Лабараторная работа “ Выкарыстанне тэхналогіі RMI. Шыфраванне метадам квадрата Палібія “

Выканала: студэнтка гр.272301, Фядзюк Кацярына

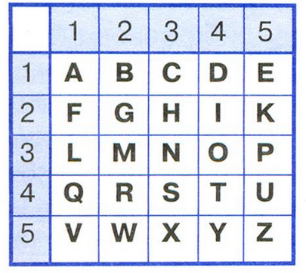
Далей пойдзе падрабязная справаздача аб выканай працы.

1. **Мэта**: пазнаёміцца з такой тэхналогіяй, як RMI , навучыцца выкарыстоўваць яе на практыцы.
2. **Тэорыя**. RMI – Remote Method Invocation. Карацей, механізм, які дазваляе вызваць метад адалёнага аб’екта. Усе аперацыі па падрыхтоўке і перадачы дадзеных інкаспулюцца ў метадзе кліентскага аб’екта-заглушкі (stub), які вызываецца. Асаблівасцямі метада з’яўляюцца:

* Лакальныя аб’екты перадаюцца па значэнні;
* Аб’екты, якія перадаюцца, павіны быць Serializable.
* Пры перадачы адалёнага аб’екту, калі ён экспарціраван, перадаецца stub гэтага аб’екта.

1. **Апісанне лабараторнай работы**.

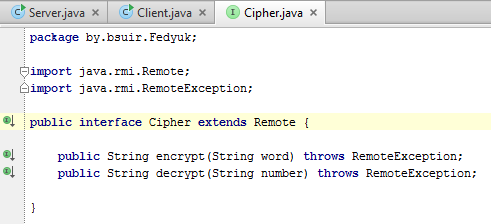
Праграмма будзе выконваць шыфрацыі і дэшыфрацыю уведзеных сімвалаў з дапамогай метада квадрата Палібія. Квадрат Палібія ўяўляе сабой матрыцу, якая ў нашым выпадку, складаецца з літар ангельскага алфавіту .



Кожная літара мае свой адрас, які ўключае ў сабе нумар радка і слупка. Шыфрацыя заключаецца ў тым, што калі карыстальнік уводзіць нейкія словы ці тэкст, яму павіннен вяртацца набор адрасоў кожнай уведзенай літары з квадрату Палібія ў парадку, у якім былі ўведзены літары. Дэшыфрацыя адбываецца наадварот: у асаблівым парадку ўведзеныя лічбы параўноўваюцца з адрасамі літар у квадраце Палібія, а потым вяртаецца пэўны расшыфраваны з лічбаў тэкст.

Такім чынам, каб рэалізаваць дадзеную праграму нам патрэбны 2 асноўных класа: Server і Client, а таксама інтэрфейс Cipher. Зараз падрабязней аб кожным класе.

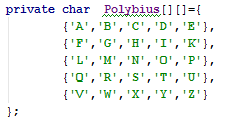
**Cipher.** Гэта інтэрфейс, у якім аб’яўлены два асноўных метада, якія будуць рэалізованы ў нашай праграме: гэта метад шыфрацыі - *encrypt* і метад дэшыфрацыі – *decrypt.*



Малюнак 1. – Інтэрфейс Cipher.

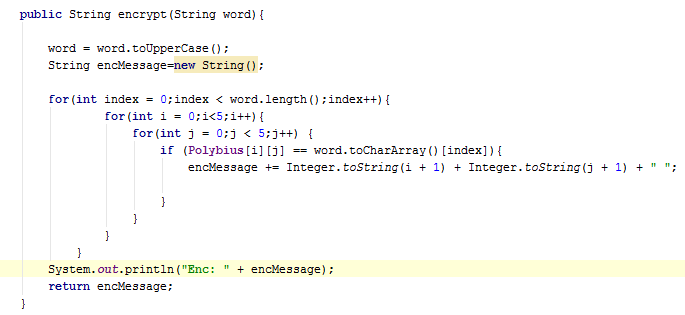
Саму рэалізацыю двух метадаў разгледзім у наступным класе Server.

**Server.** Перш за ўсё для таго, каб рэалізаваць у гэтым класе тэхналогію RMI падключым да яго тры пакеты:java.rmi.registry.LocateRegistry, java.rmi.registry.Registry, java.rmi.server.UnicastRemoteObject; Далей укажам, што клас Server выконвае інтэрфейс Cipher. Яшчэ важным момантам з’яўляецца тое, што адным з полей дадзеных у класе Server гэта сам квадрат Палібія.



Малюнак 2. – Квадрат Палібія.

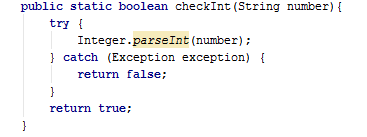
У класе бачым рэалізацыю метада шыфрацыі *encrypt*. Параметрам метада з’яўляецца аб’ект тыпу String. Спачатку аб’яўляем новы аб’ект тыпу String, у які мы і будзем запісваць канчатковае паведамленне, яго і вяртае метад шыфрацыі. Потым у цыкле кожны сімвал з квадрата Палібія параўноўваем з кожнай літарай слова, якое адправілі на шыфрацыю. Калі якісьці сімвал з квадрата Палібія супадае з літарай слова, то запісваем адрас ( нумар радка і слупка) гэтага сімвала ў канчатковае паведамленне праз прабел. Далей выводзім паведамленне для сервера аб тым, што шыфрацыя пройдзена і вось вынік.



Малюнак 3. – Рэалізацыя метада encrypt.

У метадзе main проста ствараецца аб’ект тыпу Server, а таксама аб'ект тыпу Cipher.

**Client.** У кліенце існую толькі галоўны метад main. Там апісана асноўнае кліентскае меню, дзе ён можа выбраць тры варыянта працягу працы. Уводзячы нумар неабходнай аперацыі, таксама выконваецца праверка на тое, ці з’яўляецца ўведзены сімвал лічбай. Гэта выконваецца з дапамогай метада checkInt.



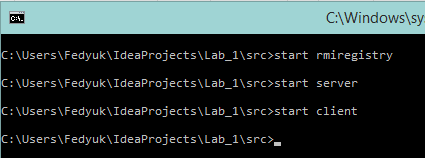
Малюнак 4. – Апісанне метаду checkInt.

Далей у залежнасці ад выбару кліента выконваюцца аперацыі шыфрацыі, дэшыфрацыі, або проста выхад з праграмы. Важным момантам з’яўляецца тое, што ў абодвух выпадках выкарыстоўваюцца рэгулярныя выразы. Калі робім шыфрацыю, правяраем ці з’яўляюцца ўведзеныя сімвалы літарамі і наадварот, калі робім дэшыфрацыю, правяраем, каб уведзеныя сімвалы былі лічбамі. На гэтым асноўнае апісанне праграмы скончваецца.

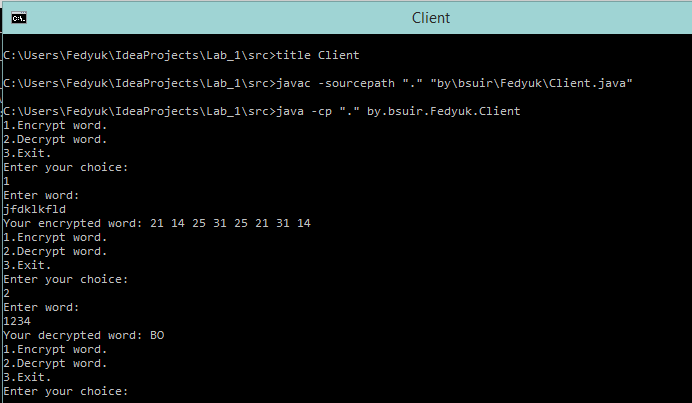
1. **Вынікі работы праграммы.**

Спачатку на аснове файлаў Server.java і Client.java ствараем два файла з пашырэннем .bat. Далей з дапамогай каманднага радка запускаем праграму. Спачатку рэгіструем rmi, а потым з дапамогай запытаў start server і start client запускаем файлы server.bat, client.bat.

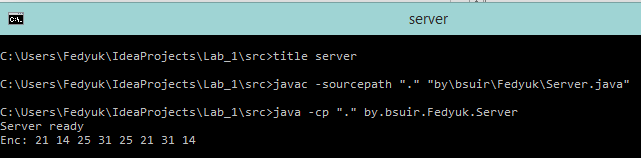
Прыклад работы праграмы прыведзены на малюнках.



Малюнак 5. Галоўны камандны радок.



Малюнак 6. Выгляд кліенту.



Малюнак 6. Выгляд серверу.

1. **Вынік.** Такім чынам, я пазнаёмілася з тэхналогіяй RMI і зрабіла кліент-сервернае прыкладанне, якое выконвае шыфрацыю і дэшыфрацыю тэксту.