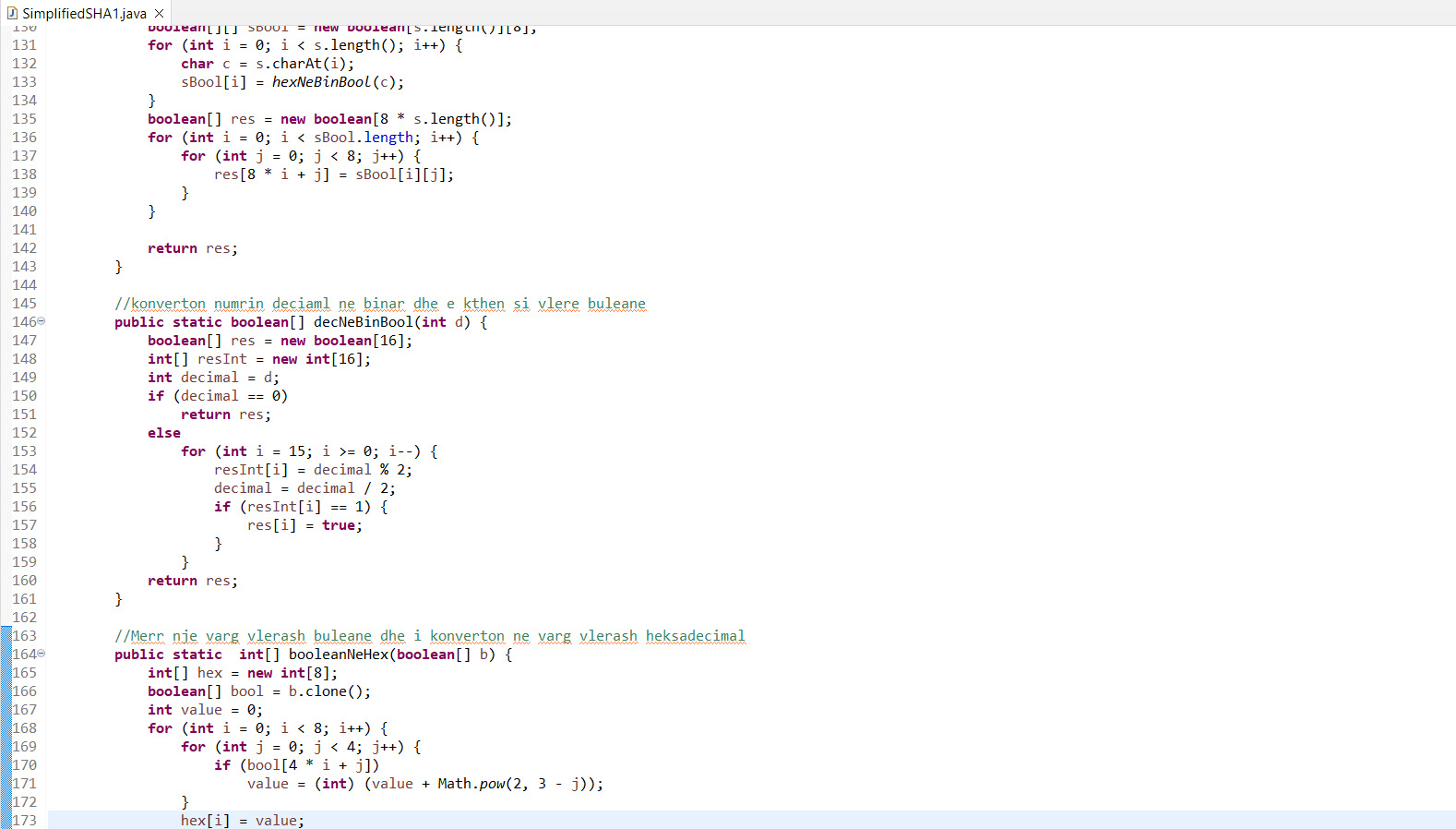
Metoda XOR thirret dy herë për të kryer operacionin XOR. Ai merr dy vargje heksadecimal (a dhe b), i konverton ato në vargje binare, kryen operacionin XOR pak nga pak dhe e kthen rezultatin si një varg heksadecimal.

Metoda concat bashkon tre vargje (a b dhe c) duke përdorur || operator dhe kthen vargun e bashkuar.

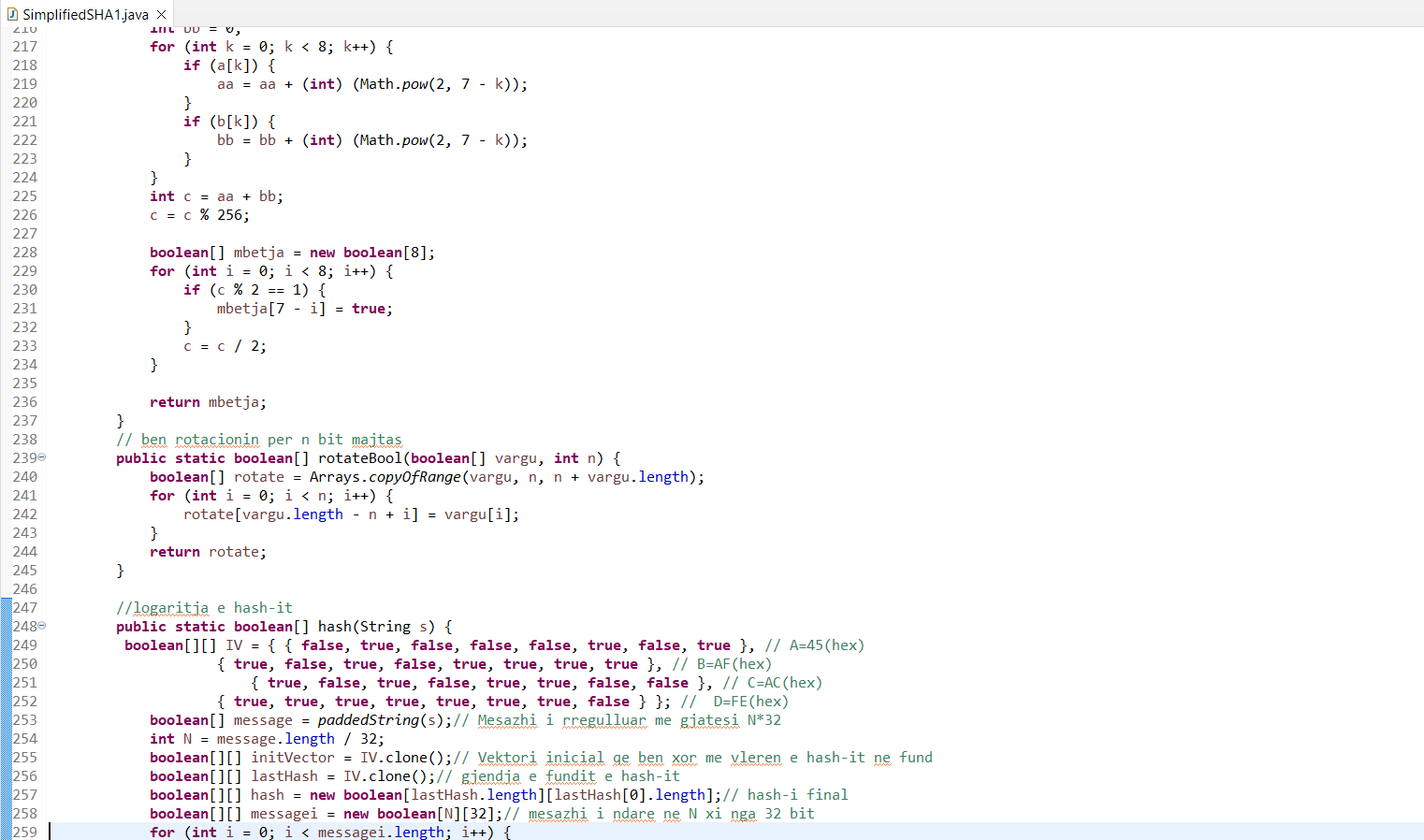


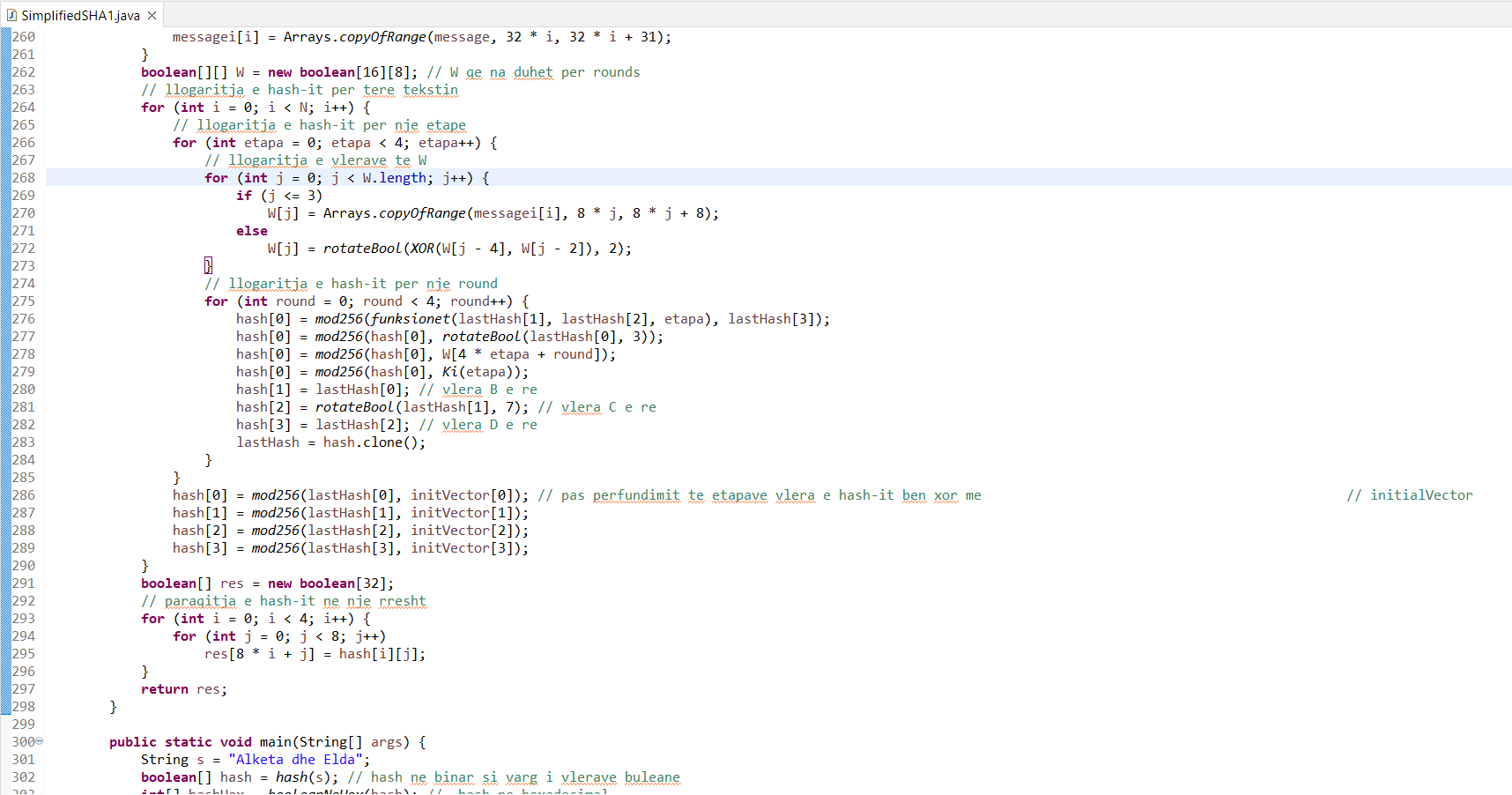


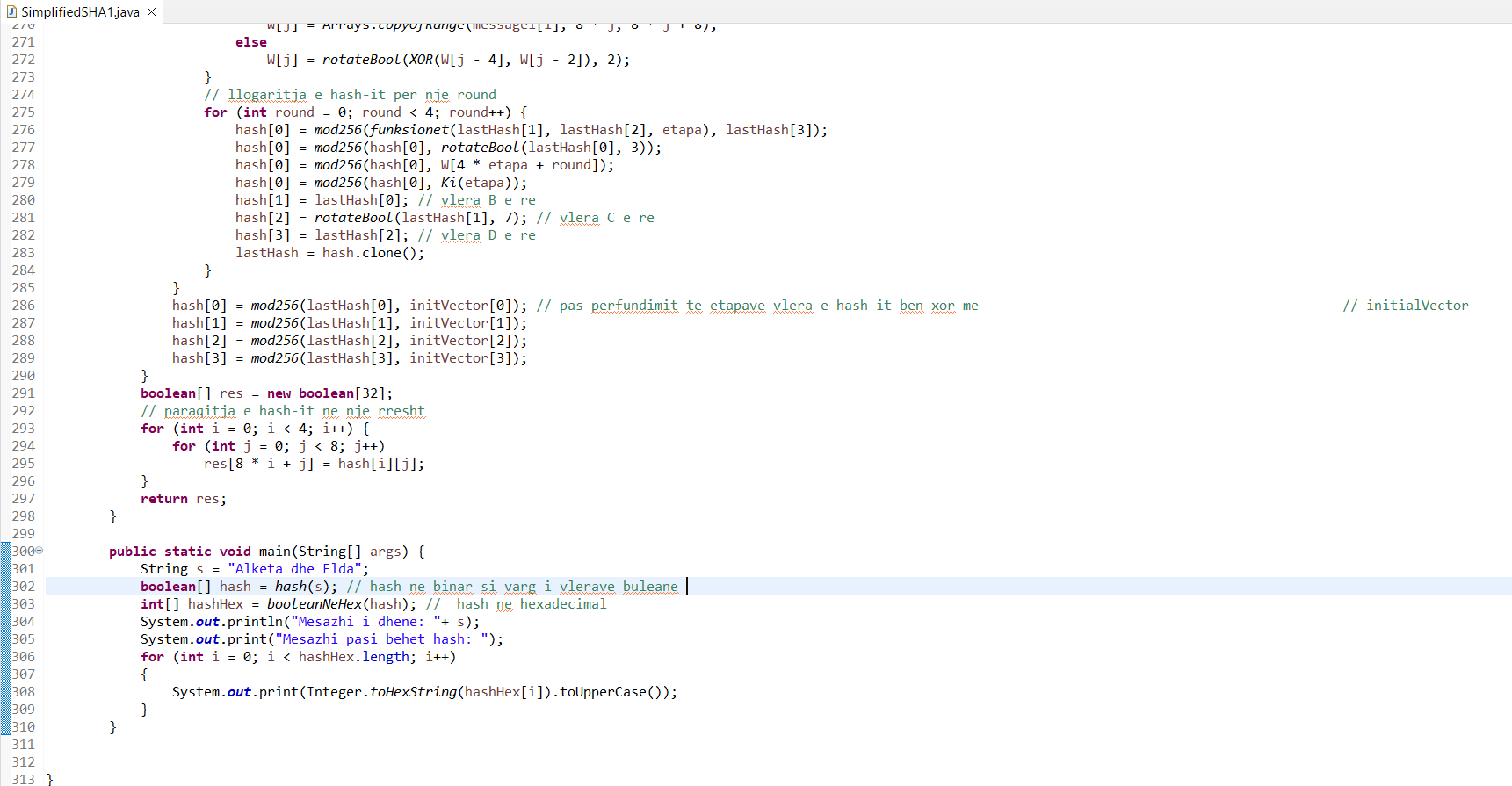












Kodi zbaton një version të thjeshtuar të algoritmit SHA-1 (Secure Hash Algorithm 1). SHA-1 është një funksion hash kriptografik që prodhon një vlerë hash 160-bit (20-byte) të njohur si një përmbledhje e mesazheve. Kodi i dhënë zbaton funksione dhe operacione të ndryshme të përdorura në algoritmin SHA-1, të tilla si operacione në bit, operacione logjike, konvertime midis formateve të ndryshme të numrave dhe funksionin kryesor hash.

Këtu është një përmbledhje e funksioneve dhe operacioneve të zbatuara në kod:

Ki(int i): Ky funksion kthen vlerat konstante (Ki) të përdorura në faza të ndryshme të algoritmit SHA-1.

XOR(boolean[] p, boolean[] q): Kryen operacionin XOR midis dy vargjeve boolean.

AND(boolean[] a, boolean[] b): Kryen veprimin AND midis dy vargjeve boolean.

negacion(boolean[] a): Negacionon një grup boolean.

funksionet(boolean[] b, boolean[] c, int etapa): Zbaton funksionet e ndryshme logjike të përdorura në secilën fazë të algoritmit SHA-1.

hexNeBinBool(int h): Konverton një numër heksadecimal në një paraqitje të vargut boolean.

stringNeBoolean(String a): Konverton një varg në një paraqitje të vargut boolean.

decNeBinBool(int d): Konverton një numër dhjetor në një paraqitje të vargut boolean.

booleanNeHex(boolean[] b): Konverton një grup boolean në një paraqitje të numrave heksadecimal.

intToBoolean(int[] a): Konverton një grup numrash të plotë në një paraqitje të vargut boolean.

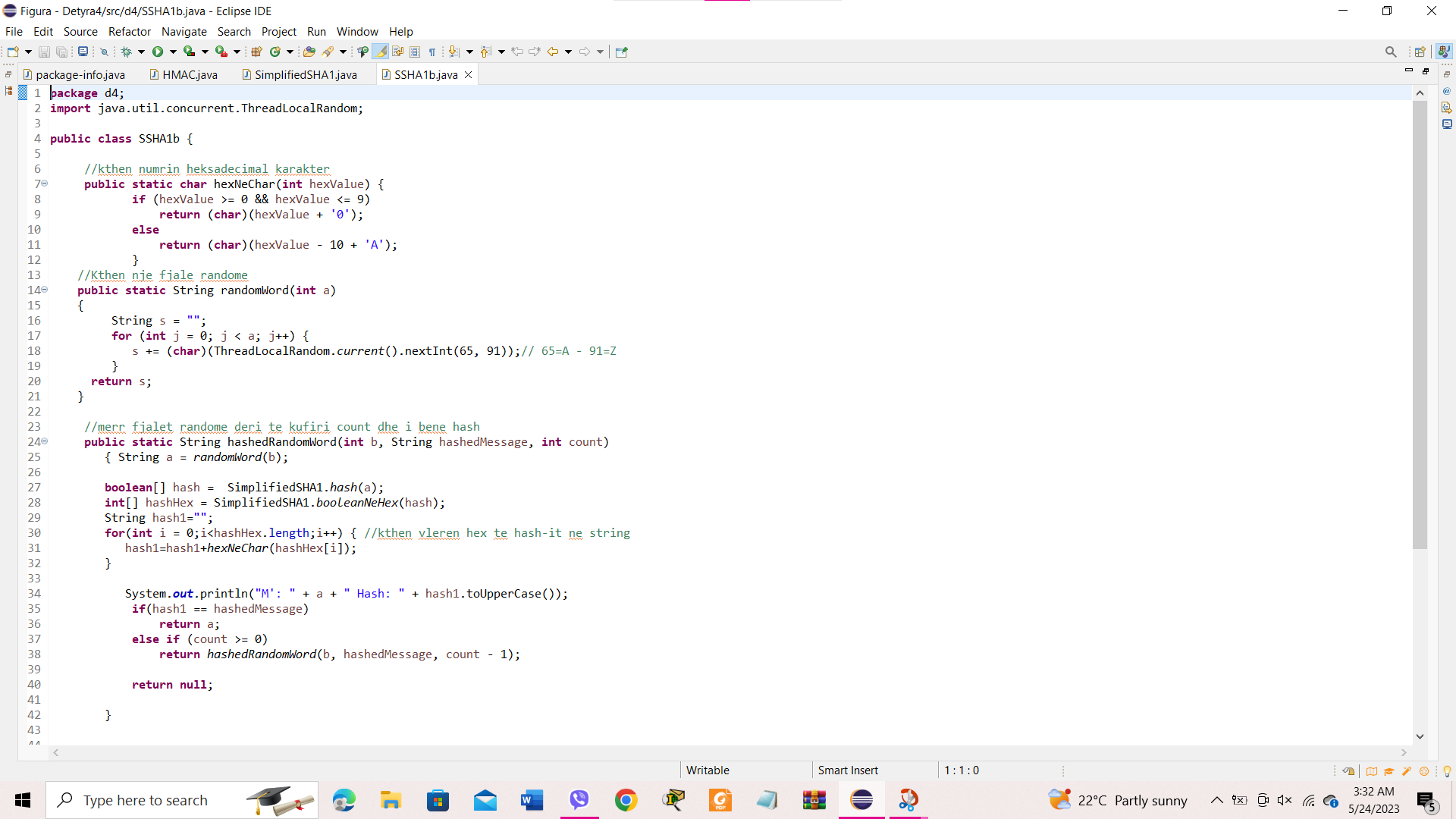
paddedString(String a): Rikthen paraqitjen e vargut të mbushur të mesazhit hyrës për algoritmin SHA-1.

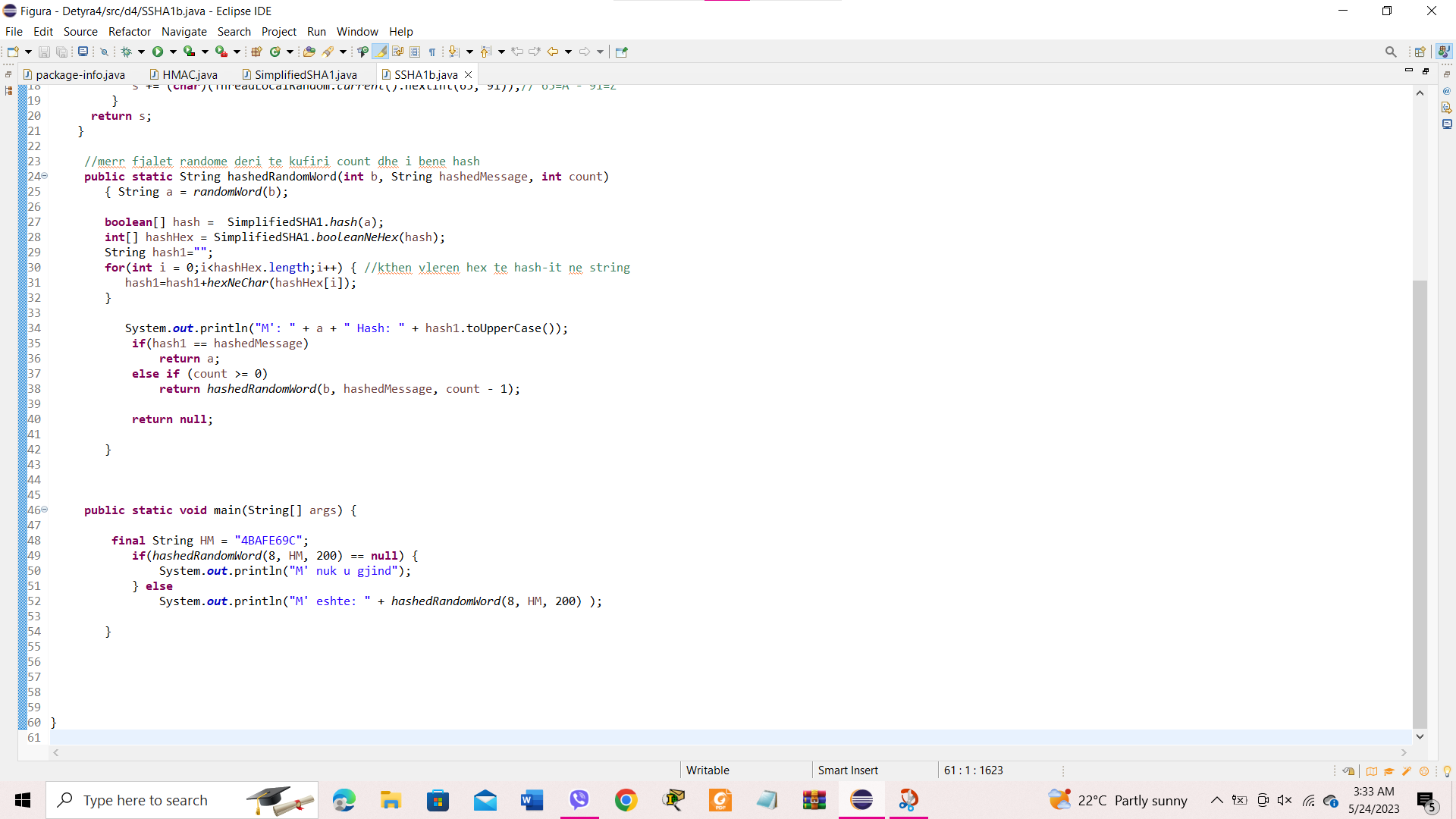
mod256(boolean[] a, boolean[] b): Kryen operacionin modulo 256 në dy vargje boolean.

rotateBool(boolean[] vargu, int n): Kryen një rrotullim majtas të një vargu boolean me n bit.

hash(String s): Zbaton funksionin kryesor hash të algoritmit të thjeshtuar SHA-1.

Është e rëndësishme të theksohet se ky zbatim është një version i thjeshtuar i algoritmit SHA-1 dhe mund të mos ofrojë të njëjtin nivel sigurie si algoritmi zyrtar SHA-1. Në praktikë, rekomandohet përdorimi i zbatimit zyrtar ose një algoritmi hash më i sigurt, si SHA-256 ose SHA-3, për qëllime kriptografike.





Ky kod Java është një zbatim i thjeshtuar i algoritmit SHA-1 për gjenerimin e një mesazhi hash. Le të kalojmë kodin hap pas hapi:

Funksioni hexNeChar konverton një vlerë heksadecimal (0-15) në paraqitjen e tij përkatëse të karaktereve. Ai paraqet vlerat 0-9 në '0'-'9' dhe vlerat 10-15 në 'A'-'F'.

Funksioni randomWord gjeneron një fjalë të rastësishme me gjatësi a. Ai përdor klasën ThreadLocalRandom për të gjeneruar numra të rastësishëm midis 65 dhe 90, të cilët korrespondojnë me vlerat ASCII të shkronjave të mëdha 'A' në 'Z'. Ai bashkon këto karaktere për të formuar fjalën e rastësishme.

Funksioni hashedRandomWord merr tre parametra: b (gjatësia e fjalës së rastësishme), hashedMessage (vlera e pritur hash) dhe count (një numërues për rekursion).

Brenda funksionit:

Ai gjeneron një fjalë të rastësishme a me gjatësi b duke përdorur funksionin randomWord.

Ai llogarit hash-in SHA-1 të fjalës a duke përdorur funksionin SimplifiedSHA1.hash, i cili kthen një grup boolean që përfaqëson hash-in.

Ai konverton paraqitjen e grupit boolean të hash-it në një paraqitje heksadecimal duke përdorur funksionin SimplifiedSHA1.booleanNeHex, i cili kthen një grup me numra të plotë.

Ai konverton vlerat heksadecimal në një paraqitje të vargut.

Ai krahason hash-in e krijuar (hash1) me hash-in e pritur (hashedMessage).

Nëse ato janë të barabarta, ajo kthen fjalën e gjeneruar a.

Nëse nuk është i barabartë dhe numri është më i madh ose i barabartë me 0, ai thërret në mënyrë rekursive funksionin hashedRandomWord me një numërim të zvogëluar.

Nëse numërimi arrin 0 dhe nuk gjendet asnjë përputhje, ai kthehet null.

Në funksionin kryesor, ai inicializon një HM konstante me vlerën e pritur hash si varg.

Nëse funksioni hashedRandomWord kthehet null, ai printon "M' nuk u gjind" (M' nuk u gjet).

Përndryshe, printon "M' eshte: " e ndjekur nga vlera e kthyer nga funksioni hashedRandomWord.

Ky kod përpiqet të gjejë një fjalë të rastësishme (a) hash-i i së cilës SHA-1 përputhet me hash-in e pritur (hashedMessage). Ai e bën këtë duke gjeneruar fjalë të rastësishme, duke llogaritur hash-et e tyre dhe duke i krahasuar ato me hash-in e pritur. Nëse gjendet një përputhje, ajo kthen fjalën; përndryshe, ai vazhdon të gjenerojë dhe k