

P. Kasparaitis. Objektinis programavimas C++

12 pratybos. Paveldėjimo praktinis uždavinys

Šiose pratybose reikės suprogramuoti žaidimą „Kartuvės“ panaudojant jau sukurtas klases. Kaip žaidžiamas žaidimas „Kartuvės“ žr. [Kartuvės \(žaidimas\) – Vikipedija \(wikipedia.org\)](https://lt.wikipedia.org/wiki/Kartuvės_(žaidimas)).

1) Išanalizuokite pateiktą programą. Įrašyto kodo keisti negalima, tačiau galima papildyti.

```
#include <iostream>
using namespace std;
class GuessWord;
class Action {
public:
    virtual void run(GuessWord* guess_word) = 0;
};

class GuessWord {
    std::string word;
    std::string hidden_word;
public:
    GuessWord(const std::string& word);
    void guess_char(char chr);
    bool have_won();
    void reveal_letter(int index);
    void set_word(const std::string& word);
    void show_hidden_word();
};

GuessWord::GuessWord(const std::string& word) {
    set_word(word);
}

void GuessWord::guess_char(char chr) {
    for (int i = 0; i < word.size(); i++) {
        if (word[i] == chr && hidden_word[i] == '*') {
            reveal_letter(i);
        }
    }
}

bool GuessWord::have_won() {
    return word == hidden_word;
}

void GuessWord::reveal_letter(int index) {
    if (index >= 0 && index < word.size()) {
        hidden_word[index] = word[index];
    }
}

void GuessWord::set_word(const string& word) {
    this->word = word;
    this->hidden_word.clear();
    for (int i = 0; i < word.size(); i++)
        hidden_word.push_back('*');
}

void GuessWord::show_hidden_word() {
    cout << "WORD: " << hidden_word << endl;
}

int main(int argc, char** argv) {
    // Programuojate jus
}
```

2) Realizuokite tris skirtingas **Action** vaikinės klases:

- **GuessLetter**, kuri leidžia spėti raidę žodyje.
- **RevealLetter**, tai pagalba žaidėjui (angl. powerup), kuri leidžia atidengti vieną žaidėjo pasirinktą raidę spėjamame žodyje,
- **RevealRandom**, tai dar viena pagalba žaidėjui, kuri atidengia atsitiktinę raidę žodyje.

3) Metode **main** realizuokite „Kartuvių“ žaidimą.

Išveskite į ekraną pranešimą žaidėjui su galimų veiksmų sąrašu ir „uždengtu“ spėjamu žodžiu. Tada žaidėjas pasirenka veiksmą, veiksmas įvykdomas ir parodomas rezultatas. Pvz. galima veiksmų seka:

```
ACTIONS: GuessLetter letter; RevealLetter index; RevealRandom
WORD: *****
> GuessLetter l
WORD: **11*
> RevealLetter 5
WORD: **1lo
> RevealRandom
WORD: h*1lo
> GuessLetter e
WORD: hello
```

Čia žaidėjo įvestos eilutės prasideda simboliu „>“. Žinoma, komandų pavidalą galite sugalvoti patys.

4) Į klasę **Action** įdėkite statinį lauką, kuriame būtų skaičiuojamas bendras panaudotų komandų skaičius. Įdėkite apribojimą, kad jei žaidėjas įvykdęs 10 komandų žodžio neatspėja, jis žaidimą pralaimi. Įdėkite apribojimą, kad pagalbos žaidėjui komandas (**RevealLetter** ir **RevealRandom**) žaidėjas gali panaudoti ne daugiau kaip 2 kartus.

5) Sudėkite visas žaidėjo įvestas komandas į vektorių. Vektoriaus elementus atspausdinkite. Naudodami **dynamic_cast** sužinokite, kuriuose vektoriaus elementuose yra pagalbos žaidėjui komandos ir prie jų atspausdinkite „(*)“.