

Испит траје 3 сата.

Задатак	1	2	3	4	5	6	7	8	9	10	1	Укупно
Максималан број поена	5	5	5	5	5	5	5	5	5	5	50	100

ТЕОРИЈСКИ ДЕО

...

ПРАКТИЧНИ ДЕО

1. Написати апликацију коришћењем програмског језика C++ и библиотеке Qt која симулира кретање путника по градовима. Слика 1 илуструје како програм треба да изгледа приликом покретања. У наставку следи опис рада програма.

- (a) Графички кориснички интерфејс имплементирати тако да изгледа као интерфејс приказан на слици 1. Елементе прозора распоредити тако да се приликом промене његове величине, компоненте аутоматски померају да задрже приказани распоред. Поља “Градови” и “Покупљени артикли” су типа QListWidget. Пољу “Пређени пут” онемогућити кориснички унос.
- (b) Сваки град представити као објекат класе `City`. Објекти ове класе се састоје од наредних атрибута: (1) име града као ниска, (2) његова X координата као неозначен цео број, (3) његова Y координата као неозначен цео број и (4) списак производа који се могу купити у том граду као низ ниски. Обезбедити наредни јавни интерфејс ове класе:
 - Метод `fromQVariant` десеријализује податке о једном граду из прослеђеног аргумента (`QVariant`). Аргумент садржи податке о свим атрибутима класе.
 - Метод `toString` израчунава текст са подацима о граду у формату који је приказан на слици 2.
 - Метод `getRandomArticle` који враћа псеудонасумични производ из тог града. Изабрани производ се уклања из листе производа града (потроши се). Ако град нема производ враћа празну ниску.
 - Метод `distance` који враћа Менхетн растојање између града над којим је позван метод и града који се прослеђује као аргумент. Менхетн растојање се рачуна по формулама: $|x_1 - x_2| + |y_1 - y_2|$

Дозвољено је додавање произвољних потребних поља и метода.

- (в) Кликом на дугме “Учитај градове”:

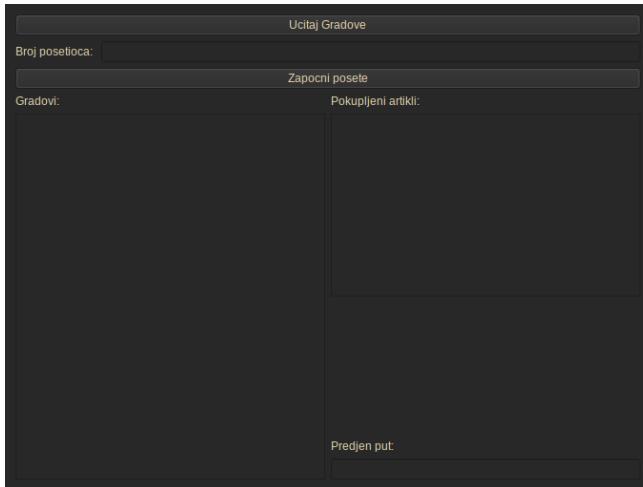
- Учитати из текстуалних датотека податке о градовима који се обилазе. Текстуална датотека је записана у JSON формату (видети пример датотека у директоријуму `gradovi` у поставкама; број градова и производа у сваком се може разликовати). Податке учитати као објекте класе `City`.
- Приказати градове у QListWidget контроли “Градови” као на слици 2.

Корисник може више пута учитавати податке пре започињања обилазака.

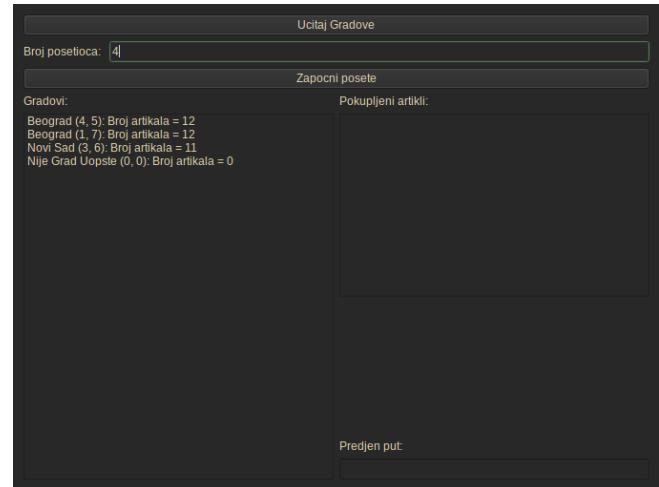
- (г) Обиласци се започињу кликом на дугме “Започни посете”. Уколико је листа градова празна или уколико у пољу “Број посетиоца” није унета целобројна вредност **строго већа од нуле**, не радити ништа. У супротном онемогућити кликтање на дугмад “Учитај градове” и “Започни посете”. Испразнити поље “Пређени пут”. QListWidget контролу “Покупљени артикли” такође испразнити. Покреће се број нити једнак броју у пољу “Број посетиоца”, где свака нит представља једног путника.
- (д) Након што је покренута, нит понавља псеудонасумичан број пута наредни низ корака, где се број понављања бира из интервала [3, 7]. Нит спава псеудонасумичан број милисекунди из интервала [250,500]. Потом се псеудонасумично бира нови град који ће путник посетити, где путник не може посетити град у којем се тренутно налази. Из тог града путник узима један производ и обавештава главну нит који је производ узео. Када обави предвиђени број посета, обавештава се главна нит о укупном пређеном путу тог путника у виду Менхетн растојања.

Када се бира први град, путник може отићи у било који град и не прелази никакво растојање. Све нити морају да конкурентно оперишу над истим подацима.

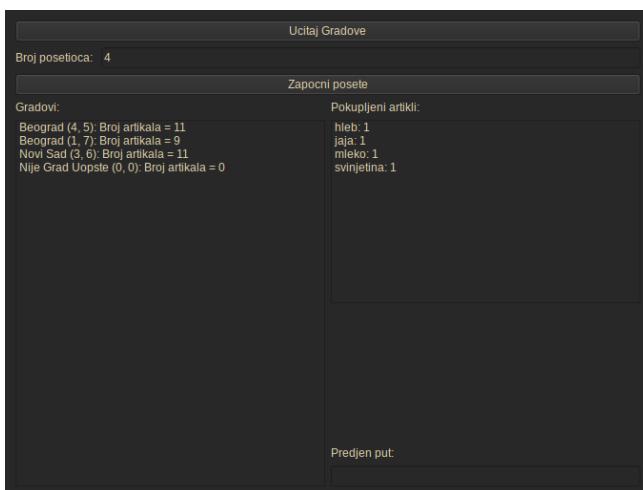
- (ђ) Сваки пут када главна нит добије информацију о узетом производу, она треба да ажурира податке у QListWidget контролама “Градови” и “Покупљени артикли” као на слици 3, где се приказује сваки производ који је сваки путник покупио и колико сваког производа има. Када се сви обиласци заврше, дугмад “Учитај Градове” и “Започни посете” се поново омогући, у пољу “Пређени пут” се уписује укупан пређени пут свих путника и главни прозор изгледа као на слици 4.
- (е) Осигурати се да не долази до проблема у конкурентном окружењу и водити рачуна о раду са динамичким ресурсима.



Слика 1: Приказ графичког интерфејса.



Слика 2: Приказ након учитавања градова.



Слика 3: Приказ током посете. Апликација приказује тренутно стање градова и тренутно покупљене производе.



Слика 4: Приказ након што се заврше све посете. Апликација приказује резултате и укупан пређени пут.