**Snieguma līmeņa apraksts Python projekta veidošanā**

**Uzdevums:** Izvēlēties ideju savam Python projektam, uzrakstīt specifikāciju un, pamatojoties uz to, izstrādāt projektu.

|  |  |  |  |
| --- | --- | --- | --- |
| **Kritēriji** | **Pamata līmenis** | **Vidējais līmenis** | **Augstākais līmenis** |
| **Specifikācijas izveide (5 punkti)** | Specifikācija ir vispārīga (derīga arī līdzīgiem projektiem), trūkst kāda sadaļa (ievaddati, izvaddati vai tml.) | Specifikācija ir vispārīga (derīga arī līdzīgiem projektiem), viss nepieciešamais ir minēts | Specifikācija ir precīzi atbilstoša izstrādātajam projektam, uzskatāmi parādīta paredzamā datu izvade |
| **Projekta darbība (3 punkti)** | Programma atbilst uzdevumam | Programma atbilst uzdevumam un specifikācijai | Programma atbilst uzdevumam un specifikācijai, izmantota radoša pieeja (ideja vai realizācija) |
| **Projekta gatavība (3 punkti)** | Programma ir izstrādāta, taču nedarbojas (uzrāda kļūdas, nedarbojas kā paredzēts) | Programma ir izstrādāta un darbojas, taču ir atsevišķas kļūdas (piemēram, korekti nenostrādā spēles beigas, neievērojot spēles noteikumus iespējams tos pārkāpt utml.) | Programma darbojas bez kļūdām |
| **Izmantotais kods (3 punkti)** | Izmantoti tikai stundās mācītie paņēmieni un metodes | Izmantoti stundās mācītie paņēmieni un metodes, bet darba gaitā apgūti arī jauni elementi | Programmā izmantoti vairāki elementi, kas stundās netika mācīti; ir skaidrs, ko katrs no tiem nozīmē un dara uzdevumā |
| **Funkciju izmantošana (3 punkti)** | Ir izveidota viena funkcija vienkāršam risinājumam | Ir izveidotas vairākas funkcijas vienkāršiem risinājumiem | Izveidotas funkcijas, kur tās loģiski nodrošina programmas darbību |
| **Koda optimizācija (3 punkti)** | Kods nav optimizēts, ir izveidoti lieki mainīgie, programmā atrodamas nevajadzīgas koda daļas | Programmā nav atrodami nevajadzīgi mainīgie vai koda daļas | Kods ir optimizēts, uzrakstīts iespējami īsi |
| **Termiņa ievērošana (1 punkts)** | Darbs nodots termiņā |  |  |
| **Klasesbiedru vērtējums (3 punkti)** | Darbs atbilst uzdevumam, bet ir vairāki trūkumi | Darbs atbilst uzdevumam | Darbs atbilst uzdevumam, pārsteidz ar savu ideju/izpildījumu utml. |

**PROGRAMMATŪRAS IZSTRĀDES SPECIFIKĀCIJA**

|  |  |
| --- | --- |
| Programmatūras (Idejas) apraksts Kas programmā paredzēts (1 punkts) | Izveidota spēle krustiņi un nulītes |
| Konkrētas prasības (3 punkti) Kāds ir aprakstītās programmas algoritms? Kādā secībā norisinās spēles gaita? Kādas funkcijas programmai jānodrošina? | Spēle ļauj diviem spēlētājiem spēlēt kopā 3x3 laukumā. Pirmais spēlētājs sāks ar krustiņiem (X), bet otrais spēlētājs sāks ar nullēm (O). Gājēju kārta mainīsies pēc katra gājiena. Piemēram, ja pirmais gāja (X) tad otrais ies(O) pētām atkal (X) šis norisināsies tik ilgi līdz vairs nebūs vietas laukumā. Mērķis ir, lai spēlētājam būtu pirmais, kurš novieto savus simbolus pa diagonāli, vertikāli vai horizontāli secīgi, lai izveidotu triju līniju pēc kārtas, lai uzvarētu. |
| Ārējā saskarne (1 punkts) Kā izskatīsies datu ievade un izvade | Datus ievadei izmantojos ciparus un skatīsies kurš spolētājs iet , lai saprastu vai ir jāievieto(x) vai (o). Izvadē pēc izvēlētās vietas tiks ievietots simbols un to varēs redzēt izmaiņās.  Laukums varētu izskatīties šādi:  +---+---+---+  | 1 | 2 | 3 |  +---+---+---+  | 4 | 5 | 6 |  +---+---+---+  | 7 | 8 | 9 |  +---+---+---+ |

**PROGRAMMAS IZSTRĀDĒ IZMANTOTAIS**

|  |  |
| --- | --- |
| Izmantotās metodes un rīki, kas stundās netika mācīti (3 punkti, jābūt pierādījumam) | 2d grid array, Nested loops, Eror handling, Indexing |
| Interneta adreses vai citi materiāli, kas tika izmantoti darba procesā | <https://www.architectalgos.com/navigating-the-grid-a-comprehensive-guide-to-2d-arrays-a81fd6b2e16f>  <https://textbooks.cs.ksu.edu/intro-python/05-loops/08-nested-while/index.html>  <https://www.geeksforgeeks.org/python-exception-handling/>  <https://www.geeksforgeeks.org/python-key-index-in-dictionary/> |