

დავალების ჩაბარების წესი

პროგრამირების ენა

პროგრამები შეგიძლიათ ნებისმიერ ენაზე დაწეროთ, რომლის კომპილატორი ან ინტერპრეტატორი უფასოა, მუშაობს ლინუქსზე და მარტივად მოიპოვება. მათ შორის შეგიძლიათ გამოიყენოთ: Java, C, C++, Python 2, Python 3 თუ რომელიმე სხვა პროგრამირების ენაზე გინდათ კოდის დაწერა, წინასწარ შემითანხმდით და მითხარით როგორ შეიძლება თქვენი პროგრამის ლინუქსზე გაშვება.

ამ კურსზე შედარებით პუპულარული ენებია: Python 3, Python 2 და Java, მაგრამ წლების განმავლობაში გვექნია C++, Ruby, C# და C-ს მაგალითებიც.

თუ წერთ Java-ზე: გაითვალისწინეთ რომ თქვენი პროგრამები ერთ package-ში არ უნდა იყოს გაერთიანებული. Package-ებში როგორც წესი ერთიანდება კლასები რომელთაგანაც მხოლოდ ერთ-ერთი უნდა გაეშვას და დანარჩენი დამხმარე კლასებია. ამ შემთხვევაში ყველა პროგრამა სათითაოდ უნდა გაეშვას და ამიტომ package-ში გაერთიანება არასწორია. თუ რომელიმე IDE-ში წერთ რომელიც ავტომატურად აერთიანებს package-ებში, გამოგზავნის წინ ჩამოუარეთ ყველა source ფაილს და ხელით წაუშაღეთ ხაზი რომელიც package-ში აერთიანებს. სხვა შემთხვევაში კოდი ტერმინალიდან არ გაეშვება.

თუ წერთ Python 2-ზე: გაითვალისწინეთ რომ Python 2 და Python 3 სხვადასხვა ენაა და ერთი კოდი მეორეს ინტერპრეტატორით ხშირად არ ეშვება ხოლმე. ამიტომ, თუ Python 2-ზე წერთ გამოგზავნისას, მიუთითეთ (Classroom-ზე პირად კომენტარად) რომ ეს Python 2 უნდა გამოვიყენო და არა Python 3. (default-ად Python 3-ს ჩავეთვლი).

დაგვიანება

თითოეულ დავალებაში, ვადას გადაცილებულ ყოველ შესრულებულ საათზე დაგაკლდებათ ამ დავალებაში თქვენს მიერ მიღებული ქულის 0.5% (სანამ ქულა 0-ს არ მიაღწევს). ეს წესი მოქმედებს შესწორებული დავალების ხელმეორედ გამოგზავნის შემთხვევაშიც. თუ დავალება ნაწილებად ატვირთეთ ან შეასწორეთ, ჯარიმა გამოითვლება ბოლო ნაწილის ან ბოლო შესწორების ატვირთვის დროით. ჯარიმა მთლიან დავალებას ეხება და არა მხოლოდ იმ ნაწილს რომელიც ბოლოს ატვირთეთ/შეასწორეთ.

ეს წესი იმათაც ეხებათ, ვისაც დავალების ვადები წლის ბოლომდე აქვთ გადაწეული. ამ შემთხვევაში საჯარიმო დროის დათვლა გადაწეული ვადის გასვლიდან დაიწყება.

პროგრამის გაშვება

პროგრამა უნდა ეშვებოდეს თქვენს მიერ არჩეული ენის სტანდარტული კომპილატორით (ან ინტერპრეტატორით). არ გამოიყენოთ არასტანდარტული (და/ან ფასიანი) ბიბლიოთეკები.

ზოგიერთი პროგრამა ფაილიდან კითხულობს ან ფაილში წერს ხოლმე ინფორმაციას. ასეთ შემთხვევებში წასაკითხი/ჩასაწერი ფაილების სახელები პროგრამას ტერმინალიდან არგუმენტებად გადმოეცემა ხოლმე. გაითვალისწინეთ ეს ფაქტი და არ შეიტანოთ ფაილების სახელები თქვენს პროგრამაში კონსტანტებად. ასევე არ მოსთხოვოთ მომხმარებელს მათი შეყვანა პროგრამის მსვლელობის დროს.

ფაილების ორგანიზება და გამოგზავნა

ყველა პროგრამას უნდა დაარქვას ის სახელი, რაც საკითხში იქნება მითითებული. (მაგალითად თუ მითითებულია “Example.xxx” და პროგრამას წერთ ჯავაზე დაარქვით მას Example.java) დაწერილი პროგრამები (მხოლოდ source ფაილები!) მოათავსეთ ერთ folder-ში*. ამ folder-ს დაარქვით თქვენი freeuni მეილის პრეფიქსი ანუ თქვენი ელ-ფოსტის მისამართი “@” სიმბოლომდე (მაგალითად ntsim04). ეს folder-ი შეკუმშეთ zip და ისე ატვირთეთ [google classroom](#)-ზე. არ დაგავიწყდეთ classroom-ზე submit-ის გაკეთება.

გთხოვთ ზუსტად დაიცვათ ფაილების გამოგზავნის წესი, რადგან დავალებების გასწორება ავტომატურად ხდება ხოლმე და არასწორი ფორმატით გამოგზავნილი ფაილები სავარაუდოდ ვერ გაივლის ამ ავტომატურ შემოწმების პროცესს.

* ზოგიერთი შემკუმშავი საშუალებას გაძლევთ folder-ში მოთავსების გარეშე შეკუმშოთ რამდენიმე ფაილი ერთად, მაგრამ ამ კურსის დავალებებისთვის ეს არ გააკეთოთ. არქივის გახსნის შემდეგაც, ფაილები ერთ folder-ში უნდა იყოს.