

# Informatyka I

Wymagania formalne i sposób oceny projektu

## Spis treści

Format projektu.....	2
Ważne terminy .....	2
Sposób punktacji .....	2
Za co przyznawane są punkty .....	2
W jaki sposób można stracić punkty .....	3
Ile należy zebrać punktów, aby otrzymać daną ocenę .....	3
Przykłady.....	3
Przykład 1: .....	3
Przykład 2: .....	3
Przykład 3: .....	3
Przykład 4: .....	3
Przykład 5: .....	4
Tematyka projektu .....	4
Przykładowe tematy projektów: .....	4
Kontakt z prowadzącym .....	4
Zgłoszenia i propozycje tematów .....	5
Prezentacja stanu prac .....	5
Wysłanie projektu do oceny .....	5
Oddanie projektu.....	5
Obrona projektu .....	6
Przykładowe pytania.....	6

## Format projektu

W celu uzyskania pozytywnej oceny z zajęć projektowych należy zrealizować wybrany przez studenta i zatwierdzony przez prowadzącego temat projektu za pomocą języka Python. Projekt uznaje się za zrealizowany, gdy spełnione są **WSZYSTKIE** następujące wymagania:

1. Dotyczy wybranego wcześniej tematu, który został zatwierdzony przez prowadzącego
2. Zdobył dostateczną ilość punktów, zgodnie z systemem punktacji przedstawionym poniżej
3. Był wykonany przez studenta samodzielnie<sup>1</sup>
4. Został przedstawiony prowadzącemu przed końcem semestru

## Ważne terminy

Data	Co należy wykonać <u>przed</u> tą datą
<b>21.10</b>	Przedstawić prowadzącemu proponowany temat
<b>31.10</b>	Ustalić w prowadzącym ostateczny temat
<b>30.11</b>	Przynajmniej raz skontaktować się z prowadzącym (osobiście w trakcie projektów, konsultacji, czy też mailowo) i pokazać stan prac
<b>16.12</b>	Oddać gotowy projekt
<b>Pierwszy dzień sesji zasadniczej</b>	Obronić projekt

## Sposób punktacji

Ocena projektu zależy od sumarycznej liczby zdobytych punktów.

### Za co przyznawane są punkty

Cecha projektu	Ilość punktów
Poprawne działanie	35
Estetyczny i łatwy w interpretacji kod	3
Obecność logicznych komentarzy w kodzie	2
Przemyślany interfejs użytkownika	4
Spójny i zrozumiały format nazewnictwa zmiennych	3
Użycie list	3
Użycie list asocjacyjnych (słowników, map)	6
Zastosowanie podejścia obiektowego	6
Operowanie na plikach	5
Uwzględnienie stanów wyjątkowych (podanie błędnych danych przez użytkownika, obecność kilku identycznych danych, itp.)	4
Wykorzystanie bibliotek, modułów i/lub pakietów zewnętrznych (np. do wyświetlenia okienkowego interfejsu użytkownika, skorzystania z zewnętrznej bazy danych, itp.)	15 punktów za każdą <sup>2</sup> użytą bibliotekę/pakiet/moduł
Sortowanie danych	3 punkty za każdą właściwość względem której można sortować

<sup>1</sup> Konsultacje projektu z prowadzącym, zasięgnięcie opinii i wiedzy osób trzecich oraz szukanie pomocy w literaturze jest dozwolone, a nawet wskazane.

<sup>2</sup> O ile jego użycie jest uzasadnione i projekt wykorzystuje przynajmniej 4 różne klasy/funkcje biblioteki/modułu/pakietu.

## W jaki sposób można stracić punkty

Działanie	Ilość punktów
Przekroczenie terminu zgłoszenia/prezentacji/oddania/obrony	-8 za każdy <b>rozpoczęty</b> tydzień zwłoki
Nieuzgodnienie z prowadzącym tematu w podanym terminie	<b>-10</b>
Nieprawidłowy format wysłanych plików	<b>-6</b>
Wysłanie wiadomości z nieprawidłowym tematem	<b>-6</b>
Plagiat lub kupno/kradzież projektu	<b>-30</b>
Brak zrozumienia własnego kodu podczas obrony projektu	<b>-30</b>

## Ile należy zebrać punktów, aby otrzymać daną ocenę

Ocena	Projekt jednoosobowy	Projekt dwuosobowy
<b>3.0</b>	40	70
<b>3.5</b>	45	80
<b>4.0</b>	50	90
<b>4.5</b>	55	100
<b>5.0</b>	60	110

## Przykłady

### Przykład 1:

Adam stworzył program symulujący bazę danych samochodów. Program działa poprawnie (**+35 punktów**), kod jest estetyczny (**+3 punkty**), a zmienne mają spójne i logiczne nazwy (**+3 punkty**). Adam wysłał projekt w terminie i podczas obrony projektu udowodnił, że rozumie kod źródłowy. Adam zebrał łącznie **35+3+3=41** punktów. To wystarczy na ocenę **3.0**.

### Przykład 2:

Anna stworzyła program symulujący bazę danych filmów. Program działa poprawnie (**+35 punktów**), kod jest estetyczny (**+3 punkty**), a zmienne mają spójne i logiczne nazwy (**+3 punkty**). Anna oddała projekt cztery tygodnie po wyznaczonym terminie ( $4 * -8$  **punktów**). Anna zebrała łącznie **35+3+3+4\*-8=41-32=9** punktów. To **za mało** na ocenę 3.0, więc Anna **nie uzyskała oceny pozytywnej** z projektu.

### Przykład 3:

Jan stworzył program symulujący magazyn sklepu. Program działa poprawnie (**+35 punktów**), kod jest estetyczny (**+3 punkty**), a zmienne mają spójne i logiczne nazwy (**+3 punkty**). Jan użył list (**+5 punktów**), a także dodał możliwość posortowania względem nazwy produktu, stanu magazynowego i nazwy dostawcy ( $3 * +4$  **punkty**). Jan zebrał łącznie **35+3+3+5+3\*4=58** punktów. To wystarczy na ocenę 4.5. Jan po konsultacjach z prowadzącym, postanowił uwzględnić stany wyjątkowe (**+4 punkty**). Wysłał projekt w terminie i podczas obrony projektu udowodnił, że rozumie kod źródłowy. Uzyskał ocenę **5.0**.

### Przykład 4:

Mateusz stworzył równie zaawansowany program co Jan (**62 punkty**). Wysłał projekt w terminie, ale nie przyszedł przed końcem semestru na obronę. Mateusz **nie uzyskał oceny pozytywnej** z projektu, bo **nie spełnił wszystkich wymagań** na ocenę pozytywną.

### Przykład 5:

Przykładowy projekt, który zasługuje na ocenę 3.0, znajduje się pod następującym adresem: <http://stawarz.edu.pl/informatyka/programNa3.0.py>. Jest to baza danych pierwiastków chemicznych. Projekt działa poprawnie (+35 punktów), oferuje możliwość wglądu w dane i dodania nowych i wykorzystuje listy asocjacyjne (+6 punktów).

## Tematyka projektu

Temat projektu może być dowolny, ale **unikalny** w ramach grupy projektowej. Projekt może być realizowany samodzielnie lub w dwie osoby.

W projekcie muszą być przetwarzane dane (samochodów, filmów, obrazów, dane astronomiczne i/lub inne). Użytkownik musi mieć możliwość wprowadzać dane do „bazy” i wyświetlać zbiór wprowadzonych danych. Zbiór może ulegać usunięciu podczas restartu programu i może (ale nie musi) zawierać jakieś dane wzorcowe w momencie uruchomienia.

**Dozwolone** jest wybranie tematu niezwiązanego z przetwarzaniem danych. W takim wypadku temat musi być na podobnym poziomie złożoności co tematy zaproponowane poniżej. Niestandardowe tematy (gry, symulacje, inne przydatne programy) będą oceniane według osobnego klucza i nagradzane dodatkowymi punktami.

Temat projektu należy zgłosić do prowadzącego zajęcia. Realizacja projektu zaczyna się dopiero po zatwierdzeniu przez prowadzącego tematu. Prowadzący może odrzucić temat, jeżeli taki pojawił się w grupie lub zasugerować zmianę, jeżeli jest zbyt prosty lub złożony. Zgłoszenia można dokonać **wyłącznie drogą mailową**.

W treści wiadomości należy podać tytuł projektu i krótki opis tego jakie dane będą użyte, czy i po czym będzie następować sortowanie, jakie funkcje będzie miał program i tym podobne.

### Przykładowe tematy projektów:

- Baza danych samochodów przechowująca model, markę, rok produkcji, producenta, etc.
- Baza danych produktów w sklepie przechowująca nazwę produktu, cenę, ilość na magazynie, nazwę dostawcy, typ, etc.
- Program do zarządzania kinem przechowujący informacje o filmach, salach, seansach, etc.
- Kalkulator posiadający pamięć poprzednich operacji. Możliwość sortowania po czasie wykonania, wartości, ilości operacji, etc.

### Kontakt z prowadzącym

Kontakt z prowadzącym odbywa się w trybie hybrydowym. Zajęcia organizacyjne i obrona projektu są zawsze przeprowadzane w trybie stacjonarnym. Czynności wymagające archiwizacji,

przeprowadzane są drogą elektroniczną na adres [p.stawarz@prz.edu.pl](mailto:p.stawarz@prz.edu.pl). Wiadomości wysyłane na inny adres, będą **odrzucone**.

Wiadomość **musi** również mieć określony **tytuł**. Wiadomości wysyłane z niestandardowym tytułem, będą **odrzucone**.

### Zgłoszenia i propozycje tematów

Wiadomości zawierające propozycje tematów, muszą mieć następujący tytuł:

Imię nazwisko numerGrupy – zgłoszenie tematu

Na przykład:

Adam Mickiewicz L01 – zgłoszenie tematu

Treść wiadomości powinna zawierać uproszczony opis tematu.

### Prezentacja stanu prac

Wiadomości zawierające prezentację stanu prac, muszą mieć następujący tytuł:

Imię nazwisko numerGrupy – postęp prac

Na przykład:

Adam Mickiewicz L01 – postęp prac

Do wiadomości powinien być dołączony załącznik w odpowiednim formacie, który prezentuje aktualny w momencie wysłania wiadomości stan prac.

### Wysłanie projektu do oceny

Wiadomości zawierające gotowy projekt, muszą mieć następujący tytuł:

Imię nazwisko numerGrupy – oddanie projektu

Na przykład:

Adam Mickiewicz L01 – oddanie projektu

Do wiadomości powinien być dołączony załącznik w odpowiednim formacie, zawierający gotowy projekt, który podlegać będzie ocenie i obronie.

## Oddanie projektu

Projekt należy wysłać drogą mailową do prowadzącego. Jeżeli projekt składa się z jednego pliku, należy wysłać ten plik (w formacie \*.py). Jeżeli projekt składa się z wielu plików, należy utworzyć projekt w środowisku PyCharm i wysłać cały projekt, dbając o to, by nie wymagał dodatkowej konfiguracji.

Po otrzymaniu wiadomości zgodnej z zasadami wymienionymi w rozdziale „kontakt z prowadzącym”, prowadzący niezwłocznie oceni temat i odpisze informując o proponowanej ocenie. Jeżeli prowadzący nie odpisze w ciągu tygodnia, należy wiadomość wysłać ponownie, dołączając dodatkowo zdjęcie z datą wysłania wiadomości poprzedniej (jeżeli data ma wpływ na ocenę).

## Obrona projektu

Obrona projektu następuje po zatwierdzeniu projektu przez prowadzącego. Na obronę należy przyjść z laptopem lub innym urządzeniem pozwalającym przetestować projekt. Prowadzący zada kilka prostych pytań dotyczących projektu i wystawi ocenę.

### Przykładowe pytania

- Dlaczego w danym miejscu kodu wstawiono tabulator? Co by się stało, gdyby go brakło?
- Co dokładnie robi ta linijka kodu?
- Proszę omówić algorytm działania programu.
- Co przechowuje dana zmienna?