1, 2, 3, кодирај ! - Активност циклуса 4 - Пројект « Криптографија » - Час 8: Програмирање Цезаревог шифрирање(1/4)

Доминантна дисциплина	Математика
Резиме	Ученици приказују свој алгоритам и листу етапа које ће реализовати при структуирању свог пројекта програмирања Цезаревог шифрирања. Програмирају, у *Скрачу*, прву функцију која им омогућује да нађу одговарајуће слово на месту које му је одређено у азбуци.
Појмови	« Језици » • Програм представља исказ неког алгоритма у неком програмском језику • *Скрач* представља окружење графичког програмирања • Неке инструкције се извршавају једна после друге па говоримо о секвенцијалном програмирању. « Машине » • Машине које нас окружују само извршавају наредбе (инструкције) дате у одговарајућим језицима • Комбинацијом елементарних инструкција у могућности смо да решимо комплексне задатке. • Нека варијабла је име које дајемо у некој зони меморије. Омогућује стокирање неке вредности а касније и њено поновно коришћење или модификацију. « Добре навике програмирања » • Када се неки блок инструкција мора употребити више пута у неком програму погодно га је интегрисати у неку функцију. • Почетну вредност неке варијабле је потребно дати при њеној креацији. • Варијабле и функције морају имати јасно дефинисана имена. « Алгоритми » • Неки алгоритам може да садржи инструкције, петље, тестове, варијабле.

	• Тест омогућује избор реализације неке акције зависно од тога да ли је услов верификован или не.
Материјал	За сваки пар ученика • Компјутер с везом на интернет (при коришћењу on-line верзије *Скрача*) или је *Scratch* претходно инсталиран • (факултативно) фотокопија Радног листа-B11 За одељење • Видеопројектор који се користипри заједничком представљању

Педагошка напомена

- Претпостављамо да су ученици (и наставници) већ програмирали у *Скрачу* и да знају основе (креирања неког програма, његово меморисање, употребу најчешће коришћених команди …).
  - Ако је то случај, онда часови који следе неће представљати неки пробле.
  - Ака пак то није случај, онда читаоца усмеравамо на поглавље « Општи увод у Скрач » где могу да се упознају с уопштеном дискусијом (зашто смо одабрали *Скрач,* које су његове предности и ограничења, како га инсталирати, итд.) као и уводним где могу да се сроде с његовим интерфејсом, најједноставнијим вежбама, итд.).
- Као што је објашњено у поглављу « Откривање Скрача », ученици у својим активностима програмирања напредују различитим ритмом. Наставнике охрабујемо да им дозволе, колико год је то могуће, да напредују својим ритмом.
- Управо из тог разлога је подела часа индикативна и не мора бити прихваћена у свакој конкретној ситуацији. Неки парови ученика неће прећи све предвиђене етапа, на крају сваког часа, док ће други већ прећи и на следеће часове …

Етапа 1: дефинисање различитих етапа пројекта (20 минута; заједнички)

Наставник објашњава ученицима да ће правити програме који омогућују знатно лакшу реализацију неких активности него када су ту исту активност реализовали, као до сада, « без компјутера » (или « ручно »). Предлаже да почну с најједноставнијим Цезаревим шифрирањем.

Одељење, на почетку часа, заједнички размишља о алгоритму оваквог шифрирања и праве листу неопходних елемената програма који би произвео оно што они од њега очекују:

- Свако слово које фигурише у реалној поруци је замењено словом које је померено за известан број места (« кључ ») у азбуци.
- Програм дакле мора да:
    - Омогући кориснику да сазна шифрирану поруку (која мора бити стокирана у некој варијабли, названој на пример « реална_порука »)
    - Омогући кориснику да сазна кључ (који такође мора бити стокиран у оквиру неке варијабл, назване « кључ »)
    - Стокира азбуку у некој варијабли (коју једноставно назива « азбука »)
    - Шифрира поруку, и стокира резултат у некој новој варијабли (названој, на пример « шифрирана_порука »), која мора да буде приказана на екрану.

Педагошке напомене:

- Врло је мала вероватноћа да су се ученици, чак и када су већ користили *Скрач*, толико сродили с појмом алгоритма да могу да речима опишу оно што мора да ради програм пре него што реализују само програмирање. Зато вам предлажемо да ову етапу урадите заједнички. Ако су пак већ доста напредовали у овом пројекту онда могу и да покушају да задатак реше индивидуално.
- Заједничко објашњење има и другу предност јер дозвољава фиксирање општег оквира који ће ученици забележити, да би затим могли да раде аутономно без неке присиле да сви напредују истим ритмом.

Пошто одељење утврди шта програм треба да уради, и како то мора да уради, преостаје им важна етапа у оквиру заједничке дискусије којом би требало да утврде како ће то урадити?
У питању је разлагање пројекта (што је врло комплексан задатак) на више подпројеката (елементарнији задаци).
Ево како би, на пример, могли да « изделимо » један такав пројект:

- У почетку ћемо тражити да се разматра само једно слово, а не комплетна порука
- Да би занали како да померимо слово за одређен број места у азбуци, морамо претходно знати на ком је месту то слово у азбуци (на пример, слово n°3 је C (*у латиници, а В у ћирилици, п.п*) и обрнуто, које је слово на месту које одговара датом рангу (слово F заузима место n°6). Наставник сугерише да се решава један по један задатак, и да се почне с оним најлакшим, рецимо налажење слова коме одговара неки ранг у азбуци.
- Морамо да померамо слова за одређен број места, али кад пређемо 26 (слово Z), морамо да се вратимо на 1 (слово A) (*слично је и у ћирилици, кад пређемо 30 (слово Ш), морамо да се вратимо на 1 (слово А), п.п).*

Одељење је дефинислао следеће етапе пројекта:

Тежина	Број етапе	Природа задатак које треба реализовати
	Етапа 1 – Дефинисање етапа пројекта	• Ово је рад који одељење управо реализује, и то је врло битна етапа почетка пројекта.
	Етапа 2 – Налажење слова одређеног ранга у азбуци	• Писање програма који тражи неки ранг и даје слово које има тај ранг у азбуци.

	Етапа 3 – Креирање функције (1/2)	• Упознавање манипулације функцијама у неком прогаму.
	Етапа 4 – Креирање функције (2/2)	• Писање функције која задовољава циљ етапе 2. • Писање програма за тестирање ове функције.
	Етапа 5 – Налажење ранга неког слова	• Писање програма који тражи слово и даје његов ранг у азбуци. • Формирање и тестирање функције.
	Етапа 6 – Шифрирање слова	• Писање програма који тражи кључ, а затим слово које шифрира тим кључем. • Формирање и тестирање функције. • Трансформација ове функције да би она функционисала ако је ранг + кључ веће од 26.
	Етапа 7 – Шифрирање целе поруке	• Писање програма који тражи кључ и поруку, и који приказује шифрирану поруку.

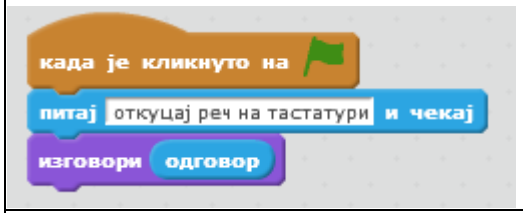
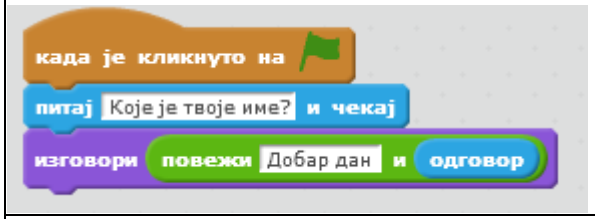
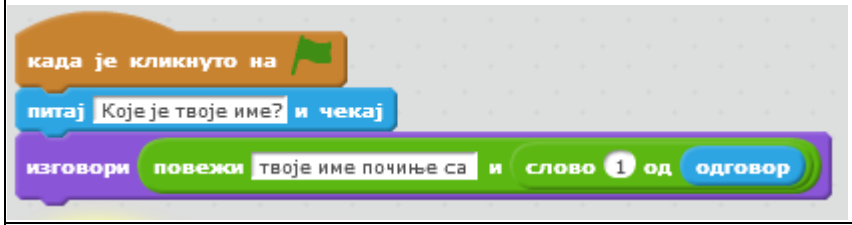
Научне напомене

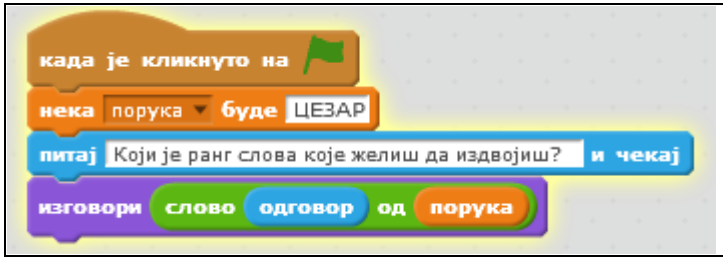
- Разлагање рада програмирања на елементарне задатке омогућује поступно тестирање различитих делова програма. Што је знатно лакше и сигурније него када се напише дуг програм и тестира на крају (у овом случају се обично налазе бројни багови које је тешко елиминсати)
- Рад помоћу функција омогућује, пошто се функција једном напише, да се пређе на неку другу ствар без потребе да се разматра део тог програма.

Етапа 2: налажење слова одређеног ранга у азбуци (50 минута)

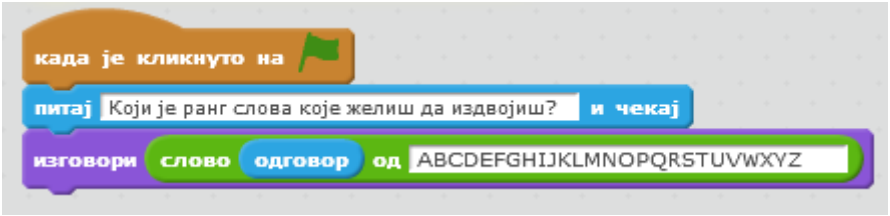
Педагошка напомена:
Ова прва етапа је врло комплексан задатак јер ученици морају да реализује много операција по први пут (манипулација низом карактера и варијабли …).
Предлажемо вам да правите често заједничка представљања резултата ученичког рада.

Ова етапа захтева употребу команде « питај » у *Скрачу*. Која је доступна у делу светло плаве боје назване « осећаји », ова команда омогућује да се постави питање и стокира резултат у некој варијабли (која већ постоји) названој« одговор ».
Ако ученици нису до сада користили ове функционалности, и нису манипулисали низом карактера у Скрачу, предлажемо вам да им дате вежбе које се налазе на Радном листу-B11. Корекција је:

Програм 1 	Програм копира на екрану реч коју је корисник написао на тстатури (а која је меморисана у варијабли « одговор », креираној аутоматски).
Програм 2 	Програм понавља име које је откуцано на тастатури, коме претходи « добар дан » (с простором иза « добар дана » да не би дошло до спајања речи).
Програм 3 	Програм шаље прво слово имена откуцаног на тастатури.
Програм 4	Програм шаље одговарајуће слово које одговара броју откуцаном на тастатури. На пример, ако је корисник откуцао « 2 », програм шаље 2-го слово речи ЦЕЗАР, односно слово « Е ».

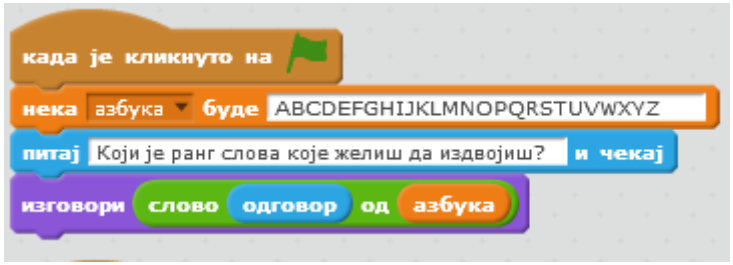
Одговор на постављени проблем (прикажи слово неког ранга у азбуци) може да се програмира на следећи начин:



Наставник, ако је потребно, може да раздели ову етапу на два једноставнија задатка, тражећи:

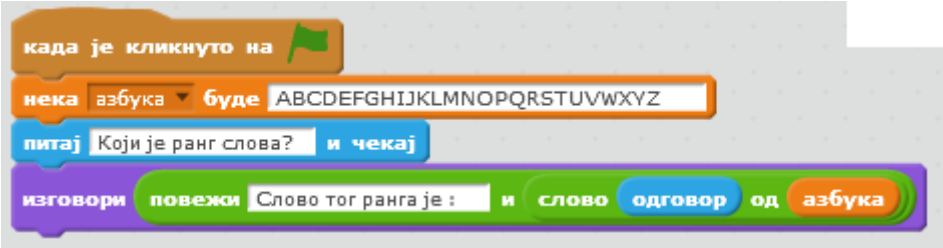
- 1. да се прикаже азбука
- 2. да се прикаже 13-то слово азбуке
- 3. да се постави питање ранга, а затим и приказ слова тог ранга

Вишеструко понављање целе азбуке се избегава, и поједностављује читање програма, стокирањем азбуке као варијабле. Програм постаје:



Научна напомена:
Креирање варијабле се реализује кликом на део « Подаци ».

Одговор можемо побољшати и тако што ћемо се више сродити с (string concatenation) конкатенацијом низова карактера (што ће нам бити од користи у последњој етапи):



Шта је научено

Одељење сумира шта је научено у овој етапи:

- програм омогућује да се нађе које слово се налази на датом месту у азбуци
- Коришћени су неки нови кључни концепти програмирња (или су већ од раније познати):
  - секвенца инструкција
  - варијабле
  - улаз, излаз
