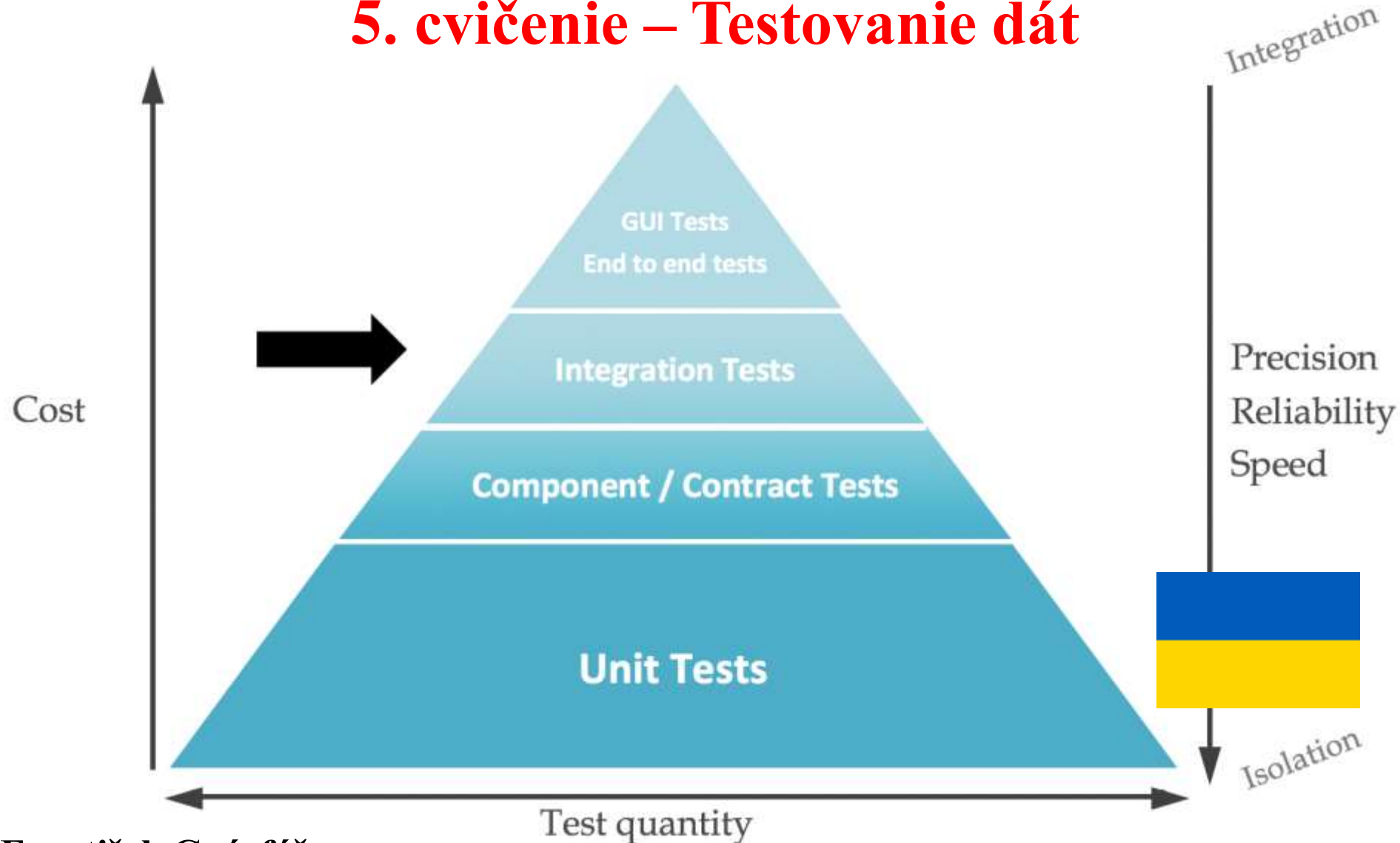


# Extrémne programovanie

2021 – 2022

## 5. cvičenie – Testovanie dát



# K minulému cvičeniu

K cvičeniu TDD Kata sa vrátíme nabudúce. Teraz iba krátke zhodnotenie:

Väčšina z Vás **zvládla konverziu správnych rímskych čísel** . . .

. . . a viac, či menej **nezvládla tie nesprávne**.

Máte problémy s **odhaľovaním a ošetrovaním možných nesprávnych vstupov**.

```
@Test
public void testRimVIV() {
    assertEquals(-9999, Main.convertToInt("VIV"));
}
```

# Dôvody za/proti písaniu testov

## Dôvody proti:

Lenivosť, nechut'

Neschopnosť objaviť zlyhávajúce.

Správnosť kódu, ktorý píšeme.

Strata času.

**Testovací paradox:** napísané testy nie je treba.

Strach

## Dôvody za:

Príkaz zhora

Chybovosť netestovaného kódu.

Nesprávnosť kódu, ktorý píšeme.

RefaktORIZÁCIA

**Testovací paradox:** napísané testy nás chránia pred budúcimi chybami.

Strach

# POSLEDNÁ PREDNÁŠKA

# Čo je lepšie a prečo?

```
TEST_F(Test, Pravdivost) {  
    Number number;  
    number.set(55);  
    ASSERT_TRUE(number.get() == 3);  
}
```

```
TEST_F(Test, Rovnost) {  
    Number number;  
    number.set(55);  
    ASSERT_EQ(number.get(), 3);  
}
```

# Aký je problém s týmto testom?

```
TEST_F(Test, Double) {  
    Double number;  
    number.set(7.77);  
    ASSERT_THAT(number.get(), Eq(3.33 + 4.44));  
}
```

# Načo je dobré toto?

```
TEST_F(Test, KonverziaMedziMenami) {  
    KalkulackaKurzov kalk;  
    kalk.nastavKurzMeny("DOLAR", 1.22);  
  
    string hodnota = kalk.hodnota("DOLAR", 50.5);  
    ASSERT_EQ("41.39", hodnota);  
  
    hodnota = kalk.hodnota("DOLAR", 78.2);  
    ASSERT_EQ("64.10", hodnota);  
}
```

# Aký je problém s týmto?

```
def test_try(self):
    self.assertEqual(self.convertToInt('I'),1) #F
    self.assertEqual(self.convertToInt('V'),5) #T
    self.assertEqual(self.convertToInt('IIV'),-9999) #T
    self.assertEqual(self.convertToInt('IV'),4)#F
    self.assertEqual(self.convertToInt('XX'),20) #T
    self.assertEqual(self.convertToInt('MMLXIX'),2069)
    self.assertEqual(self.convertToInt('MDLCL'),-9999) #F
    self.assertEqual(self.convertToInt('MMM'),-9999) #T
    self.assertEqual(self.convertToInt('IIIII'),-9999) #F
    self.assertEqual(self.convertToInt('VV'),-9999) #T
    self.assertEqual(self.convertToInt('MXXXVI'),-9999) #F
    self.assertEqual(self.convertToInt('mxx'),1020) #T
    self.assertEqual(self.convertToInt(12),-9999) #F
    self.assertEqual(self.convertToInt('VIIPi'),-9999) #F,T
    self.assertEqual(self.convertToInt('CDC'),-9999)
    self.assertEqual(self.convertToInt('LXXIII'),73)
    self.assertEqual(self.convertToInt('IIX'),-9999)
    self.assertEqual(self.convertToInt('MCCMI'),-9999)
    self.assertEqual(self.convertToInt('ID'),-9999)
```



# Ako sa to dá vyriešiť?

```
TEST_F(Test, PrvaDesiatkaExpect) {  
    EXPECT_THAT(rimskieNaInt("I"), Eq(1));  
    EXPECT_THAT(rimskieNaInt("II"), Eq(2));  
    EXPECT_THAT(rimskieNaInt("III"), Eq(3));  
    EXPECT_THAT(rimskieNaInt("IIII"), Eq(4));  
    EXPECT_THAT(rimskieNaInt("IV"), Eq(4));  
    EXPECT_THAT(rimskieNaInt("V"), Eq(5));  
    EXPECT_THAT(rimskieNaInt("VI"), Eq(6));  
    EXPECT_THAT(rimskieNaInt("VII"), Eq(7));  
    EXPECT_THAT(rimskieNaInt("VIII"), Eq(8));  
    EXPECT_THAT(rimskieNaInt("IVIV"), Eq(8));  
    EXPECT_THAT(rimskieNaInt("IX"), Eq(9));  
}
```

# Aký je problém s týmto testom?

```
TEST_F(Test, usporiadanie) {  
    string poleCisel =  
        "3,66,-202,9212,0,-41356,6,12,87,241,55,412,1,-7768,893,-445";  
    ASSERT_EQ(sort(poleCisel),  
        "-41356,-7768,-202,-445,0,1,3,6,12,55,66,87,241,412,893,9212");  
}
```

# Ako testovať toto?

```
void atm::vypisBankovky() const {  
    if (prihlaseny == nullptr) {  
        throw neprihlaseny_ucet();  
    }  
    if (prihlaseny->isAdmin()) {  
        if (getHotovost() == 0) {  
            cout << "V bankomate nie je ziadna hotovost" << endl;  
            return;  
        }  
        getBankovky();  
    }  
}
```

```
TEST_F(testATM, atmPrihlasenySpravca) {  
    spravca testovaciSpravca(2345);  
    bankomat->prihlas(&testovaciSpravca, 2345);  
  
    internal::CaptureStdout();  
    bankomat->vypisBankovky();  
    string vystup = internal::GetCapturedStdout();  
    ASSERT_EQ(vystup, "5: 10, 10: 10, 20: 25, 50: 20, 100: 20");  
}
```

# Na čo je dobré toto?

```
class Kolekcia : public Test {  
    public:  
        Retazec retazec;  
        void SetUp() {  
            retazec.set("Empty");  
        };  
};
```

```
TEST_F(Kolekcia, JePrazdny) {  
    ASSERT_THAT(retazec.get(), Eq("Empty"));  
}  
  
TEST_F(Kolekcia, NieJePrazdny) {  
    retazec.set("IV");  
    ASSERT_THAT(retazec.get(), Eq("IV"));  
}
```

# DNEŠNÉ CVIČENIE

# Testovanie dát

Úlohou cvičenia je napísať program, ktorý pomocou jednotkových testov skontroluje správnosť dát v reálnom súbore, obsahujúcom výsledky sociologického prieskumu na väčšej množine respondentov.

**Testy majú skontrolovať a objaviť chyby vo vstupnom súbore:**

- 1) Nájsť, kde sa nachádza chybný údaj a akú má hodnotu.
- 2) Vypísať, prečo je chybný.

# Zadanie na jednotkové testovanie

Ako vstup dostanete pomerne veľký súbor reálnych dát `suborDat.csv` (vyše 1150 záznamov), popisujúcich výsledky sociologického prieskumu. K tomuto súboru dostanete aj súbor `Dotaznik 1-36.pdf` so zoznamom 36 otázok.

```
cisdot;q1;q2;q3;q4;q5;q6;q7;q8;q9;q10;q11;q12;q13;q14;q15;q16;q17;q18;q19;q20;q21;q22;q23;q24;q25;q26;q27;q28;q29;q30;q31;q32;q33;q34;q35;q36
1;5;7;6;6;3;5;4;5;4;3;3;2;2;3;3;2;2;1;4;4;2;2;4;3;4;4;2;7;7;7;5;7;7;6;99;3
2;5;7;6;6;5;5;6;6;4;2;2;2;1;2;4;2;2;1;4;4;1;2;3;3;4;3;3;7;7;7;7;7;6;7;4;6
3;6;6;6;6;4;5;2;4;2;2;2;2;3;4;2;2;4;4;4;1;2;4;3;3;4;4;7;7;7;6;7;4;6;5;7
4;3;7;7;4;2;5;4;4;3;2;2;3;3;3;4;4;1;4;4;4;2;1;4;3;2;3;4;7;7;7;4;6;4;5;4;7
5;5;7;7;5;4;6;6;6;4;3;2;3;3;3;3;2;1;4;3;3;1;2;4;3;4;2;4;7;7;7;7;7;3;5;4;7
6;5;7;6;5;5;6;5;6;4;3;3;3;2;3;3;1;3;4;4;4;2;1;4;3;4;4;4;7;7;7;6;7;99;5;4;7
7;6;7;6;5;5;6;5;6;5;2;2;3;1;3;2;3;2;3;4;4;2;1;4;3;3;4;3;7;7;7;6;7;5;5;4;7
8;6;7;6;5;3;5;3;5;3;2;2;2;3;1;4;3;3;4;4;3;6;2;4;4;4;1;2;7;7;7;7;7;4;5;4;7
9;7;7;7;4;1;6;4;6;6;4;3;4;3;4;2;4;4;4;4;3;2;4;2;4;4;4;4;3;7;7;5;3;7;7;5;7;7
10;7;4;7;1;1;7;4;4;4;2;1;4;1;4;4;2;3;3;3;4;3;1;4;4;4;3;4;7;7;7;1;7;7;3;7;7
11;5;7;7;7;1;7;4;6;5;4;4;4;1;4;4;3;3;3;3;4;2;2;4;4;4;4;4;7;6;7;3;6;7;7;1;7
.
.
.
1155;7;7;7;4;4;3;4;7;6;1;3;4;1;4;4;3;3;4;4;4;2;2;3;3;4;4;4;7;7;6;7;7;7;4;6;7
1156;7;6;7;6;99;7;6;7;5;2;2;2;1;3;3;3;3;3;4;4;1;1;4;4;2;4;3;7;7;3;4;7;3;6;7;7
1157;4;5;5;2;2;5;5;5;3;4;4;3;2;99;4;3;3;3;4;99;1;3;4;3;4;4;4;6;4;4;99;5;6;99;99
```

V tomto súbore sa nachádza **aspoň desať formálnych chýb** rôzneho typu.

# Pomocný súbor s otázkami

So zadáním dostanete aj pomocný súbor Dotazník 1-36.pdf, v ktorom nájdete zoznam otázok s možnými odpoveďami.

## PREDLOŽTE **POMOCNÝ LÍSTOK 1-9**

**ot. 1-9.** "Existujú rôzne názory na to, čo znamená **byť dobrým občanom**. Uvedte na stupnici 1 až 7, kde "1" znamená "vôbec nie dôležité" a "7" znamená „veľmi dôležité“, **akú dôležitosť pripisujete nasledujúcim konaniam.**"

**POSTUPNE ČÍTAJTE RIADKY 1-9! V KAŽDOM OZNAČTE ODPOVEĎ!**

			vôbec nie dôležité							veľmi dôležité	N	neodpov.
			①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>1.</b>	"Vždy hlasovať vo voľbách.	<b>1</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>2.</b>	Nikdy sa nepokúšať o daňové úniky.	<b>2</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>3.</b>	Vždy dodržiavať zákony a predpisy.	<b>3</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>4.</b>	Pozorne sledovať činnosť vlády.	<b>4</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>5.</b>	Aktívne pôsobiť v občianskych či politických združeniach.	<b>5</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>6.</b>	Snažiť sa pochopiť ľudí s odlišnými názormi.	<b>6</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩
<b>7.</b>	Uprednostňovať niektoré výrobky z politických, etických či ekologických dôvodov aj napriek tomu, že sú trochu drahšie.	<b>7</b>	①	②	③	④	⑤	⑥	⑦		⑨	⑩



# Úloha na jednotkové testovanie dát

## Vašou úlohou bude:

- 1) Načítať dáta do programu v zvolenom jazyku (**v žiadnom prípade nie Excel ani akákoľvek databáza**).
- 2) Napísať metódy, ktoré budú kontrolovať formálnu správnosť dát.
- 3) Napísať jednotkové testy na kontrolu týchto metód.
- 4) Napísať testy, ktoré pomocou týchto metód nájdu chyby v množine dát.

## Zadanie riešte:

1. vo dvojiciach,
2. v jazyku, aký si vyberiete,
3. s testovacím frameworkom pre daný jazyk, ktorý si vyberiete.

# Výsledok

Ako výsledok odovzdajte (zbalené do jedného zip súboru):

- 1) Všetky zdrojáky programu, ktorý ste použili na riešenie.
- 2) Všetky jednotkové testy, ktoré ste vytvorili.
- 3) Textový súbor s výpisom zo zbehnutia testov.
- 4) Textový súbor s konkrétnym zoznamom nájdených chýb,  
Např: riadok 765, hodnota 5 je mimo intervalu hodnôt.

Riešenie odovzdajte do **28. 3. 2022, 04:44 AM**